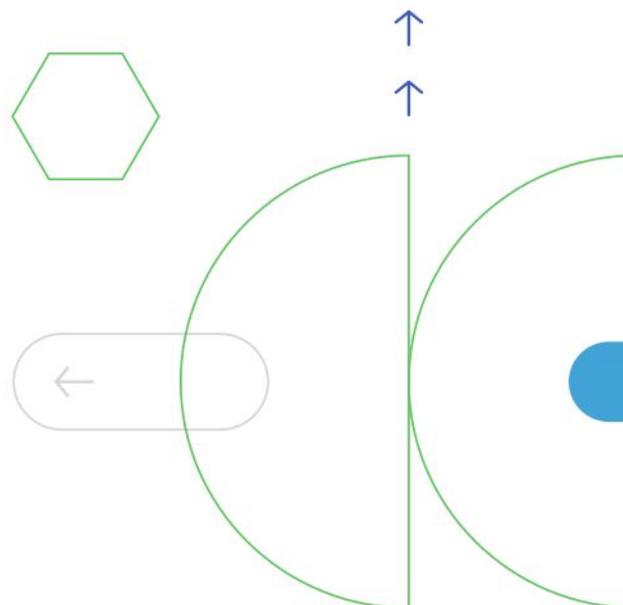


2023

# JADicc

Jornadas Argentinas de Didáctica  
de las Ciencias de la Computación



## LIBRO DE ACTAS

Facultad de Informática

UNIVERSIDAD NACIONAL DEL COMAHUE



Facultad de Informática  
UNIVERSIDAD NACIONAL DEL COMAHUE



<Program.AR/>  
DIEZ AÑOS

# JORNADAS ARGENTINAS DE DIDÁCTICA DE LAS CIENCIAS DE LA COMPUTACIÓN

JADiCC 2023

DICIEMBRE 2023

NEUQUÉN, PATAGONIA, ARGENTINA

Organizadores:

Facultad de Informática  
Universidad Nacional del Comahue

Fundación Sadosky



Fundación  
**SADOSKY**

<Program.AR/>  
DIEZ AÑOS

**Universidad Nacional del Comahue**

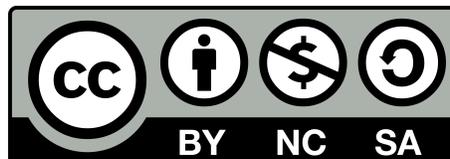
JADiCC 2023 : libro de actas de las Jornadas Argentinas de Didáctica de las Ciencias de la Computación ; Compilación de Jorge P. Rodriguez ; Gabriela N. Aranda ; Laura A. Cecchi. - 1a ed. - Neuquén : EDUCO - Universidad Nacional del Comahue. Editorial Universitaria del Comahue, 2024.

Libro digital, PDF

Archivo Digital: descarga y online

ISBN 978-987-604-652-7

1. Didáctica. 2. Computación. 3. Formación Docente. I. Rodriguez, Jorge P. , comp. II. Aranda, Gabriela N. , comp. III. Cecchi, Laura A. , comp. CDD 370.711



**AUTORIDADES DE LA  
UNIVERSIDAD NACIONAL DEL COMAHUE**

**Rectorado**

**Rectora**

Dra. María Beatriz GENTILE

**Vice-Rector**

Téc. Fernando Paul OSOVNIKAR

**Secretario de Extensión**

Sr. Damián Andrés CANCELO

**Director de la Editorial**

Lic. Enzo Dante CANALE

**Facultad de Informática**

**Decano**

Lic. Guillermo Alejandro GROSSO

**Vice-Decana**

Lic. Silvia Nayum AMARO

# Comité de Programa JADiCC 2023

## Chairs

### Artículos Científicos

Pablo E. Martínez López Gabriela N. Aranda Laura A. Cecchi

### Experiencias

Francisco Bavera Jaqueline Fernandez

### Demostraciones y Pósteres

Francisco Bavera Jaqueline Fernandez Claudia Querugia Claudia Casariego

### Talleres

Marcos Gómez Jorge Rodríguez

Acosta, Araceli (UNC)	Gutiérrez, María de los Milagros (UTN)
Arroyo, Marcelo (UNRC)	Irrazabal, Emanuel (UNNE)
Banchoff Tzancoff, Claudia (UNLP)	Lasso, Marta (UNPA)
Blanco, Javier (UNC)	Leonardi, María Carmen (INTIA-UNICEN)
Bordignon, Fernando (UNIPE)	Letzen, Diego (UNC)
Bourdetta, Viviana (UADER)	Lopez y Rosenfeld, Matías (UBA-CONICET)
Brassesco, Virginia (UBA-UNIPE)	Mendez, Analía (UNSE)
Casali, Ana (UNR - CIFASIS)	Millán, Ilda Flavia (UNSJ)
Colussi, Natalia (UNR)	Monjelat, Natalia (IRICE-CONICET-UNR)
Company, Ana María (UNNE)	Pintos, Noelia (UTN)
Daniele, Marcela (UNRC)	Poliche, Valeria (UNCA)
Dapozo, Gladys (UNNE)	Puricelli, Fernando (UNaHur)
Del Dago, Gustavo (F. Sadosky)	Queiruga, Claudia (UNLP-LINTI)
Dughera, Lucila (CONICET-e TCS-CCTS)	Rambo, Alice Raquel (UNaM)
Echeveste, Maria Emilia (UNC)	Reyes, Nora (UNSL)
Fernandez, Jacqueline (UNSL)	Rodriguez, Jorge (UNComa)
Fernández, Gonzalo Pablo (UNQ-UBA)	Romero, Lucila (UNL)
Frati, Fernando Emmanuel (UNdeC)	Sanzo, Alfredo (F. Sadosky-UBA)
Golobisky, María Fernanda (UTN)	Tugnarelli, Mónica (UNER)
Gómez, Marcos J. (UNC-F. Sadosky)	Werenizky Curia, María Cristina (UNT)
Gonzalez, Carolina (UNJu)	Wolovick, Nicolás (UNC)
Guatelli, Renata (UNLaM)	Zanarini, Dante (UNR-F. Sadosky)

# Comité Organizador JADiCC 2023

Jorge P. Rodríguez (UNComa)  
Magdalena Garzon (Fundación Sadosky)  
Gabriela N. Aranda (UNComa)  
Laura A. Cecchi (UNComa)  
Pablo E. Martínez López (UNQ)  
Francisco Bavera (UNRC)

Ian Franco Matías Acosta  
María Claudia Allan  
Angel Avellaneda  
Leo Bruno  
Jonathan Cabrera  
Claudia Casariego  
Luis Coralle  
Marcos Cortez  
Franco Fabris  
Marcos Gómez  
Micaela Malaspina  
Federico Nahuel Mamani  
Alejandra Miranda

María Elvira Monserrat Vidal  
Alejandro Mora  
Susana Beatriz Parra  
Rodrigo Piersigilli  
Inés Reutemann  
Inés Roggi  
Sandra Emilce Roger  
Valeria Rotter  
Alfredo Sanzo  
Romina Tealdi  
Valeria Vortisch  
Valentina Villarroel

# Índice

## *Trabajos científicos*

- Ciclo de Seminarios “Ciencias de la Computación en la Escuela”. Aspectos centrales y primeras conclusiones sobre su implementación en la Provincia de Córdoba**  
*Martín Torres, Verónica Pacheco, Painé Pintos* 2
- Resultados en el uso de ChatGPT para resolver ejercicios de Teoría de la Computación**  
*Emanuel Irrazábal, Rubén A. Bernal, Lourdes Romera, Jeremías González* 19
- El impacto en la concepción neutral de la Inteligencia Artificial de un taller para estudiantes de escuela secundaria**  
*Marcos J. Gómez, Julián Dabbah, Alfredo Sanzo* 39
- Reflexiones sobre la enseñanza de Confiabilidad y Seguridad del Software en Bioingeniería con Programación Basada en Contratos**  
*Jordán F. Insfrán, Javier E. Diaz Zamboni* 56
- Formar profesores en Ciencias de la Computación para la educación obligatoria. Un aporte y un desafío desde la universidad**  
*Virginia Brasesco, Fernando Bordignon, Cecilia Sagol, Javier Kullock* 72
- Mejorando la enseñanza en situaciones de ACSC síncronas basadas en texto mediante reconocimiento de emociones**  
*Germán Lescano, Daniela Missio, Rosanna Costaguta* 87
- Relevamiento de conocimientos previos de programación en el nivel universitario**  
*Gonzalo Pablo Fernández, Cecilia Martínez, Pablo E. “Fidel” Martínez López* 103
- La Accesibilidad Web: una perspectiva de formación profesional en las Ciencias de la Computación**  
*Javier Díaz, Alejandra Schiavoni, Paola Amadeo, Ivana Harari, Ana M. Ungaro* 118
- Comarca Digital: Impulsando la enseñanza de la Computación en el Nivel Inicial**  
*Cristián Rojo Pérez, Natalia Andrea Zalazar* 130
- Enseñanza de la Inteligencia Artificial en la Escuela Secundaria. Un enfoque metodológico en acción**  
*Sol Bertochi, Jorge Navarro, Jorge P. Rodriguez, Laura A. Cecchi* 146
- Minirobot app: Un caso de estudio y puesta en aula**  
*Laura Felice, Paula Tristán, Fermín de Fiore, Francisco Prieto* 159

<b>Hacia un análisis de contenidos y lenguajes de programación en materias iniciales de programación de carreras de Ciencias de la Computación</b> <i>José Massa, Carmen Leonardi, Virginia Mauco, Laura Felice, Jorge Doorn</i>	173
<b>ArguiBot IDE: Un entorno de desarrollo basado en la programación en bloque para la enseñanza de la robótica usando la plataforma Arduino</b> <i>Nicolás Streri Marcela Daniele Marcelo Uva</i>	189
<b>Los juegos de adivinación: Análisis didáctico de una propuesta para la enseñanza</b> <i>Flavia Buffarini, Fabiana Rosso, Francisco Bavera</i>	204
<b>Resultados preliminares del curso La Programación y su Didáctica</b> <i>Francisco Bavera, Teresa Quintero, Marcela Daniele</i>	222
<b>Una primera aproximación al contexto de implementación en Jujuy de los espacios curriculares de “Robótica y Programación”</b> <i>María Fernanda Villarrubia, María Emilia Echeveste</i>	237
<b>Festivales de robótica del Gran Río Cuarto: Alternativa para el aprendizaje en comunidad en contextos no formales</b> <i>Ariel Ferreira Szpiniak, Carlos Federico Etcheverry, Carlos Maximiliano Correa</i>	250
<b>Enseñar Ciencias de la Computación en la escolaridad obligatoria. Aportes para una didáctica específica</b> <i>Magdalena Garzón, Alfredo Héctor Sanzo</i>	268
<b>Hola Mundo. Tus primeros pasos en programación</b> <i>Jorge Rodriguez, Silvana Hubinsky, Viviana Sanchez, Guillermo Guerrero, Valeria Flores, Aldo Matamala, Federico Amigone, Celeste Ramos, Pablo Kogan, Jorge Navarro, David Torchinsky</i>	284
<b>Nuevo contexto en las carreras de informática. ¿Con qué estrategias lo enfrentamos?</b> <i>Gladys Dapozo, Cristina Greiner, Raquel Petris, Ana María Company, María Cecilia Espíndola</i>	294
<b>Pasayo: Una plataforma extendida para enseñar a programar a personas en el espectro autista</b> <i>Jorge Rodríguez, Federico Amigone, Guillermo Ariel Guerrero</i>	309
<b>PRENDER: Una propuesta didáctico-pedagógica para la enseñanza de las Ciencias de la Computación</b> <i>Christian Cossio-Mercado, Gonzalo Pablo Fernández</i>	322
<b>Medición del Pensamiento Computacional</b> <i>Fabio Fernando González</i>	355

**Los objetos de aprendizaje: Insumos para la tutoría en programación** 366  
*Marcela F. López, Eduardo F. Fernández, Paola del Olmo*

## *Experiencias*

**Vamos a la Escuela Primaria. Contactos tempranos con las Ciencias de la Computación** 381  
*Rodrigo Piersigilli, Ian Franco Acosta, Susana Parra, Jorge Rodríguez*

**Enseñanza de fundamentos conceptuales de programación usando Arduino** 390  
*Gonzalo Pablo Fernández, Christian Cossio-Mercado*

**La programación y robótica en la formación docente del Nivel Inicial. Estrategias y resultados de un dispositivo de formación situada** 409  
*Mariela I.S. Hirtz, Silvina Manzor, Susana Morales*

**¿Cómo incorporan los dispositivos robóticos escolares los profesores del nivel primario y medio de la Provincia de Río Negro? Resultado de un dispositivo de formación docente** 420  
*Laura J. Peña, M. Eugenia Walker, Nicolás A. I. Ticac, Facundo Bárcena*

**Robótica educativa con hardware y software libre como herramientas para la enseñanza de programación. Una experiencia en bibliotecas populares y espacios no formales de aprendizaje** 435  
*Valentín Basel*

**La operacionalización de los conceptos de Alfabetización, Pensamiento y Participación Computacional en el ámbito escolar. Revisión de antecedentes** 447  
*Ma. Cecilia Martínez, Natalia Monjelat, Echeveste María Emilia, Valeria González, Martín Torres*

**Integración de las prácticas científicas y la programación con JavaScript para el aprendizaje de la genética en la escuela secundaria** 457  
*Gimena B. Fussero, Maricel Occelli, Nicolás Wolovick, Héctor García*

**La construcción contextualizada de materiales didácticos digitales interactivos, desde una perspectiva de Soberanía Tecnológica** 467  
*Ana Carolina Flores Perrero*

**Propuestas de modelización para fortalecer el desarrollo profesional de docentes de Computación, Matemática y Educación Tecnológica** 482  
*María Emilia Echeveste, Araceli Coirini, Lourdes Aguiar Cau, Jonathan Alonso, Nicolás Balmaceda, Érica Zarate, Talía Monjes, Fabricio Carinelli*

**Acercar la Inteligencia Artificial a la escuela. Una experiencia de extensión de la UNLP** 494  
*Isabel Kimura, Claudia Queiruga, Claudia Banchoff Tzancoff, Sofía Martín*

**Taller AYDS-IS: propuesta didáctica para la integración de Análisis y Diseño de Sistemas e Ingeniería de Software**

*Marcela Daniele, Marcelo Uva, Franco Brusatti, Ariel Arsaute, Daniela Solivellas, Mariana Frutos*

506

**Actividades de aplicación en curso de análisis de los datos para Ingeniería sobre sesgo de género, abordaje de la equidad e inclusión**

*Verónica Herrero*

517

**Acercamiento a la programación visual. Una experiencia virtual con docentes de la región patagónica**

*Edith Lovos, Martín Goin*

526

**Los dispositivos de formación docente en educación digital que acompañan la entrega de equipamiento de programas nacionales en la Provincia de Río Negro**

*Estefanía Zurbrigk, Mariela I.S. Hirtz*

536

# JADiCC

Jornadas Argentinas de Didáctica  
de las Ciencias de la Computación

1 y 2 de diciembre de 2023  
NEUQUÉN  
PATAGONIA - ARGENTINA

---

*Trabajos científicos*

**Ciclo de Seminarios “Ciencias de la Computación en la Escuela”  
Aspectos Centrales y Primeras Conclusiones sobre su Implementación  
en la Provincia de Córdoba**

Martín Torres, Verónica Pacheco, Painé Pintos  
{mtorres341, vpacheco, ppintos}@isep-cba.edu.ar

Dpto. de Ens. de Tec. Digitales e Informática  
Instituto Superior de Estudios Pedagógicos

## **Resumen**

El diseño e implementación de propuestas de formación docente para llevar adelante la integración de contenidos de Ciencias de la Computación en los sistemas educativos, implica asumir una serie de desafíos, vinculados a la diversidad de los contextos institucionales y del propio colectivo docente. También constituye una oportunidad de pensar y proponer enfoques críticos para la enseñanza de estos temas. El presente trabajo intenta condensar los aspectos centrales del Ciclo de Seminarios: “Ciencias de la Computación en la Escuela”, desarrollado por el ISEP (Instituto Superior de Estudios Pedagógicos) y destinado a docentes de educación primaria y secundaria de la Provincia de Córdoba. El mismo se está implementado actualmente en la jurisdicción.

El escrito organiza su abordaje en cuatro apartados. En la introducción se presenta el contexto en el cual surge el ciclo de seminarios. En el siguiente se describe el enfoque que guía la propuesta y su materialización en los contenidos seleccionados, así como los principios pedagógicos y didácticos que orientan la enseñanza. En el tercer apartado, se pone el acento en las características del dispositivo de formación para, finalmente, en el apartado de las primeras conclusiones, compartir aquellas valoraciones que hemos podido construir en estos primeros meses de puesta en marcha de la iniciativa.

**Palabras clave:** formación docente, Ciencias de la Computación, enseñanza, cultura digital, tecnologías digitales.

## **Introducción**

Antes de las actuales definiciones curriculares para la incorporación de contenidos de Ciencias de la Computación en los sistemas educativos, la integración de la computación en las escuelas -iniciada en la década de 1980- atravesó distintas etapas. Estas pueden distinguirse según la perspectiva que las orientaba. La jurisdicción de Córdoba, como otras jurisdicciones del país, no estuvo ajena a este proceso. No es objeto de este trabajo profundizar en los distintos enfoques de enseñanza de la informática. Sólo se pretende poner de relieve que todas dejaron huellas en escuelas y docentes respecto a las valoraciones y representaciones de lo que implica enseñar estos contenidos.

Por otra parte, ya en el presente siglo, un elemento importante a tener en cuenta son las iniciativas desplegadas desde el estado para la provisión de equipamiento informático. Vamos a considerar las tres más importantes en nuestro país: el Plan Conectar Igualdad (PCI), Primaria Digital (PD) y Aprender Conectados (AC), lanzados en 2010, 2012 y 2018 respectivamente. Una primera cuestión se refiere al alcance. El PCI se pensó para secundaria y algunos institutos de formación docente; PD, como su nombre lo indica, fue para primaria y AC se lanzó para los tres niveles de la educación obligatoria. En segundo término, hay que valorar la filosofía que orientó la implementación de estas iniciativas. El PCI se orientó según la filosofía de las iniciativas OLPC (One Laptop Per Child, una computadora por chica/o), esto es, entregar sin devolución una computadora portátil para cada estudiante, incluso permitiendo su uso fuera de las escuelas. Las otras dos, en cambio, se implementaron mediante la distribución de Aulas Digitales Móviles (ADM), un kit de equipamiento destinado a cada escuela. Finalmente, cabe destacar que el equipamiento distribuido por AC refleja el cambio de perspectiva y la búsqueda de incluir contenidos vinculados a las Ciencias de la Computación en las aulas, incorporando robots educativos, además de computadoras portátiles y pizarras digitales).

Si bien el análisis de los resultados de estas políticas públicas exceden largamente los objetivos y el alcance de este trabajo, nos interesa señalar que las características diferenciadas que presentaron, contribuyeron a diferentes prácticas de integración entre los niveles educativos, tanto en estudiantes como en docentes.

La formación docente inicial que acompañó la distribución de equipamiento, también describe un comportamiento desigual según los niveles y asignaturas a las que atiende. Esto se expresa en los diseños curriculares vigentes en la actualidad. En el caso del profesorado de Educación Inicial, el diseño curricular (2014) dispuesto a partir de la cohorte de 2015 presenta los espacios de “Lenguaje Digital y Audiovisual” en 1er año y “TIC y la enseñanza en el Nivel Inicial” en 2do año. Similar es el caso del diseño curricular (2014) para el Profesorado de Educación Primaria, con “Lenguaje Digital y Audiovisual” en 1er año y “TIC y la enseñanza en el Nivel Primario”, en 3ero.

En el caso de la Educación Secundaria, el escenario es distinto. Por tomar algunos ejemplos, los diseños curriculares para los profesorados de Matemática (2015), Historia (2017) y Lengua y Literatura (2019) no contemplan ningún espacio curricular específico relacionado con la integración de la informática. Diferente es el caso del diseño del Profesorado de Educación Tecnológica (2019), que presenta “Lenguaje Digital y Audiovisual” en 2do año y “Tecnologías educativas”, en 3ero.

Estos ejemplos ilustran la notable diversidad de formación del colectivo docente respecto a estos contenidos al momento de comenzar a transitar este cambio de perspectiva.

En resumen, el sistema educativo de la provincia (de modo análogo al del resto del país), presenta una importante heterogeneidad, que es ineludible a la hora de considerar los desafíos de la formación docente actual para la enseñanza de Ciencias de la Computación.

En el marco de la aprobación de los Núcleos de Aprendizaje Prioritario de Educación Digital, Programación y Robótica (NAP-EDPR, 2018) la jurisdicción fue tomando definiciones direccionadas a la implementación del Plan Aprender Conectados en su territorio. El lanzamiento del Plan Leer, Escribir y Pensar en la Cultura Digital en 2021, aporta definiciones y finalidades formativas respecto a los contenidos priorizados por la provincia y se vincula con lo propuesto por los NAP-EDPR. Posteriormente, se publicó el material Actualización Curricular “Aportes de la cultura digital en la Educación Tecnológica” (2023), que propone la integración de saberes y conocimientos provenientes mayormente de las Ciencias de la Computación en diferentes espacios curriculares, en los distintos ciclos y niveles de la educación obligatoria.

En ese marco se diseña la experiencia de formación docente que socializamos en este escrito, denominada Ciclo de Seminarios “Ciencias de la Computación en la Escuela”, que el Ministerio de Educación de la Provincia de Córdoba ofrece, de manera virtual, a través del Instituto Superior de Estudios Pedagógicos.

Se trata de una propuesta integral, compuesta de dos seminarios y un seminario-taller, denominados:

- Seminario Ciencias de la Computación y su Enseñanza.
- Seminario - Taller de Herramientas Digitales.
- Seminario Análisis de Propuestas de Enseñanza.

El primer seminario presenta el enfoque general de todo el ciclo y por lo tanto, propone una introducción a cada una de las dimensiones que atraviesan la propuesta. El seminario-taller, se propone como un espacio de experimentación y aprendizaje específico del campo de las CC. El Seminario de Análisis de Propuestas de Enseñanza pone el énfasis en la enseñanza de estos contenidos, enfatizando en sus modos de abordaje y su didáctica específica.

Cada uno tiene una duración de 8 semanas y puede cursarse de manera independiente. La propuesta presenta la opción de un Seminario de Trabajo Final, que, sumado al cursado de los tres seminarios, permite acreditar la Actualización académica.

En los siguientes apartados se describe el enfoque tecno-pedagógico que orienta la propuesta, las características del dispositivo de formación, y las primeras conclusiones, que hemos podido construir en estos primeros meses de puesta en marcha de la iniciativa.

## **Enfoque tecno-pedagógico**

Enseñar, en parte, es la invitación que hacemos a los estudiantes a entrar en mundos no conocidos de la mano de los docentes. Implica un acto inventivo de poner a disposición de otros, saberes y experiencias que creemos vale la pena compartir porque hicieron sentido para nosotros como educadores. (Meirieu, 2005 en Coria y Basel, 2014) Esta invitación requiere reconocer contextos, culturas, valoraciones, saberes, intereses, disputas que existían antes de que llegemos con nuestra propuesta de contenidos. Enseñar no es un acto que se produzca en el vacío, ya que múltiples aspectos condicionan nuestra actividad docente y desafían el accionar de la escuela. En este sentido, reconocer desde qué perspectiva ético-política-pedagógica pensaremos y asumiremos la enseñanza implica necesariamente posicionarnos desde un lugar en el mundo. En este caso, además, la propuesta se diseña en el seno de una institución pública como lo es el ISEP, que sostiene un enfoque singular en relación al enseñar en general y a la formación docente en particular que permea y dialoga con el enfoque disciplinar y que a su vez sirve de base y de hilo conductor de toda la propuesta.

En busca de claves que nos permitan ir construyendo ese posicionamiento -tomando en cuenta el marco institucional y en conjunto con quienes cursan la propuesta de formación que aquí presentamos-, trabajamos en tres dimensiones que se entrelazan y que describimos a continuación. La primera problematiza la cultura digital y las tecnologías en el contexto global y educativo, la segunda, avanza sobre la disciplina, las Ciencias de la Computación, sus ramas y conceptos claves y, la última presenta algunos principios pedagógicos y didácticos que orientan las prácticas de enseñanza que proponemos. Cada una de estas dimensiones nos invitan a visitar conceptos y definiciones en la búsqueda de un lenguaje común para construir una voz propia, escolar, sobre por qué, para qué y cómo trabajar estos temas en educación.

## **Cultura digital y computación**

Una primera dimensión, problematiza la cultura digital y las tecnologías en el contexto general y educativo actual. Partimos para ello del concepto central: cultura digital porque articula nociones claves.

La propuesta asume como propia una definición del Ministerio de Educación de la Nación:

La llamada cultura digital hace referencia al conjunto de saberes y prácticas surgidas y desarrolladas en torno al uso cotidiano, ubicuo y masivo de tecnologías digitales, y particularmente, de Internet. Como usuarias y usuarios contribuimos cotidianamente a constituirla y —a la vez— se nos presenta como un conjunto de lógicas que las tecnologías digitales traen consigo (así como las tecnologías digitales no son neutras, la cultura digital tampoco lo es). (2021, p. 27)

La misma, resulta una idea potente porque alberga tres concepciones relevantes:

- *Que se trata de un conjunto de conocimientos y prácticas generadas en nuestras interacciones con las tecnologías computacionales.* Esto implica reconocer que cualquier abordaje de cultura digital no se reduce a un dispositivo o una red de conexiones, sino que debe incorporar análisis, reflexiones y, por supuesto, aprendizajes vinculados a las posibilidades y limitaciones que estas tecnologías nos presentan
- *Que contribuimos a su construcción de manera cotidiana.* Por ello, no debe aceptarse como dada e inamovible, sino todo lo contrario. Es una cuestión clave a la hora de valorar el papel de la escuela, como proveedora de saberes y herramientas con las cuales intervenir activamente en la construcción de esa cultura digital.
- *Que las tecnologías no son neutrales.* Quizás uno de los debates más necesarios y menos realizados en el mundo escolar. Entender que los artefactos y los programas que los hacen funcionar son obra de personas, grupos, organizaciones, orientadas por intereses, miradas, escalas de valores, pone en valor la necesidad de adquirir conocimientos sobre el diseño y funcionamiento (y no sólo sobre el uso) de dispositivos, redes y software que se integran a las aulas y/o se usan fuera de ellas.

La introducción a estos debates se realiza en el Seminario de Ciencias de la Computación y su Enseñanza y se realiza mediante la visita a materiales audiovisuales con aportes de Sztajnszrajber (2019) y Dussel (2022), respecto al papel de la escuela en el mundo actual; así como de las contribuciones de Bijker (2005) y Sandrone (2020) sobre la mirada determinista-hegemónica- y la necesidad nuevos puntos de vista para analizar las tecnologías. En los siguientes seminarios, estos debates funcionan como marco general que encuadra los contenidos específicos referidos al campo disciplinar y a su enseñanza. Enunciar esta primera dimensión como contexto y soporte teórico- analítico de toda la propuesta, la distingue de otras, cuyo énfasis está puesto en el desarrollo de una relación utilitaria con las tecnologías computacionales. Propuestas en las que se pondera su cualidad de herramientas dejando de lado su condición de producción humana y como tal, atravesadas por disputas y tensiones propias del campo de lo social.

### **Enseñar CC en una escuela comprometida frente a las brechas digitales**

Recuperando la función democratizadora e igualitaria de la escuela; en tanto institución que "... ofrece tiempo libre que transforma los conocimientos y destrezas en bienes comunes y, por lo tanto, tiene el potencial para proporcionar a cada cual, independientemente de sus antecedentes, de su aptitud o de su talento natural, el tiempo y el espacio para abandonar su entorno conocido, para alzarse sobre sí mismo y para renovar el mundo". (Simons y

Masschelein, 2014, p.12); no podemos dejar de reconocer las llamadas brechas digitales en el contexto de la enseñanza de la CC como una característica sostenida en el actual escenario, a nivel global y más particularmente, en las comunidades en la que la escuela actúa. Brechas digitales que reflejan y actualizan brechas sociales, económicas, geográficas, etarias y de género que se evidencian en el acceso desigual a dispositivos, conectividad y conocimientos significativos.

Actuando todas a la vez o por separado, estas brechas se materializan en nuestras aulas, entre nuestros estudiantes y sus familias. También en la mayoría de los docentes. Así se refleja en el Informe de la Relatora Especial sobre el Derecho a la Educación, presentado a Naciones Unidas en 2022:

La digitalización de la educación también conlleva graves riesgos para los derechos humanos, incluido el derecho a la educación. Algunos riesgos son exactamente lo contrario de los beneficios potenciales: aumento de la exclusión en lugar de mejora del acceso, uniformidad en lugar de enseñanza personalizada, aumento de los estereotipos en lugar de diversidad, reducción de la autonomía y la libertad en lugar de creatividad y participación, y minería de datos en beneficio de unos pocos en contra del interés público. (2022, punto 50, p.11)

La escuela, su proyecto de formación para una ciudadanía plena y participativa, tiene que dimensionar adecuadamente el componente digital de la vida humana. Y, en ese marco, poner a disposición conceptos, habilidades y herramientas que contribuyan a comprender y modificar el contexto actual y aporten a la construcción de sociedades más justas.

Buscando caminar hacia este horizonte, la propuesta aporta en su perspectiva tres claves para pensar la enseñanza de Ciencias de la Computación en la escuela:

- *Concebirla como parte de la formación integral* de cada estudiante, desde la primera infancia. Apostar a que, lejos de convalidar esa realidad, la escuela haga un aporte significativo a la construcción de una cultura digital más justa, democrática e inclusiva.
- *Apuntar a una alfabetización digital desde un enfoque crítico*. En consonancia con lo que decíamos en el apartado anterior, esto implica, además de poner a disposición saberes vinculados a las Ciencias de la Computación, brindar elementos que contextualicen la inserción de las tecnologías computacionales con mayor presencia y exploren en qué medida su diseño y funcionamiento colaboran la profundización de las desigualdades existentes. Acompañando esta lectura del mundo, con una búsqueda emancipadora, que visibilice desarrollos tecnológicos alternativos, que

apuestan a prácticas transformadoras en nuestras comunidades.

- *Una tarea integral, de todo el equipo docente.* Afrontar una tarea de tales características y alcances trae aparejada la articulación de los contenidos de Ciencias de la Computación con todas aquellas cuestiones vinculadas al contexto sociocultural e histórico, así como a las necesidades e intereses de la comunidad en la que actúa la escuela. Esto no puede depender exclusivamente de cada área/docente de informática.

### **Las CC en la comprensión e intervención en el mundo actual**

Partiendo de esta noción de cultura digital y de las claves para pensar la enseñanza en el contexto actual, la siguiente dimensión, incorpora conocimientos disciplinares, desde una propuesta que implica “abrir la caja negra” de las tecnologías computacionales para saber lo que pasa en su interior, qué capacidad de acción tienen, qué nos hacen hacer. En ese marco se abordaron conceptos claves de las Ciencias de la Computación.

La forma de tratamiento de este contenido, en sintonía con los principios didácticos que sostenemos, propone en el primero de los seminarios la interacción, a partir de una propuesta lúdica, con un sistema de IA -específicamente una red neuronal- en la que éste debe “adivinar” a qué objeto corresponden los dibujos que hace el cursante en la pantalla. Se propone un experiencia con esta tecnología más asociada a los usos cotidianos que a aquellos que ocurren en ámbitos de investigación y/o desarrollo avanzados.

Se acerca a los cursantes preguntas apuntadas directamente a las razones que permiten que el sistema realice tales “adivinaciones”. Esta estrategia nos permite presentar y describir, tomando a Turner (2018), las tres ramas de las Ciencias de la Computación

-máquinas, informática teórica, programación- y los contenidos asociados a ellas. Este recorrido, distingue a la propuesta de otras, en tanto que pone el énfasis en ir reconstruyendo el campo disciplinar específico a partir de una experiencia concreta. Esto permite diferenciarlo, caracterizarlo tomando como punto de partida ideas previas e hipótesis de los y las cursantes, permitiéndoles advertir, a partir del recorrido de las clases, la importancia de contar con estos saberes, para poder comprender la manera en que interactuamos con las tecnologías y los modos en que habitamos un mundo contemporáneo, que se encuentra atravesado por ellas. Tenemos en claro que enseñar a programar, cómo funciona un algoritmo de búsqueda o cómo funciona una computadora, por nombrar algunos contenidos dentro de esta área, no aseguran por sí mismo que cada estudiante logre complejizar su mirada sobre el mundo que lo rodea o el lugar que ocupa en él. Pero no tenerlos, constituye una limitación estructural para que puedan conseguirlo y, también muy importante, para que puedan generar y/o demandar soluciones

computacionales situadas.

Por esto mismo, la profundidad y modo de abordaje de cada tema busca brindarles a los cursantes más posibilidades de intervenir en la construcción de la cultura digital y a su vez, abordar estrategias didácticas que permiten enseñar estos contenidos a partir de la experiencia y no de la mera enunciación teórica. Decimos intervenir, pensando en dos planos: por un lado, en la creación de soluciones computacionales afines a las demandas y visiones compartidas por cada comunidad; por otro, como condición necesaria para participar lo más activamente posible en la disputa por posibles diseños de hardware y software orientados a la soberanía tecnológica.

Estas perspectivas se retoman en el Seminario-Taller de Herramientas Digitales, poniendo el énfasis en contenidos propios de las Ciencias de la Computación que, en sintonía con “abrir la caja negra” permita profundizar la comprensión del mundo digital actual.

### **Hacer foco en la enseñanza de las CC**

Con esas claves circunscribiendo nuestro planteo, nos adentramos en la tercera dimensión, la pedagógico-didáctica. Aquella que orientan las prácticas de enseñanza que proponemos, haciendo foco en la enseñanza de las Ciencias de la Computación. En el primero de los Seminarios, nos aproximamos a esta dimensión desde un breve recorrido histórico sobre las perspectivas que orientaron la integración de computadoras en la escuela. El siguiente paso es la lectura de documentos curriculares actuales, tanto nacionales como provinciales, a fin de comprender el marco prescriptivo desde el cual se está pensando la enseñanza de estos temas y cuáles son los requerimientos actuales.

Este camino continúa con la indagación de las perspectivas o tendencias didácticas actuales que, en distintos aspectos, guardan coherencia con el enfoque propuesto también en el primero de los Seminarios y que describimos en los dos apartados anteriores. Así como se hizo antes con la disciplina y la experiencia de una red neuronal, aquí se introduce el tema de la enseñanza también desde la revisión y análisis de casos concretos. Se analizan dos manuales que abordan la enseñanza del tema “la computadora”: el de que publicó la organización Ondula (2015) y el editado por la Fundación Sadosky (2019). Buceamos en ellos en busca de pistas sobre cómo se concibe a la tecnología en el marco de la cultura digital, qué estrategias y actividades se proponen, con qué criterios se eligen los materiales y recursos, entre otras cuestiones. Esto es, reflexionar sobre cómo el enfoque y los principios pedagógicos y didácticos que los sostienen influyen en los modos de abordaje de los contenidos y en las formas de vinculación con el conocimiento que se propician en los estudiantes.

Este análisis, tiene como propósito visibilizar cuestiones cruciales que deben incluirse en la

formación docente y, a partir de ella, en las prácticas de enseñanza. Teniendo en cuenta esto, en los materiales se buscó distinguir:

- A quiénes está dirigido el material, el grado de reconocimiento y valoración sobre sus contextos particulares y diversos.
- El tipo de comunicación, su claridad y la recuperación de la cultura local.
- El lugar que tiene el contexto (histórico, sociopolítico, etc.) en el abordaje del tema de estudio.
- La perspectiva (crítica, descriptiva, adaptativa) con que se trabaja el tema.
- Las metodologías de enseñanza que utilizan (puramente teóricas, experienciales, de diálogo, artísticas, de resolución de desafíos)
- La modalidad de acercarse al contenido (desenchufadas, físicas, con dispositivos)
- El tratamiento del tema (espiralado, diversificado, etc.)
- Los tipos de materiales y recursos (libres o privativos, abiertos)

Finalmente, teniendo en cuenta el enfoque crítico-conceptual, el disciplinar y los aportes del análisis de los materiales, se recuperan y presentan una serie de principios y criterios pedagógico-didácticos para el diseño de propuestas de enseñanza sobre Ciencias de la Computación. Algunos de ellos son:

- *Promover la participación activa, la resolución de problemas, la colaboración y experimentación.* Trabajo con problemas, proyectos y casos, así como propuestas que impliquen la experimentación con artefactos tecnológicos/ciberfísicos (hardware y software), con distinta complejidad según edades y niveles
- *Proponer actividades de “piso bajo, techo alto y paredes anchas”.* Esto es, la posibilidad de realizar proyectos sencillos (piso bajo), que puedan complejizarse significativamente (techo alto) y articularse con otras creaciones (paredes anchas), (Martínez y Echeveste, 2018).
- *Trabajar con el error desde un punto de vista constructivo.* La posibilidad de diversos modos de testeo sobre las creaciones, para su posterior refinamiento y/o modificación.
- *Acompañar la conversación pedagógica.* Sosteniendo, desafiando a cada estudiante, para ir avanzando en nivel de complejidad en el tratamiento del contenido, en los desafíos cognitivos puestos en juego.

A partir de estos principios, en el Seminario-Taller Herramientas Digitales se pone el énfasis en ofrecen espacios de experimentación sobre conceptos básicos de programación, proponiendo en cada una de las clases actividades concretas, pequeños desafíos y problemas en los que deberán reconocer y aplicar conceptos y prácticas computacionales para su resolución.

El abordaje de estos contenidos inicia con situaciones de la vida cotidiana en la que es posible distinguirlos. Esta estrategia es adecuada tanto para la apropiación conceptual, como para la incorporación de la “buena práctica” de bocetar, escribir notas, diseñar algoritmos en papel antes de ir al teclado. No menos importante, sirve como una alternativa concreta de iniciación en la enseñanza de esta temática en instituciones con acceso nulo o muy limitado a computadoras y conectividad. Por estas razones, las primeras actividades que propone el Seminario son “desenchufadas”. Esto es, utilizando lápiz y papel, o algún cuaderno de apuntes, en el que ir completando lo que la consigna de trabajo plantea.

Tras esta aproximación, se proponen actividades de experimentación con entornos de programación educativa desarrollados como software libre y de uso extendido en nuestro sistema educativo: Pilas Bloques y Scratch. Se inicia con Pilas Bloques, que es un entorno de programación cerrado, para luego cambiar a Scratch, una propuesta abierta.

Por último, en el Seminario Análisis de Propuestas de Enseñanza se retoma la dimensión pedagógico-didáctica, poniendo el énfasis en el abordaje de contenidos de CC en diferentes secuencias didácticas del sitio del Ministerio de Educación provincial [Hacemos Escuela](#). De este modo, se complejiza el ejercicio propuesto en el primero de los seminarios. Se retoma la misma lógica que se ha venido sosteniendo durante los otros dos Seminarios a partir de ejemplos y casos, habilitando un punto de partida concreto para abrir el juego a los docentes para que imaginen propuestas áulicas propias y situadas.

Tanto el primer y el segundo seminario, se trabaja con las mismas clases para docentes tanto de nivel primario como secundario. Pero en este último seminario, se ofrece un conjunto de secuencias diferenciadas por niveles de manera tal que los y las cursantes puedan seleccionar aquellas que son afines a su práctica cotidiana y contextualizar el análisis que se propone en el marco del nivel concreto.

Cada una de las decisiones que han dado forma a esta iniciativa, se corresponden con reconocer que su implementación se da en un contexto de desigualdades e injusticias. Y por ello, urge fortalecer las posibilidades que puede ofrecer la escuela para contribuir a la generación de vocaciones tecnológicas que se orienten a construir una cultura digital inclusiva, democrática y abierta.

### **Características del dispositivo de formación**

Como se dijo antes, la propuesta se piensa para un universo docente de los niveles primario y secundario de la Provincia de Córdoba. Se destina específicamente a quienes están al frente del espacio de [5ta hora](#) (jornada extendida en nivel primario, primer ciclo) y a docentes de la asignatura Educación Tecnológica (secundario, ciclo básico). Esta primera edición, comenzó con 6 aulas, con 60 docentes en cada una de ellas.

El recorrido por los distintos momentos que ofrece el ciclo de seminarios apuesta a la formación integral, brindando espacios, saberes y herramientas para que cada docente pueda iniciar la implementación de propuestas de enseñanza de CC en sus aulas, desde una perspectiva situada y crítica.

A continuación, se destacan algunas características esenciales de la implementación de la propuesta:

### **Equipos de trabajo**

El número de cursantes, así como la diversidad de instituciones en las que desarrollan sus tareas docentes implica un desafío al que se responde con el trabajo coordinado de distintos equipos. La escritura de las clases y las orientaciones para su cursada está en manos del Departamento de Enseñanza en Tecnologías Digitales e Informática y el equipo del Programa Cultura Digital. Las clases que luego se alojan en el campus virtual en el que se cursan los seminarios, se producen coordinando además con el Área de Producción de Materiales de ISEP, que acompaña el proceso de escritura con sugerencias, producción de recursos, diseño y maquetación del contenido en las aulas virtuales. “Así, las clases están conformadas por una pluralidad de voces, de “decires” que comunican de manera multimodal. En este sentido, decimos que las clases son una obra -nos gusta la imagen del artesano de Sennet (2009)- que resulta de una producción singular y colectiva a la vez”. (Equipo ISEP, 2020). Ya en el terreno de la implementación, la Coordinación del Ciclo de Seminarios Ciencias de Computación en la Escuela trabaja con un equipo de tutoras/es que asume la responsabilidad pedagógica en el recibimiento, apoyo frente a dificultades, acompañamiento virtual asincrónico y desarrollo de los encuentros sincrónicos. El trayecto de cada docente participante, es registrado y acompañado por el Área de Gestión de Cursada.

### **Clases y encuentros virtuales**

Lo primero que debe considerarse es que la propuesta se plantea para ser cursada de manera completamente virtual.

En ese marco, se organizan clases virtuales y encuentros sincrónicos. Las clases, entendidas como unidad de sentido, son asincrónicas. Duran dos o tres semanas y en ellas se trabajan

todos los contenidos propuestos en cada seminario. El campus virtual de ISEP -una plataforma digital- se convierte en el territorio en el que se plasma y desarrolla la propuesta de enseñanza. Es allí donde se da el diálogo pedagógico, donde ocurren las interacciones (Tarasow, 2014) mediante foros, chats, recursos interactivos, documentos en línea. En este espacio, la figura central son los y las tutores que hacen de puente entre la clase escrita y la instancia de cursada.

Esta actividad se integra y complementa con la realización de encuentros sincrónicos, en los que se recuperan contenidos, se revisan y/o realizan actividades y se evacúan dudas relacionadas con los temas trabajados. Las orientaciones para la realización de estos encuentros resultan del intercambio entre la Coordinación General y el Departamento de Enseñanza en Tecnologías Digitales e Informática.

### **Actividades integradas y espiraladas**

Las actividades de las clases, se articulan con las de acreditación de distintas maneras, teniendo en cuenta el conjunto de temáticas propias de cada seminario. Para ejemplificar este aspecto, tomaremos como referencia lo que se propuso para el primero de los Seminarios. En éste, se realizaron tres actividades integradas, pensadas en torno al análisis de experiencias escolares que abordan algunos contenidos de Ciencias de la Computación. Estas experiencias eran seleccionadas por los/las docentes cursantes de un conjunto ofrecido al inicio de la cursada. Cada una de las actividades propuso focalizar en aspectos particulares de la experiencia elegida, recuperando el contenido y las reflexiones que se fueron haciendo a lo largo de cada clase. Juntas constituyeron un todo integrado.

La progresión de las actividades propone, en un primer momento que los y las docentes cursantes puedan comprender el enfoque sobre cultura digital y tecnologías digitales que se propone en el seminario, reconociendo en las experiencias la perspectiva de abordaje que le dieron los/las autores/as. Luego, se pide identificar los contenidos, habilidades y herramientas de las distintas áreas que se ponen en juego en la experiencia, utilizando además del contenido de la clase, un recurso que llamamos la “brújula informática”<sup>1</sup> que sintetiza aspectos centrales de la alfabetización digital y permite reconocer la especificidad de los contenidos propios de las CC, cuestión que suele ser compleja. Además se le propone enriquecer dichas experiencias con nuevos contenidos disciplinares. Finalmente se los invitó a profundizar en el abordaje metodológico, proponiendo actividades, tareas etc, que permita enriquecer las experiencias de los estudiantes desde nuevas perspectivas.

### **Cierre y primeras conclusiones**

Tal como se menciona al inicio del presente trabajo, la propuesta que se describe está en plena

implementación de su primera edición. Esto subraya, más que en otras circunstancias, lo provisorio de las conclusiones.

Sin embargo, la experiencia transitada y construida hasta el momento por el equipo a cargo, junto a la información disponible en las distintas aulas de cursado, arrojan algunas pistas sobre cuestiones que consideramos valiosas compartir y que aquí se recuperan.

En cuanto a los logros:

Podemos mencionar que la propuesta de clases, tanto sincrónica como asincrónica, ha sido valorada positivamente por quienes cursan, así como por el equipo de tutores y tutoras.

- Destacan el contenido preciso y claro, que posibilita la comprensión de conceptos que son complejos.
- Valoran la propuesta de actividades por tener una complejidad acorde a los saberes previos de los cursantes y propiciar el intercambio, aportando al grupo en general. Respecto a las actividades de acreditación obligatorias, al desarrollarse como unidad integradora, consideran que permiten comprender las distintas temáticas del seminario de manera articulada.
- Se reconoce adecuada la extensión de las clases considerando los tiempos previstos.

También se reconocen prácticas que vale la pena sostener en futuras implementaciones:

- Propiciar espacios que den lugar a mirar “con otros ojos” la experiencia cotidiana con artefactos ciber-físicos, de modo que genere en los docentes la necesidad de “...experimentar, explorar, informarse, interrogarse, preguntarse sobre las tecnologías digitales, desde su relevancia social, económica, política y cultural, echando mano a saberes propios de las Ciencias de la Computación, en diálogo con otros campos de conocimiento” (Pacheco, Torres y Pintos, 2023: 17)
- Ofrecer oportunidades para conocer propuestas de enseñanza diversas, modos de abordaje de los contenidos vinculados a las CC, que equipos de trabajo expertos y docentes comparten y experimentan a diario en distintas instituciones educativas, prácticas que se sostienen en los principios pedagógicos que hemos descripto. No para que sean tomadas como modelos rígidos a seguir, sino que sirvan de punto de partida concreto, trampolín que le permitan inventar, crear, poner a prueba dichas creaciones en sus aulas y luego, con el insumo de su propia experiencia, mejorar y complejizar sus propuestas.

- Alentar el trabajo colaborativo. Generar comunidad resulta fundamental cuando se trata de pensar en la articulación de los contenidos de Ciencias de la Computación con espacios curriculares que aborden y problematicen las tecnologías ciber-físicas en contextos sociocultural e históricos concretos teniendo en cuenta las necesidades e intereses de la comunidad en la que actúa la escuela.
- Seguir ofreciendo y desarrollando materiales y herramientas que posibiliten el reconocimiento de contenidos y prácticas propias de CC (como la brújula). Guías concretas que ayuden a los y las docentes a ubicarse en un campo nuevo como objeto de enseñanza, que integra múltiples dimensiones y además está en diálogo con múltiples disciplinas. Esta cuestión, en particular, fue valorada positivamente por los docentes.

Por otro lado, respecto a la dificultad que manifiestan los tutores en cuanto a la comprensión de algunos conceptos complejos, así como en los modos de trabajo propuestos a los cursantes, entendemos que habría que repensar algunas de las decisiones que se tomaron y:

- Proponer otros modos de acercamiento al concepto de no neutralidad de la tecnología, cuestión que por su novedad y complejidad no ha sido fácil de comprender para los cursantes. Por ejemplo, propuestas de enseñanza y/o experiencias de clase en que se incorpore esta temática.
- Revisar la propuesta de trabajo en grupos. Si bien el trabajo grupal es algo innegociable por la potencia de la colaboración y el diálogo en el aprendizaje, se presenta como un desafío cuando participan en las aulas, docentes de toda la provincia y de distintos niveles educativos. Por un lado, suele ser complejo crear grupos en los encuentros sincrónicos; por el otro, en las actividades asíncronas reúne a cursantes con distintos niveles de entrega de actividades previas. Al finalizar la implementación de los tres seminarios se planea evaluar, además, si es o no necesario proponer aulas distintivas para cada nivel educativo.

Algunas de las prácticas que deberían potenciarse:

- Teniendo en cuenta la importante participación que tuvieron los cursantes al compartir sus propias experiencias de aula, consideramos valioso incorporar más instancias para que se potencie este tipo de intercambio, así como su análisis y sugerencias entre colegas.
- Un aspecto preocupante es la falta de habilidades digitales básicas (uso aula virtual,

compartir docs, utilizar videoconferencias, etc.) por parte de los cursantes y esto se expresa en el desenvolvimiento durante la cursada. Por este motivo es necesario pensar y diseñar instancias previas que permitan a quienes deseen cursar, recibir acompañamiento particular en estas cuestiones, permitiéndoles así participar de la experiencia de cursado virtual, de la mejor manera posible.

Recuperar en estas líneas finales la valoración de la propuesta que la coordinación del ciclo, el equipo de tutores y los y las docentes cursantes han realizado, es sumamente significativo, ya que entendemos a la enseñanza como una práctica reflexiva, una práctica social (Edelstein, 2011) que se enriquece con los intercambios entre colegas, en este caso, en la conversación que pueda surgir de nuestra experiencia con la suya, haciendo visibles un conjunto de criterios que resultan de la síntesis de las decisiones que, como equipo, tenemos que tomar a diario. Permite, en este sentido, reconocer aciertos y/o imaginar otros caminos posibles que potencien las propuestas futuras.

### **Bibliografía**

- Alliaud, A. (2018). *Los artesanos de la enseñanza. Acerca de la formación de maestros con oficio*. Paidós.
- Bijker Wiebe, E. (2005). ¿Cómo y por qué es importante la tecnología? *Revista Redes*, 11(21), 19-53. Universidad Nacional de Quilmes.  
<https://www.redalyc.org/articulo.oa?id=90702101>
- Córdoba. Ministerio de Educación. Secretaría de Educación. Subsecretaría de Promoción de Igualdad y Calidad Educativa. (2015). *Diseños Curriculares Profesorado Educación Inicial - Profesorado de Educación Primaria*.  
[https://dges-cba.infed.edu.ar/sitio/curriculares/upload/Disenio\\_Curr\\_Primaria\\_Inicial\\_2015.pdf](https://dges-cba.infed.edu.ar/sitio/curriculares/upload/Disenio_Curr_Primaria_Inicial_2015.pdf)
- Córdoba. Ministerio de Educación. Secretaría de Educación. Dirección General de Educación Superior. (2015). *Diseño Curricular Profesorado de Educación Secundaria en Matemática*.  
[https://dges-cba.infed.edu.ar/sitio/curriculares/upload/DCurricular\\_Matem-2015.pdf](https://dges-cba.infed.edu.ar/sitio/curriculares/upload/DCurricular_Matem-2015.pdf)
- Córdoba. Ministerio de Educación. Secretaría de Educación. Dirección General de Educación Superior. (2017) *Diseño Curricular Profesorado de Educación Secundaria en Historia*.  
[https://www.google.com/url?q=https://dges-cba.infed.edu.ar/sitio/curriculares/upload/2018-PROFESORADO\\_DE\\_EDUCACION\\_SECUNDARIA\\_EN\\_HISTORIA.pdf&sa=D&source=docs&ust=1700856085572065&usg=AOvVaw1RTN0WnIF\\_YCttPazlec1\\_eH](https://www.google.com/url?q=https://dges-cba.infed.edu.ar/sitio/curriculares/upload/2018-PROFESORADO_DE_EDUCACION_SECUNDARIA_EN_HISTORIA.pdf&sa=D&source=docs&ust=1700856085572065&usg=AOvVaw1RTN0WnIF_YCttPazlec1_eH)
- Córdoba. Ministerio de Educación. Secretaría de Educación. Dirección General de Educación Superior. (2019) *Diseño Curricular Profesorado de Educación Secundaria en Lengua y Literatura*.  
[https://www.google.com/url?q=https://dges-cba.infed.edu.ar/sitio/wp-content/uploads/2019/05/24\\_de\\_Abril\\_2019\\_LENGUA\\_Y\\_LITERATURA.pdf&sa=D&source=docs&ust=1700856105482923&usg=AOvVaw3nlyUX1O\\_JLgyplgA16EfH](https://www.google.com/url?q=https://dges-cba.infed.edu.ar/sitio/wp-content/uploads/2019/05/24_de_Abril_2019_LENGUA_Y_LITERATURA.pdf&sa=D&source=docs&ust=1700856105482923&usg=AOvVaw3nlyUX1O_JLgyplgA16EfH)
- . (2019) *Diseño Curricular Profesorado de Educación Tecnológica*. Dirección General de

Educación Superior.

[https://www.google.com/url?q=https://dges-cba.infed.edu.ar/sitio/wp-content/uploads/2019/05/24\\_de\\_ABRIL\\_2019\\_ED\\_TECNOLOGICA.pdf&sa=D&source=docs&ust=1700856127023037&usq=AOvVaw3mscGHujZ5ZDsLfIB\\_SFP](https://www.google.com/url?q=https://dges-cba.infed.edu.ar/sitio/wp-content/uploads/2019/05/24_de_ABRIL_2019_ED_TECNOLOGICA.pdf&sa=D&source=docs&ust=1700856127023037&usq=AOvVaw3mscGHujZ5ZDsLfIB_SFP)

- Coria, A. y Basel, P. (2014) Clase 1: Enseñar, una práctica cultural. Prácticas de enseñanza con TIC. Especialización Docente en Educación primaria y TIC.
- Dussel, I. [Canal ISEP]. (23 de agosto de 2022). *Cultura digital como proceso sociocultural y escolar*. [Archivo de Video]. Youtube.  
<https://www.youtube.com/live/tdJLI9hGgls?feature=share&t=4168>
- Edelstein, G. (2011) *Formar y formarse en la enseñanza*. Bs. As. Paidós
- Fontana, A. y Lamelas, G. (2023). La clase frente al desafío de la sobreabundancia de recursos y materiales para enseñar. *Seminario de Formación Interna de Departamentos y Programas* [material inédito]. Córdoba: ISEP.
- Martínez, M. C., & Echeveste, M. E. (2018). Experiencias de programación en las escuelas. *Cuadernos De Educación*, 16(16).  
<https://revistas.unc.edu.ar/index.php/Cuadernos/article/view/22971>
- Meirieu, P. (2006) *Carta a un joven profesor. Por qué enseñar hoy*. Grao.
- --. (2016) *Recuperar la Pedagogía. De conceptos comunes a conceptos claves*. Paidós.
- Naciones Unidas. (2022). *Repercusiones de la digitalización de la educación en el derecho a la educación*. Informe de la Relatora Especial sobre el derecho a la educación, Koumbou Boly Barry. [A/HRC/50/32: Repercusiones de la digitalización de la educación en el derecho a la educación | OHCHR](https://www.ohchr.org/es/Document/I/HRC/50/32/Repercusiones-de-la-digitalización-de-la-educación-en-el-derecho-a-la-educación-OHCHR)
- Pacheco, V.; Torres, M.; Pintos, P.; y equipos de producción del ISEP. (2023). Clase 3: Enseñar Ciencias de la Computación. *Seminario: Ciencias de la Computación y su enseñanza*. Ciclo de Seminarios Ciencia de la Computación en la escuela. Para el Instituto Superior de Estudios Pedagógicos, Dirección General de Educación Superior, Ministerio de Educación de Córdoba.
- Simons, M. y Masschelein, J. (2014) *Defensa de la escuela. Una cuestión pública*. Miño y Dávila Editores.
- Tarasow, F. (2014) La Educación en línea ya está en edad de merecer en Schwartzman, G., Tarasow, F. y Trech, M. (Comps.) *De la Educación a Distancia a la Educación en Línea: aportes a un campo en construcción* (pp. 21 a 35). Homo Sapiens Ediciones / FLACSO Argentina.
- Torres, M.; Pintos, P.; Pacheco, V. y equipos de producción del ISEP. (2023). Clase 1: Contexto y enfoque. *Seminario: Ciencias de la Computación y su enseñanza*. Ciclo de Seminarios Ciencias de la Computación en la escuela. Para el Instituto Superior de Estudios Pedagógicos, Dirección General de Educación Superior, Ministerio de Educación de Córdoba.
- Torres, M.; Pintos, P.; Pacheco, V. y equipos de producción del ISEP. (2023). Clase 2: Campo disciplinar. *Seminario: Ciencias de la Computación y su enseñanza*. Ciclo de Seminarios

Ciencias de la Computación en la escuela. Para el Instituto Superior de Estudios Pedagógicos, Dirección General de Educación Superior, Ministerio de Educación de Córdoba.

Turner, R. y Turner, R. (2018). *Artefactos computacionales*. Springer Berlín Heidelberg.

## Resultados en el uso de ChatGPT para resolver ejercicios de Teoría de la Computación

Emanuel Irrazábal, Lourdes Romera, Rubén A. Bernal, Jeremías González  
{eirrazabal, lromera, rbernal, jegonzalez}@exa.unne.edu.ar

Grupo de Investigación en Calidad de Software  
FaCENA  
Universidad Nacional del Nordeste

### Resumen

ChatGPT ha revolucionado el uso de herramientas de Inteligencia Artificial tanto en el ámbito profesional como académico a la par que ha generado preocupación por su uso como la disminución del pensamiento crítico en los alumnos o profesores. El objetivo de este trabajo fue medir y discutir las capacidades de ChatGPT para dar respuesta a los ejercicios prácticos en el marco de una asignatura universitaria sobre Teoría de la Computación. Para ello se realizó un estudio empírico evaluando los resultados desde el enfoque disciplinar y desde la relación entre los alumnos y ChatGPT. Como resultado de ello, el porcentaje de respuestas correctas fue de un 66%, con mayores inconvenientes para la resolución de ejercicio con autómatas a pila y máquinas de Turing. Los alumnos consideraron que ChatGPT interpretó adecuadamente los enunciados, aunque hubo un sesgo en las respuestas por intentar darle la razón al usuario. Finalmente, los alumnos manifestaron que invirtieron más tiempo en el análisis crítico del enunciado y en la evaluación de las respuestas realizadas por la herramienta.

**Palabras clave:** ChatGPT, Experimento, Universidad, Teoría de la Computación.

### Introducción

La entidad OpenAI lanza GPT-3 en 2020, un modelo de aprendizaje basado en Inteligencia Artificial que genera texto examinando miles de millones de palabras utilizadas como datos de entrenamiento, con el fin de comprender las relaciones entre palabras y frases [1]. GPT-3 estuvo a la vanguardia de una revolución en la Inteligencia Artificial, provocando debates filosóficos sobre sus límites y una multitud de usos potenciales, como resumir textos legales, ayudar a los programadores o a los científicos a escribir sus publicaciones. En 2022 la misma entidad deriva una versión mejorada de GPT-3, llamada ChatGPT (siglas del inglés de Chat Generative Pre-Trained Transformer), optimizada para la interacción con el usuario a partir de una interfaz conversacional [2].

Para habilitar este tipo de interacción, OpenAI hizo uso del aprendizaje auto-regresivo a partir de la retroalimentación humana [3]. En esencia, el modelo se entrenó en conversaciones reales transcritas, luego aprendió a refinar sus respuestas en función de los comentarios de evaluadores humanos que calificaron la calidad de las respuestas en un entorno de aprendizaje

por refuerzo.

Este proceso resultó muy exitoso en la creación de una interfaz diseñada para entender el lenguaje natural y generar respuestas relevantes frente a las preguntas del usuario. Como consecuencia de ello, a los dos meses la herramienta ya contaba con 100 millones de usuarios [4], estimándose unos 13 millones de usuarios únicos diarios en la actualidad. Esta tecnología, por tanto, tiene el potencial de revolucionar varias de las actividades relacionadas con el ámbito educativo tales como la búsqueda de información, la respuesta a preguntas específicas en base a esta búsqueda, la argumentación de discusiones, la escritura de ensayos o reportes, la generación de código fuente o el cálculo de ejercicios matemáticos, lógicos y estadísticos. Sin embargo, existen varias preocupaciones asociadas con el uso de ChatGPT; estos incluyen posibles sesgos en las respuestas, preocupaciones sobre la pérdida de empleo por parte de los docentes o la no promoción de la creatividad y el pensamiento crítico de los alumnos, originados por el plagio inherente en el

uso de la herramienta [5-7].

Recientemente, en un periódico de alcance nacional se daba la pregunta: “¿Las universidades deben resistirse o deben adoptar la Inteligencia Artificial en los procesos de aprendizaje?” [8]. En efecto, ChatGPT todavía es una herramienta emergente por lo que muchos docentes y estudiantes universitarios no la han probado o explorado en profundidad. Para garantizar que la puedan conocer y utilizar correctamente, se debe brindar capacitación sobre sus características, la manera de evaluar la precisión y la información obtenida. En este sentido, los primeros experimentos con ChatGPT en las aulas dan cuenta de un abanico de sentimientos, desde el temor a que los estudiantes lo usaran para eludir el proceso de enseñanza y aprendizaje, hasta el entusiasmo por las nuevas posibilidades que provee [9].

Entre los riesgos evidenciados en el uso de esta tecnología están sus limitaciones intrínsecas ya que las respuestas pueden ser inexactas [10-12]. Asimismo, dada la conveniencia de usar ChatGPT, los estudiantes, especialmente aquellos que trabajan en tareas de última hora, pueden usar esta herramienta como un medio para crear su trabajo sin usar su pensamiento analítico y sus habilidades para tomar decisiones. Se ha argumentado que el uso de ChatGPT tiene un impacto perjudicial en el desarrollo de habilidades esenciales de los estudiantes, incluidas las habilidades de pensamiento crítico [6][13] habilidades de resolución de problemas [13] e imaginación, así como habilidades de investigación [14]. Dado que estas habilidades desempeñan un papel fundamental en el éxito académico y profesional de los estudiantes, el uso de ChatGPT puede tener otras consecuencias negativas para los estudiantes, como la falta de motivación o capacidad de emprender.

En este artículo el objetivo fue medir y discutir las capacidades de ChatGPT para dar respuesta a los ejercicios prácticos en el marco de una asignatura universitaria, en particular de un curso introductorio sobre Teoría de la Computación en la Universidad Nacional del Nordeste. Para ello se realizó un estudio empírico evaluando los resultados desde diferentes enfoques, con objeto de responder a las siguientes preguntas:

- Qué tan correctas fueron las respuestas de ChatGPT.
- Cómo han interactuado los alumnos con ChatGPT.

El artículo se encuentra dividido en cinco secciones de la siguiente manera: en la sección 1 se ha introducido el tema y los objetivos de investigación, en la sección 2 se presenta un breve resumen de algunos trabajos relacionados, en la sección 3 se describe el diseño del

experimento, en la sección 4 se presentan los resultados junto con la discusión y en la sección 5 se concluye el trabajo.

## 1 Trabajos Relacionados

La explosión de esta tecnología ha supuesto, también, una gran cantidad de trabajos académicos relacionados. Algunos se han focalizado en realizar experimentos con la herramienta tanto en entornos de laboratorio como frente al aula. Además de los mencionados en la sección anterior, algunos trabajos relacionados de utilidad para la discusión pueden ser los siguientes:

En [15] se sugiere que los educadores deberían permitir el uso de ChatGPT ya que es probable que los estudiantes lo hagan de igual manera. Por ello, el educador debe enfocarse en que los estudiantes demuestren un pensamiento más crítico en la evaluación de la información o que desarrollen y presenten nuevas ideas. Además, las habilidades de presentación son esenciales para el aprendizaje exitoso y para la defensa del trabajo, competencias necesarias para el mundo laboral real. Esto, por tanto, favorecerá el uso de la defensa oral como forma de evaluación estándar.

En este trabajo [16] se dio el cambio de rol entre estudiantes y profesores, oficiando ChatGPT de mediador. Como beneficios, el docente pudo intercambiar roles con los estudiantes para evaluar la autenticidad del aprendizaje, el pensamiento crítico y creativo, generar ejercicios prácticos o resultados que fueron evaluados por los estudiantes.

En [17] se destacan los beneficios potenciales de la integración de herramientas como ChatGPT en la educación en Ingeniería Química para ayudar a la resolución de problemas que aplican los conocimientos fundamentales dados en la carrera. Este enfoque, según los autores, promueve el pensamiento crítico, mejora las habilidades para resolver problemas y facilita una comprensión más profunda de las materias básicas. El experimento consistió en la asignación de proyectos de aplicación práctica a cada uno de los estudiantes. Los resultados dan cuenta de la accesibilidad en el uso de la herramienta y hacer más eficiente la resolución de los problemas, aunque es necesario que los alumnos y los profesores entiendan las limitaciones de esta tecnología.

En [18] los autores investigaron las capacidades matemáticas de ChatGPT probándolo en conjuntos de datos disponibles públicamente y concluyeron que, contrariamente a

5

otros informes positivos en los medios, las habilidades matemáticas de ChatGPT están significativamente por debajo de los de un estudiante graduado de matemáticas promedio. Los resultados mostraron que ChatGPT a menudo entiende la pregunta, pero no proporciona las soluciones correctas.

Finalmente, en [19] se discuten de manera sistemática los fallos de ChatGPT. Da ejemplos de estos y los categoriza, en particular habla sobre fallos en el razonamiento matemático y en aritmética. Este trabajo, además, resume una gran cantidad de experimentos en diferentes disciplinas que han utilizado algún tipo de set de preguntas para comprobar el rendimiento de ChatGPT en sus diferentes versiones.

## **2 Diseño del experimento**

La asignatura Teoría de la Computación se dicta en el cuarto año de la carrera Licenciatura en Sistemas de Información de la Facultad de Ciencias Exactas y Naturales y Agrimensura de la Universidad Nacional del Nordeste, en la ciudad de Corrientes. El cuerpo docente está formado por un profesor titular y un jefe de trabajos prácticos, ambos Doctores. Las cuatro grandes unidades prácticas de este espacio curricular son los lenguajes y gramáticas formales, los autómatas de estado finito, los autómatas a pila y las máquinas de Turing.

El curso se dicta en el primer cuatrimestre, con un promedio de entre 30 y 50 alumnos por año. El dictado del año 2023 inició en el mes de marzo terminando en el mes de junio; ha sido en este marco donde se ha llevado a cabo un análisis transversal y descriptivo trabajando con los 39 alumnos del curso utilizando la herramienta ChatGPT en su versión 3.5, gratuita.

El experimento consistió en proporcionar un ejercicio incluido en la guía de trabajos prácticos a cada uno de los alumnos al finalizar las clases de cada unidad temática. Se les propuso como objetivo que cada uno utilice ChatGPT para resolver este ejercicio. Como paso previo se realizó una breve capacitación simulando los pasos de la consigna. Como segunda instancia se pidió a los alumnos contestar un cuestionario en línea con el fin de identificar:

1. La capacidad de respuesta de la herramienta respecto de los enunciados de Teoría de la Computación.

2. La capacidad de modificación de la respuesta inicial de ChatGPT si la respuesta era incorrecta.

3. La percepción del estudiante sobre la habilidad de la herramienta para entender la explicación del propio estudiante cuando la respuesta era incorrecta.

La consigna dada a los estudiantes fue la siguiente:

1. Indique a ChatGPT que Usted le dará un enunciado sobre Teoría de la Computación, describa el tema en particular e introduzca el enunciado tal y como figura en la guía de trabajos prácticos. El ejercicio se le ha sido asignado en clase.

2. En paralelo, lleve una bitácora de lo que sucede.

3. Indique si la respuesta ha sido correcta en el primer intento.

4. Si no ha sido correcta, describa el motivo de manera general. En el cuestionario habrá diferentes opciones: el enunciado es confuso, el enunciado requiere de elementos gráficos difíciles de ser descritos con texto, el enunciado es correcto, pero es mala la interpretación del enunciado por parte de ChatGPT, otros.

5. Si la respuesta no es correcta localice el error y explíquelo brevemente a ChatGPT, sin indicarle la solución. Si el error se debe a una confusión del enunciado aclare este punto con ChatGPT.

6. Realice el punto anterior hasta 3 veces en una misma conversación e indique el resultado.

7. Si al cabo de la conversación el resultado ha sido incorrecto describa el motivo.

8. Ponga en la bitácora lo que piensa acerca de la comprensión de ChatGPT.

En la Fig. 1 se ve una captura del formulario donde se sistematizaron las respuestas de cada alumno.

7

**Cuestionario Uso CHATGPT**

A continuación encontrará una serie de preguntas respecto de la interacción con ChatGPT para la resolución del ejercicio práctico indicado por la asignatura.

Nombre:  [Cambiar de cuenta](#)

No compartido

\* Indica que la pregunta es obligatoria.

Elija su nombre de la lista \*

Indique si la respuesta ha sido correcta en el primer intento. \*

Si no ha sido completamente correcta la respuesta seleccione el motivo: \*

Resultado después de hasta tres interacciones para corregir a ChatGPT: \*

Correcto

**Fig. 1.** Formulario para capturar el resultado de la interacción alumno – ChatGPT para cada ejercicio.

### 3 Resultados y Discusión

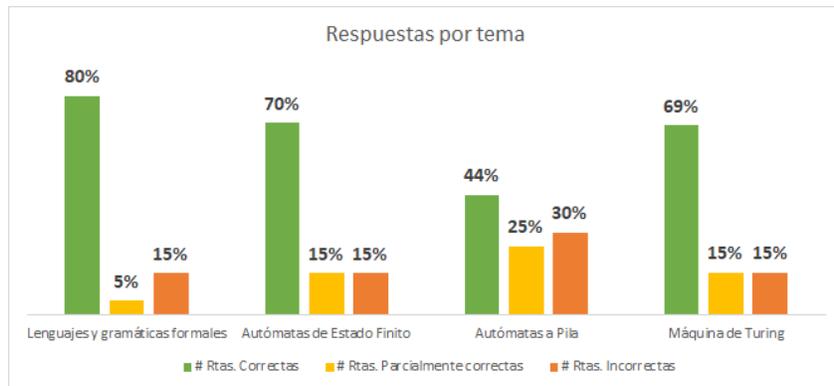
A continuación, se resumen y discuten las respuestas obtenidas por los alumnos, organizadas por objetivo:

#### 3.1 Primer objetivo: medir la capacidad de respuesta de la herramienta respecto de los enunciados de Teoría de la Computación.

En un primer intento de resolución, la herramienta ChatGPT ha sido capaz de resolver correctamente en promedio el 66% de los enunciados, de forma parcialmente correcta un 15% y no ha sido capaz de resolver un 19% de los ejercicios. Estos valores son bastante mayores a los que figuran en la literatura para otras disciplinas, que van desde el 25%

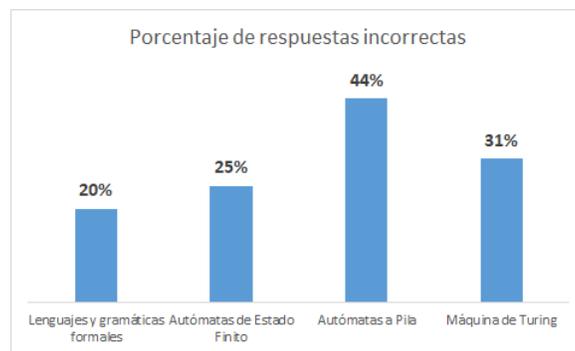
[18] al 55% [16]. En parte, esto se puede explicar por dos motivos: por un lado, el tipo y complejidad del ejercicio, más bien orientado a aspectos lógicos y de algoritmia; por otro lado, el set de ejercicios con los que se ha llevado a cabo el experimento no abarca los mismos casos que los sets de datos constituidos de manera formal y que son comunes en este tipo de experimentos.

En cuanto a la evaluación de la capacidad de resolución de ChatGPT por temática, se ha realizado un desglose de las mismas, como se puede observar en la Fig. 2.



**Fig. 1.** Categorización de las respuestas obtenidas en ChatGPT por parte de los alumnos, dividido por unidad temática.

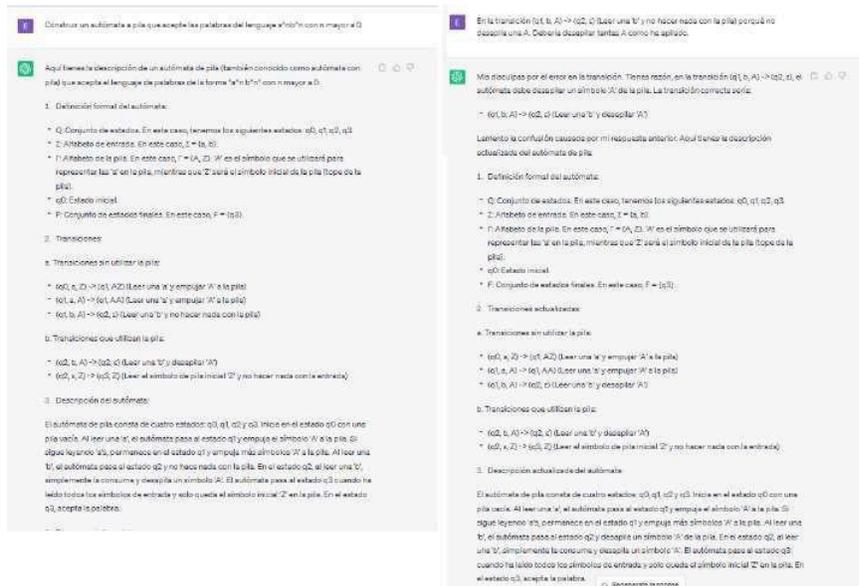
Los porcentajes de tres de las cuatro unidades temáticas son similares, en tanto que se puede ver un porcentaje superior en los ejercicios de autómatas a pila (ver la Fig. 3).



**Fig. 1.** Porcentaje de respuestas incorrectas por unidad temática.

En la Fig. 4 puede verse un resumen de una interacción con ChatGPT en un ejercicio canónico de autómata a pila. El ejercicio fue resuelto de forma incorrecta.

9



**Fig. 1.** Ejemplo de una conversación en ChatGPT para la resolución de un ejercicio de Autómata a Pila.

. El enunciado indicaba: “Construir un autómata a pila que acepte las palabras del lenguaje  $a^n b^n$  con  $n$  mayor a 0.”. La interpretación del mismo realizado por ChatGPT fue correcta, así como la estrategia de resolución.

En la Fig. 5 se presenta la tabla de transición de estados, el error fue la transición propuesta en la solución cuando se presenta el primer símbolo “b”. Esta transición no desapila ningún símbolo, esto hace que no se cumpla la consigna.

4. Diagrama de transiciones:

Estado	Entrada	Pila	Siguiente estado	Acción de la pila
q0	a	Z	q1	Empujar A
q1	a	A	q1	Empujar A
q1	b	A	q2	-
q2	b	A	q2	Desapilar A
q2	$\epsilon$	Z	q3	-

**Fig. 1.** Diagrama de transición de estados de la solución propuesta por ChatGPT.

La primera interacción intenta aclarar la duda por parte del alumno: “En la transición  $(q_1, b, A) \rightarrow (q_2, \epsilon)$  (Leer una 'b' y no hacer nada con la pila) por qué no desapila una A.

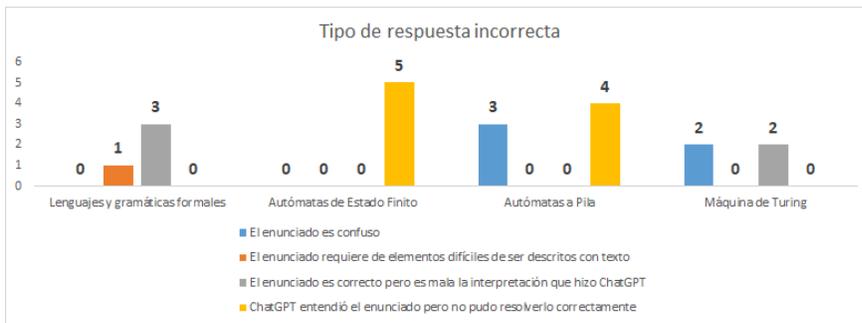
Debería desapilar tantas A como ha apilado.” Y ChatGPT reconoce el error describiendo la solución correcta, tanto en el cuerpo del enunciado como en el diagrama de transición (ver Fig. 6).

4. Diagrama de transiciones actualizado:

Estado	Entrada	Pila	Siguiente estado	Acción de la pila
q0	a	Z	q1	Empujar A
q1	a	A	q1	Empujar A
q1	b	A	q2	Desapilar A
q2	b	A	q2	Desapilar A
q2	ε	Z	q3	-

**Fig. 1.** Formulario para capturar el resultado de la interacción alumno – ChatGPT para cada ejercicio.

Respecto del análisis de las respuestas incorrectas (Fig. 7), la que más se presenta es la opción: “ChatGPT entendió el enunciado, pero no pudo resolverlo correctamente”, siendo la segunda que “el enunciado es correcto, pero es mala la interpretación que hizo ChatGPT”.



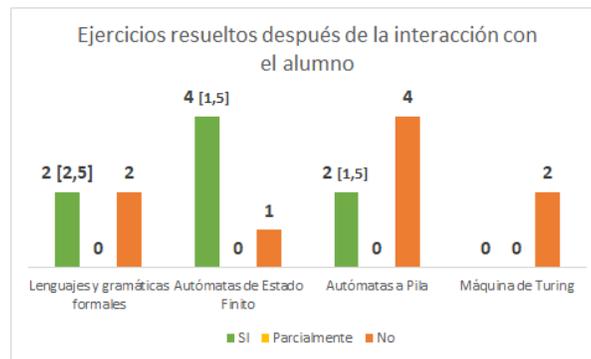
**Fig. 1.** Resumen de tipo de respuestas incorrectas por unidad temática.

**3.2 Segundo objetivo: medir la capacidad de modificación de la respuesta inicial de ChatGPT si la respuesta no era correcta.**

En la Fig. 8 se puede ver que 8 de los 17 ejercicios no resueltos pudieron ser reinterpretados y resueltos por ChatGPT con ayuda de los alumnos. Entre corchetes aparece el promedio de interacciones hasta lograrlo. Como observación de esto, nuevamente la temática de los autómatas a pila fue la que mantuvo más ejercicios no resueltos. En cuanto a

11

las máquinas de Turing, aún luego de la interacción realizada por los alumnos, las respuestas iniciales no pudieron ser modificados para obtener un resultado correcto. Estas dos temáticas, por otro lado, son las más avanzadas en cuanto a razonamiento lógico.



**Fig. 1.** Resumen de los resultados al intentar ayudar a ChatGPT a interpretar los ejercicios.

**3.3 Tercer objetivo: la percepción del estudiante sobre la habilidad de la herramienta para entender la explicación del propio estudiante cuando la respuesta era incorrecta.**

En la tabla 1 se resume la respuesta a la última pregunta del cuestionario acerca de cómo ha percibido la comprensión de ChatGPT respecto de los enunciados. Aquí puede verse un promedio alto en la percepción de los alumnos respecto de qué tanto ChatGPT comprendió los enunciados, manteniéndose la diferencia entre las temáticas iniciales y avanzadas.

**Tabla 1.** Qué tanto ChatGPT ha comprendido el enunciado. Respuesta de los alumnos.

---

Qué tanto ChatGPT ha	Lenguajes y gra-	Autómatas de	Autóma-	Máquinas
----------------------	------------------	--------------	---------	----------

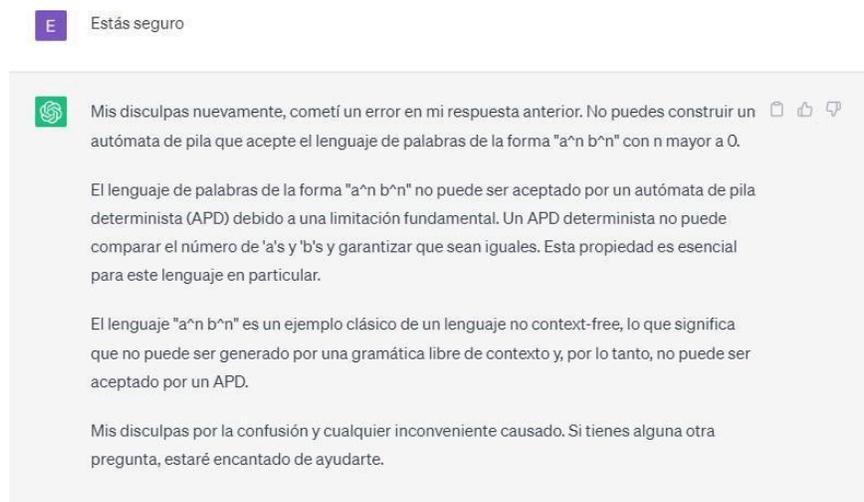
comprendido el enunciado	máticas formales	estado finito	tas a pila	de Turing
Muy buena	12	14	4	8
Buena	5	3	6	1
Regular	0	3	1	1
Mala	1	0	4	1
Muy mala	2	0	1	1

---

Promedio	4,2	4,6	3,5	3,8
----------	-----	-----	-----	-----

---

La elocuencia con la que ChatGPT reconoce el error puede prestar a confusión por parte de los usuarios. Una vez llevado adelante el experimento, se pudo regenerar la interacción de la Fig. 4. Llegados al punto donde la respuesta era correcta se hizo la pregunta: “Estás seguro” y la respuesta dada a continuación fue totalmente incorrecta (Fig. 9). Este es un ejemplo de cómo ChatGPT intenta conversar más allá de resolver los ejercicios y que las interacciones pueden generar confusión en las respuestas.



**Fig. 1.** Formulario para capturar el resultado de la interacción alumno – ChatGPT para cada ejercicio.

#### 4 Conclusiones

En este trabajo se realizó un experimento con alumnos de cuarto año que utilizaron ChatGPT para resolver ejercicios prácticos típicos de las diferentes unidades temáticas de la asignatura, con

el objetivo de cuantificar qué tan correctas fueron las respuestas a la vez que se medía la forma en la que los alumnos pudieron interactuar con la herramienta.

Respecto del primer objetivo, las respuestas fueron completas y correctas en la mayoría de los casos para todas las unidades temáticas, incluso en algunos casos en los que la descripción no era sencilla o la respuesta requería elementos gráficos. En esos casos

13

ChatGPT pudo elaborar la respuesta con notación matemática, lo cual fue provechoso para profundizar el conocimiento de los alumnos.

Se pudo observar una mayor cantidad de errores en el primer intento de respuesta para los ejercicios más avanzados: autómatas a pila y máquinas de Turing.

Respecto del segundo objetivo, es necesario dejar en claro que ChatGPT no resuelve los ejercicios como lo haría una herramienta especializada, por ejemplo, JFLAP [19]. Las respuestas necesitan ser valoradas por los usuarios con espíritu crítico.

Si bien las características conversacionales de ChatGPT ayudan al alumno a interactuar con el enunciado y las respuestas, dando otro enfoque al aprendizaje, en el coloquio final de este experimento, los alumnos manifestaron el desconcierto respecto de dos aspectos en la interacción con la herramienta: las respuestas convincentes pero incorrectas y el cambio sustancial de algunas respuestas con el fin de dar la razón al usuario.

En resumen, la utilización de este tipo de herramientas ha posibilitado otro enfoque en relación al proceso de enseñanza-aprendizaje del alumno, aumentando el análisis crítico del enunciado y la evaluación de las respuestas.

## Referencias

1. Dale, Robert. GPT-3: What's it good for?, *Natural Language Engineering*, 27(1), p. 113- 118 (2021).
2. ChatGPT, <https://openai.com/chatgpt>, last accessed: 20/05/2023
3. Hughes, Aaron, "CHATGPT: Everything you need to know about OpenAI's GPT-4 tool," *ChatGPT: Everything you need to know about OpenAI's GPT-4 upgrade — BBC Science Focus Magazine*, 16-Mar-2023. [Available: <https://www.sciencefocus.com/futuretechnology/gpt-3/>. Last accessed: 22-Mar-2023.
4. Reuters, <https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/>, last accessed 2023/06/17.
5. D'Amico RS, White TG, Shah HA, Langer DJ. I Asked a ChatGPT to Write an Editorial

About How We Can Incorporate Chatbots Into Neurosurgical Research and Patient Care....  
Neurosurgery. 2023 Apr 1;92(4):663-664. doi: 10.1227/ neu.0000000000002414. (2023)

6. Mhlanga, David, Open AI in Education, the Responsible and Ethical Use of ChatGPT Towards Lifelong Learning. at SSRN: <https://ssrn.com/abstract=4354422> or <http://dx.doi.org/10.2139/ssrn.4354422> (2023)
7. van Dis, E. A., Bollen, J., Zuidema, W., van Rooij, R., & Bockting, C. L.. ChatGPT: Five priorities for research. *Nature*, 614(7947), 224-226 (2023)
8. Clarín, [https://www.clarin.com/opinion/desafio-incorporar-chatgpt-universidades\\_0\\_vH2OfRQIFI.html](https://www.clarin.com/opinion/desafio-incorporar-chatgpt-universidades_0_vH2OfRQIFI.html) last accessed: 14/06/2023.
9. Sok, Sarin and Heng, Kimkong, ChatGPT for Education and Research: A Review of Benefits and Risks. Available at SSRN: <https://ssrn.com/abstract=4378735> or <http://dx.doi.org/10.2139/ssrn.4378735> (2023)
10. Baidoo-Anu, D., & Owusu Ansah, L. (2023). Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning
11. Gordijn, B., & Have, H. T. (2023). ChatGPT: Evolution or revolution? *Medicine, Health Care, and Philosophy*, 1-2
12. Qadir, J. (2022). Engineering education in the era of ChatGPT: Promise and pitfalls of generative AI for education. *TechRxiv*
13. Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., ... & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education.
14. Shiri, A. (2023). ChatGPT and academic integrity. *Information Matters*, 3(2), 1-5. <http://dx.doi.org/10.2139/ssrn.4360052>
15. Mohanad Halaweh, ChatGPT in education: Strategies for responsible implementation *Contemporary Educational Technology* 15(2), ep421 ISSN: 1309-517X (2023)
16. Jalil, S., Rafi, S., LaToza, T. D., Moran, K., & Lam, W. Chatgpt and software testing education: Promises & perils. In *2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pp. 4130-4137,. IEEE, (2023).
17. Tsai, Meng-Lin; Ong Chong Wei; Chen, Cheng-Liang. Exploring the use of large language models (LLMs) in chemical engineering education: Building core course problem models with Chat-GPT, *Education for Chemical Engineers*, 44(1), p. 71-95 (2023).

15

18. Frieder, Simon, et al. "Mathematical capabilities of chatgpt." arXiv preprint arXiv:2301.13867 2023.

19. Ali Borji, A Categorical Archive of ChatGPT Failures, Medium <https://medium.com/@ali-borji/a-categorical-archive-of-chatgpt-failures-2c888805d3c3> (2023)

20. JFLAP Homepage, <http://www.jflap.org>, last accessed /11/06/2023.

## **El Impacto en la Concepción Neutral de la Inteligencia Artificial de un Taller para Estudiantes de Escuela Secundaria**

Marcos J. Gómez<sup>1,2</sup>, Julián Dabbah<sup>1</sup>, Alfredo Sanzo<sup>1,3</sup>  
marcos.gomez@unc.edu.ar, jdabbah@fundacionsadosky.org.ar, alfredo.sanzo@gmail.com

<sup>1</sup>Fundación Sadosky - Iniciativa Program.AR

<sup>2</sup>Facultad de Matemática, Astronomía, Física y Computación  
Universidad Nacional de Córdoba

<sup>3</sup>Instituto de Ciencias de la Computación  
Universidad de Buenos Aires - CONICET

### **Resumen**

En este trabajo presentamos la implementación de un taller de IA para estudiantes de escuela secundaria que, mediante la creación de clasificadores de aprendizaje automático en una plataforma didáctica en línea, problematiza la neutralidad de la inteligencia artificial y pone de manifiesto el problema de los sesgos. Evaluamos su impacto y encontramos que impacta de forma estadísticamente significativa en opiniones sobre el sesgo de la IA, la neutralidad en el desarrollo de las aplicaciones de IA y en la superioridad de la IA sobre las personas en la resolución. En el trabajamos presentamos herramientas, diseño de estudios y resultados, para comenzar a discutir próximas líneas de acción.

**Palabras clave.** Sesgo de la Inteligencia Artificial. Escuela secundaria.

### **Introducción**

Las aplicaciones basadas en inteligencia artificial forman parte de nuestras vidas cada vez más y con mayor injerencia: cuando usamos filtros de imágenes "creativos" de las redes sociales o el sistema de recomendación de las plataformas de transmisión de contenidos, pero también cuando un banco utiliza un modelo de riesgo para otorgar o rechazar un crédito o un estado implementa un sistema de reconocimiento facial que puede determinar la detención de personas. Más aún, son cada vez más frecuentes y variadas las conversaciones en las que -con público general- nos asombramos y nos preocupamos por los resultados de la inteligencia artificial generativa.

Sin embargo, nuestras creencias y opiniones sobre estos desarrollos están más basados en réplicas de discursos ajenos que en razonamientos argumentados que consideren, aunque en su forma más básica, aspectos técnicos y coyunturales, pues no estamos aún en condiciones de hacer esto último. Para peor, los discursos sobre la IA más replicados suelen aportar casi exclusivamente una mirada alineada con los intereses de los dueños de estos desarrollos, en la que estas tecnologías se presentan como soluciones perfectas, superadoras e indiscutiblemente

neutrales.

Gran parte de los estudios existentes sobre experiencias de enseñanza de IA se preocupan por la incorporación de contenidos específicos de la disciplina. Para esto, utilizan entornos desarrollados para la enseñanza de estos contenidos para niños o adolescentes y evalúan la incorporación conceptual o el engagement de los participantes luego de la experiencia (*Tedre et al., 2021*). También existen, en menor medida, pocas experiencias que se proponen reflexionar sobre implicancias éticas y el impacto social, prestando atención a los sesgos que replican y profundizan las soluciones desarrolladas con IA.

Construir una mirada más amplia sobre estas tecnologías a partir de comprender su funcionamiento y de considerarlas en un contexto social, político y económico determinado se torna cada vez más importante para sostener una mirada crítica sobre el mundo tecnologizado que nos rodea. Esta construcción requiere, indudablemente, de la aparición de nuevos actores con otros discursos. En concreto, nos preguntamos *¿puede la creación y experimentación con modelos de IA -en entornos desarrollados para la enseñanza- enmarcada en una reflexión crítica impactar en las concepciones de los estudiantes para que incorporen una dimensión más crítica, menos neutral y tecno optimista?* En este trabajo evaluaremos el impacto de un taller para estudiantes que, mediante la creación de clasificadores de aprendizaje automático en una plataforma didáctica en línea, problematiza la neutralidad de la inteligencia artificial y pone de manifiesto el problema de los sesgos.

Para abordar la pregunta, definimos dimensiones vinculadas a las opiniones de los estudiantes sobre la IA. Para cada una de las dimensiones, planteamos hipótesis no direccionales para evaluar el impacto de la experiencia de haber realizado el taller en sus opiniones.

*“No existe diferencia estadística significativa en la opinión de los estudiantes antes y después de la participación del taller sobre...”*

*H1(nula) ...la influencia de las personas en los resultados de las aplicaciones de IA.”*

*H2(nula) ... la aplicación de la IA en cualquier ámbito.”*

*H3(nula) ... la posibilidad de la IA de cometer errores.”*

*H4(nula) ... la neutralidad de las aplicaciones con IA y la posibilidad que estas discriminen o prejuzguen.”*

*H5(nula) ...la superioridad de la IA frente a las personas para resolver problemas.”*

Para evaluar el impacto del taller, diseñamos un pre y un post test, en el cual los estudiantes

indicaron su grado de acuerdo o desacuerdo con cinco afirmaciones que reflejan las hipótesis planteadas y analizamos estadísticamente las diferencias entre las respuestas de ambas instancias. Además, les preguntamos sus percepciones sobre la experiencia para indagar aprendizaje autopercebido y satisfacción.

El artículo está organizado de la siguiente manera. En la Sección Trabajo previo, recuperamos trabajo previo vinculado a experiencias de enseñanza de IA en escuelas secundarias a nivel mundial. Luego, en Descripción del taller describimos el taller implementado, sus actividades y objetivos. En Diseño del estudio introducimos características generales de los grupos de estudiantes que participaron de la experiencia y las herramientas de recolección de datos. En Resultados presentamos y analizamos los resultados obtenidos a partir de las intervenciones, las posibles limitaciones del estudio y trabajo futuro.

### **Trabajo previo**

En los últimos años la enseñanza de la Inteligencia Artificial en las escuelas ha tomado fuerza (*Tedre et al., 2021*). Esto viene acompañado de la buena performance que han tenido la IA, para diferentes tareas, como los filtros de imagen o video, los sistemas de recomendación o más recientemente IAs generativas como Dall-E<sup>1</sup> o ChatGPT<sup>2</sup>.

*Tedre et al. (2021)*, realizan una revisión del estado del arte sobre las experiencias de enseñanza de contenidos vinculados a la IA en las escuelas en las que encontramos ciertos paralelismos con la enseñanza de la programación. Por ejemplo, cuando analizamos la enseñanza de la programación en contextos educativos reales, observamos que una de las primeras medidas es el desarrollo de entornos específicos (*Gómez, 2020*). Con respecto a la IA, podemos encontrar que en los últimos 5-10 años se han desarrollado entornos como Teachable Machine (*Carney et al., 2020*), Machine Learning for Kids (*Lane, 2021*), para abordar en la escuela nociones de aprendizaje automático (AA).

Otro ejemplo del paralelismo: la enseñanza de las CC muchas veces es confundida con enseñar el uso de las TICs. Con la IA sucede algo similar. Muchas experiencias que vinculan a la IA en las aulas, refieren al uso de herramientas desarrolladas con técnicas de IA, para resolver tareas puntuales. Por ejemplo, el usar ChatGPT como una herramienta auxiliar en el aula (*Zhai, 2023*), en clases de diversas disciplinas como física (*Bitzenbauer, P. 2023*), lengua (*Su et al. 2023*) y otras. En ese sentido, la IA se piensa como una herramienta para enseñar otras asignaturas, y no para la creación o comprensión de los conceptos fundamentales del área. En el caso de Teachable Machine y entornos similares, los estudiantes entrenan modelos de aprendizaje

---

<sup>1</sup> <https://openai.com/dall-e-2>

<sup>2</sup> <https://chat.openai.com>

automático, eligiendo los conjuntos de datos de entrenamiento y etiquetando los datos en clases específicas. Esto permite sacar el velo sobre parte de la disciplina de la IA (que el aprendizaje automático se basa en datos) y permite construir didáctica específica (como el taller descrito en el presente artículo).

Además del desarrollo de entornos para enseñar IA, también se han desarrollado propuestas curriculares que involucran contenidos de IA y aspectos éticos de la aplicación de la IA en diversos contextos (Lee et al., 2021; Miao & Shiokira, 2022; Dabbah et al., 2023). Las propuestas curriculares presentadas, describen alcances y contenidos a abordar de manera crítica, pero no cuentan con experiencias implementadas donde se pueda evaluar el impacto real de las creencias de los estudiantes sobre la IA.

### **Descripción del taller**

En esta sección describimos el taller implementado en las experiencias. En la subsección Propósitos del taller y perspectiva didáctica presentamos los motivos que llevaron al diseño e implementación del taller y detallamos los propósitos docentes de cada parte. En Actividades, describimos las actividades que formaron parte del taller.

### **Propósitos del taller y perspectiva didáctica**

La intención detrás del diseño del taller obedece a las preocupaciones que expresamos en la introducción: la construcción de nociones de IA desde una mirada crítica y localizada (en vez de neutral y solucionista). Ante la falta de propuestas de enseñanza que aborden nociones técnicas fundamentales a la vez que atiendan las inquietudes que estos sistemas motivan, el presente taller se propone ambos aspectos como objetivos específicos.

La primera parte del taller tiene como propósito poner de manifiesto que las computadoras son capaces de simular habilidades humanas a partir de disponer de un corpus adecuado de datos. En la segunda, el propósito es doble: por un lado, abordar nociones generales e indispensables para la creación de modelos de IA<sup>3</sup> y por otro, presentar a estas técnicas como muy poderosas pero susceptibles a sesgos y consecuencias humanas perjudiciales.

Las actividades del taller son mayormente prácticas e involucran el trabajo con datos y el uso de computadoras, junto con instancias de puestas en común para profundizar, contextualizar, relativizar o debatir hallazgos y preocupaciones surgidas durante las actividades. Estos momentos de reflexión acompañan el desarrollo y no se reducen a unos minutos al final, abonando a una construcción gradual, sostenida e integrada de la dimensión crítica. En esta misma línea, la

---

<sup>3</sup> No nos interesó problematizar la diferencia entre aprendizaje automático e inteligencia artificial; en el presente taller consideramos a la IA como una manera de resolver problemas automáticamente a partir del uso de volúmenes de datos.

intención es que esta construcción sea mucho más significativa a partir de que se desprenda de una experiencia concreta y una indagación activa, razón por la cual todas las actividades exponen a los estudiantes directamente con un problema.

Teniendo presente la brecha de género que existe en cuanto al acceso y apropiación de saberes de Computación, fue una preocupación del taller que interpele, incluya y pueda ser aprovechado por todos por igual. Para eso, durante el diseño se eligieron temáticas que no estén asociadas ni refuercen estereotipos de género y durante el dictado se alentó especialmente la participación de las estudiantes mujeres en clase.

### **Actividades**

En la primera actividad del taller, propusimos a los estudiantes simular un predictor de texto. Para esto, planteamos el problema como el desafío de adivinar cuál era la palabra que seguía en una oración incompleta, a medida que el tallerista iba revelando sus letras. La actividad contemplaba tres instancias, para contrastar la forma en la que las personas resuelven problemas "típicos" de IA y cómo intentan imitar esta solución las aplicaciones basadas en datos. En la primera instancia, los estudiantes respondían utilizando su intuición. En este caso, la predicción era muy fácil de hacer para las personas. Para una segunda instancia, les brindamos un corpus de un millón de líneas de subtítulos de películas y les pedimos que elaboraran una predicción basándose en búsquedas en el corpus. Como continuación hacia la automatización, les brindamos un archivo con el corpus organizado en bigramas con su frecuencia y restringimos a que la predicción se realizara íntegramente a partir del archivo y con criterios de selección efectivos y bien definidos (por ejemplo, la segunda palabra del bigrama más frecuente encontrado). Esta última instancia permitía experimentar una automatización casi completa con resultados aceptables.

Como cierre de la actividad, comparamos las tres instancias para resaltar que el problema de anticipar texto es un problema que las personas resolvemos muy fácilmente apelando a nuestra intuición (tanto que puede ser difícil explicitar los pasos que seguimos y, por ende, construir un programa) pero que se puede resolver de manera aceptable sin requerir de una persona inteligente, contando con un conjunto de datos suficientemente grande y procesamiento estadístico adecuado. Además, aprovechamos para señalar cómo la IA está mucho más presente en nuestra vida de lo que pensamos y ofrecimos a los estudiantes una experiencia de primera mano que involucraba análisis y procesamiento de datos para resolver un problema tipo.

Para la segunda actividad trabajamos en la plataforma TeachableMachine. Para aproximarnos a la plataforma y al tipo de problema que íbamos a resolver, les propusimos a los estudiantes que experimenten con un modelo existente, capaz de identificar a partir de una fotografía si un perro tiene pelo largo o pelo corto. El próximo paso era trabajar sobre un modelo incompleto para

familiarizarse con la creación de modelos en la plataforma (tanto para las actividades siguientes del taller como para usarla autónomamente en el futuro) y también conocer algunas ideas clave en el desarrollo de aplicaciones de aprendizaje automático, como *clase* o *categoría*, *datos de entrenamiento* y *entrenamiento*. En este caso, el desafío era construir un clasificador que indique en qué lugar (departamento, casa con patio o campo) convenía tener de mascota a un animal determinado. La primera tarea consistía en agregar la categoría *campo* con sus correspondientes datos de entrenamiento y entrenar el modelo.

A continuación, aparecía otro aspecto relevante en el desarrollo de aplicaciones con IA: la validación. Para esto, brindamos a los estudiantes un conjunto de imágenes para que prueben el modelo que acababan de crear, con el cual el modelo parece funcionar correctamente. Sin embargo, ponemos de manifiesto que no podemos saber cómo está funcionando ni en qué está basando sus decisiones.

En línea con esta última observación, proveímos a los estudiantes un nuevo conjunto de imágenes de prueba, especialmente elegidos para poner de manifiesto un sesgo marcado en el conjunto de datos de entrenamiento. Intencionalmente, los datos de entrenamiento de cada categoría estaban fuertemente sesgados por color como podemos observar en la Figura 1.

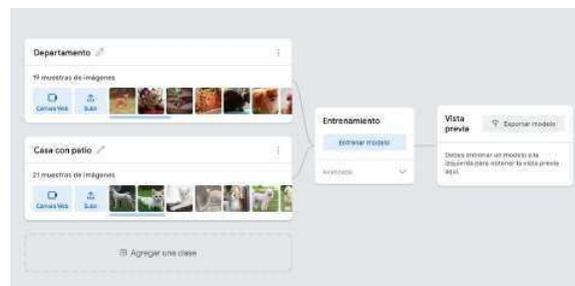


Figura 1. Modelo sesgado utilizado por los estudiantes. Los ejemplos de animales de departamento eran mayormente marrones, los de casa con patio, blancos y los de campo, negros.

Los estudiantes desconocían esta intención (al menos, no se la transmitimos explícitamente), y, por lo tanto, pudimos poner de manifiesto el problema pidiéndoles que utilicen el modelo para predecir el hogar para una vaca marrón: el modelo respondía con altísima certeza que el lugar más adecuado era un departamento. Encontrarse de primera mano con este problema tenía la intención de motivar la experimentación y volver al modelo creado con una mirada más crítica. En esta instancia, la mayoría de los grupos identificaron el problema por sí mismos y se realizó una puesta en común para compartir las experiencias y los hallazgos. A continuación, les estudiantes disponían de tiempo para buscar nuevas imágenes y agregarlas al modelo para mitigar el sesgo.

Como cierre de esta parte y de todo el taller se abordan dos temáticas: por un lado, motivar a

quiénes estuvieran interesadas a ponerse en contacto con la universidad e incentivarles a que continúen su formación, insistiendo en desarmar los mitos de que se tratan de disciplinas muy duras y que las carreras requieren muchos conocimientos previos. Y además, resaltar la importancia de participar de los desarrollos de este tipo de aplicaciones y de hacerlo con una mirada lo más amplia, localizada y diversa posible para evitar errores como los del clasificador de mascotas. Para reforzar esta idea, se relataron algunos casos existentes en los que, por un conjunto de entrenamiento mal diseñado, se crearon modelos de aprendizaje automático que, al ponerse en funcionamiento, generaban acciones discriminatorias sistemáticamente (un clasificador de CVs de una empresa de tecnología que, al estar entrenado con el historial de contrataciones, descartaba sistemáticamente las postulaciones de mujeres, o un detector de sueño para automóviles que fallaba para las personas con rasgos orientales).

### **Diseño del estudio**

En esta sección describimos el estudio diseñado para evaluar las hipótesis presentadas en la introducción. Primero presentamos el contexto en el cual fue implementado el taller y características de los estudiantes participantes de la experiencia. Luego, describimos las herramientas de recolección de datos utilizados.

### **Contexto del estudio**

El taller fue implementado en Agosto de 2022 como parte de las Jornadas Argentinas de la Didáctica de las Ciencias de la Computación (JADiCC) en la ciudad de Corrientes, Argentina. Estas jornadas, propiciadas por la Iniciativa Program.AR de la Fundación Sadosky, tienen la finalidad de favorecer el encuentro de investigadores, docentes, estudiantes de grado y posgrado, personas que forman parte de la gestión que lleven adelante proyectos de investigación o de intervención educativa en la enseñanza de las Ciencias de la Computación. Las JADiCC se organizan anualmente desde el 2018 buscando promover la incorporación de las CC en todos los niveles del sistema educativo argentino.

En esta primera edición del taller participaron alrededor de 60 estudiantes de escuelas secundarias de cuarto a sexto año de la provincia de Corrientes, de entre 16 y 18 años. Para conocer con más detalle datos demográficos de los estudiantes, completaron una encuesta inicial de la que obtuvimos un total de 41 respuestas. En la Tabla 1.a, podemos ver la distribución de los estudiantes por género. Un estudiante no completó la pregunta relacionada a género. En la Tabla 1.b podemos ver la distribución de los participantes por curso.

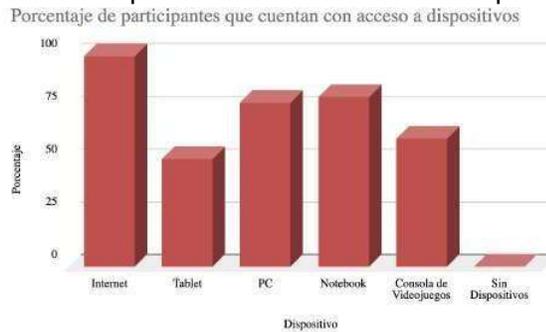
Género	Masculino	Femenino	Otro
Total	18	21	1

Tabla 1.a. Distribución de los participantes por género.

Curso	4to	5to	6to
Total	3	1	37

Tabla 1.b. Distribución de los participantes por curso.

Además de recolectar datos demográficos, los estudiantes completaron preguntas vinculadas al acceso internet y diferentes dispositivos computacionales, y sus conocimientos previos sobre programación y CC. Para ambas dimensiones, contamos con una pregunta de múltiple opción. En el caso del acceso a dispositivos podemos observar en la Figura 2, que el estudiantado encuestado tiene mayormente acceso a por lo menos uno de los dispositivos consultados y a



internet.

Figura 2. Dispositivos computacionales a los cuales los estudiantes tienen acceso.

Con respecto a la experiencia previa, las opciones estaban vinculadas con las siguientes categorías: Uso de entornos de enseñanza de programación (en particular, Pilas Bloques y Scratch), programación de artefactos computacionales (incluye la programación, programación web, programación de videojuegos y el uso de entornos de enseñanza de programación), robótica educativa (incluye arduino y robótica), Inteligencia Artificial y Datos. Solo 3 estudiantes respondieron que no contaban con experiencia o conocimientos previos mientras que 32 estudiantes tenían experiencias vinculadas a la programación. Por lo que estamos hablando de una población con ciertos conocimientos en CC. En la Figura 3, podemos observar cómo se distribuyen los conocimientos previos de los estudiantes.

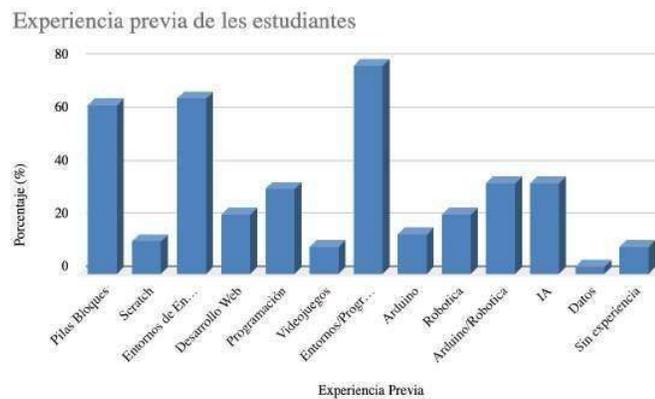


Figura 3. Experiencia previa de los estudiantes en diferentes contenidos de CC.

### Herramientas de recolección

Para conocer el impacto del taller en los estudiantes utilizamos dos encuestas como herramientas de recolección de datos. Una de las encuestas, fue diseñada para que los estudiantes completaran dos veces: como pre test (antes de comenzar el taller) y como post test (una vez finalizado el taller). Denominaremos a esta encuesta como *pre-pos test*. Con el *pre-pos test* intentamos capturar el impacto del taller de en cuanto a la opiniones de los estudiantes con respecto a la Inteligencia Artificial planteadas en las hipótesis en la Introducción. El pre-post test está compuesto por cinco afirmaciones, cada una asociada a una sola hipótesis:

*H5(nula) ...la superioridad de la IA frente a las personas para resolver problemas.”*

- A1. Las decisiones o recomendaciones que realiza la IA no dependen de las personas.
- A2. La IA puede usarse para tomar cualquier tipo de decisión.
- A3. La IA no comete errores.
- A4. La IA, como está implementada en una computadora, no tiene opinión y por lo tanto no puede discriminar ni prejuizar.
- A5. La IA resuelve los problemas mejor de que lo harían las personas.

En el pre y pos test pedimos a los estudiantes que indicaran su grado de acuerdo con cada una de las afirmaciones utilizando una escala Likert de 5 puntos – de 1 (nada de acuerdo) a 5 (muy de acuerdo) –. Para corresponder los pre y post test, asignamos un identificador para cada estudiante, para que las respuestas fueran anónimas y no revelaran nombre ni género.

Además, los estudiantes completaron otro cuestionario para reflejar su satisfacción con el taller, evaluando el aprendizaje autopercebido, claridad del equipo docente, interés generado en la temática. Para cada una de las categorías se diseñó una escala Likert (Nada, POCO, Mucho, Bastante), junto con preguntas sobre escuela, año de cursada, edad y género y preguntas abiertas sobre qué fue lo que más y menos les gustó y sugerencias.

### Resultados

En esta sección describimos los resultados obtenidos con las herramientas de recolección utilizadas. En la primera sección, analizaremos la validación estadística de las hipótesis en base a

las respuestas en los pre y pos test. En la segunda sección, compartiremos las apreciaciones de los estudiantes sobre el taller.

### Validación de hipótesis

Para validar las hipótesis descritas en la sección Introducción, analizamos el grado de acuerdo de los estudiantes sobre cada una de las afirmaciones propuestas en el pre y post test para evaluar cambios luego de la participación del taller. 33 estudiantes respondieron tanto el pre test como el pos test. Los 33 estudiantes que seleccionaron su grado de acuerdo con la afirmación *La IA puede usarse para tomar cualquier tipo de decisión*. El resto de las afirmaciones contó con 32 respuestas.

Para evaluar cuantitativamente el impacto del taller, aplicamos paired t test entre el pre y pos test, a los resultados obtenidos para cada una de las afirmaciones. En la Tabla 2, compartimos el valor del promedio para cada una de las afirmaciones en el pre y pos test. Además podemos observar los resultados de la media para el nivel de acuerdo para cada una de las afirmaciones, y la cantidad de estudiantes que respondió cada una de las afirmaciones en el pre y pos test.

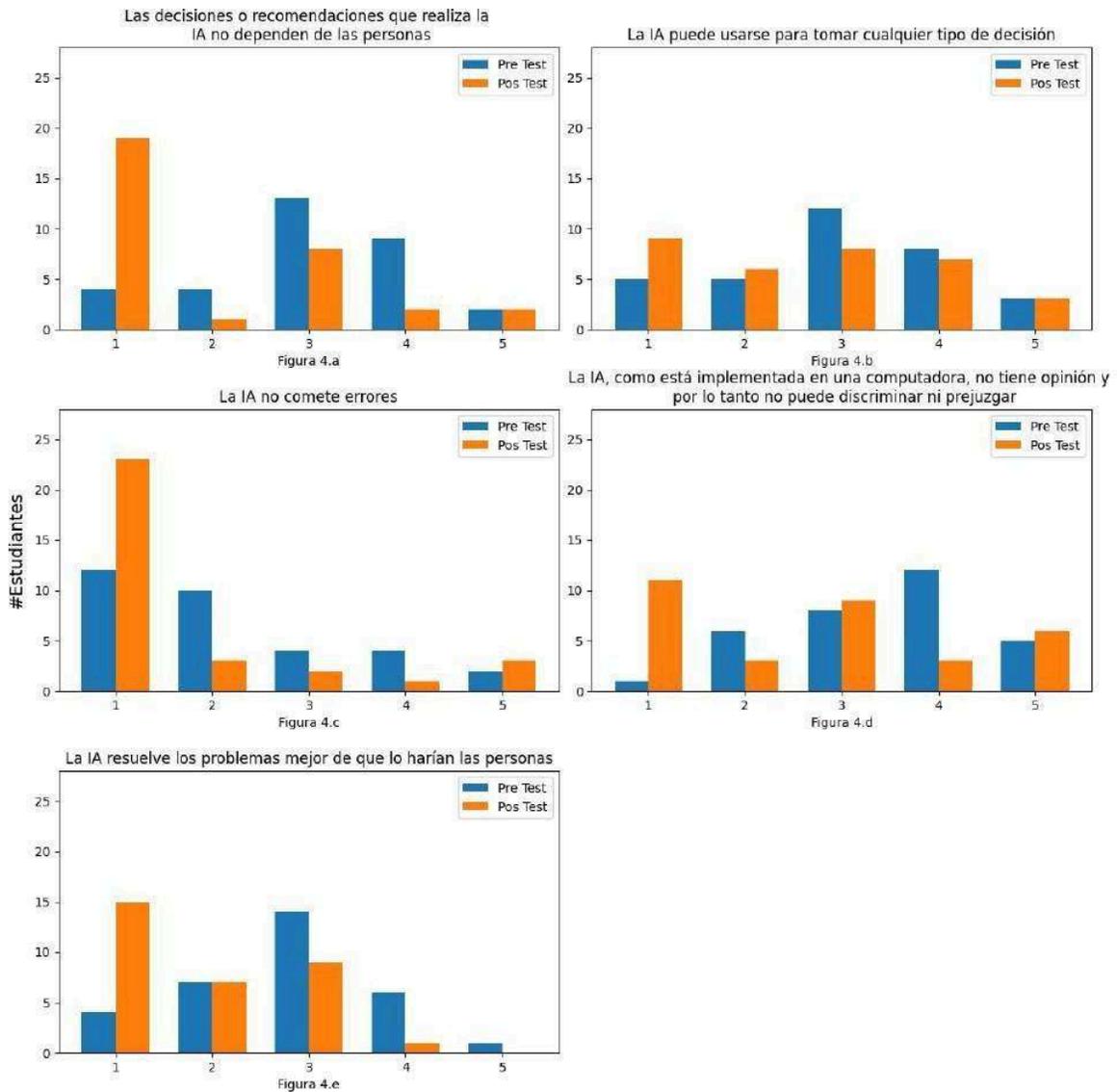
Afirmaciones	Media		#Estudiantes	paired t test	
	Pre test	Post test		p	t
A1	3,03	1,97**	32	0,0006	3,7922
A2	2,97	2,67	33	0,3096	1,0325
A3	2,19	1,69	32	0,0995	1,6982
A4	3,44	2,69*	32	0,0195	2,4628
A5	2,78	1,88**	32	0,0005	3,8587

Tabla 2. Promedio de respuestas de la escala likert en el pre y pos test, para cada una de las afirmaciones. Las diferencias estadísticamente extremadamente significativas se marcan con \*\* para las paired t test sobre las medias. Las diferencias significativas se marcan con \*.

En la Figura 4 podemos observar gráficos de frecuencia para los niveles de acuerdo del pre al pos test para cada una de las afirmaciones propuestas.

El taller generó un impacto muy significativo en la opinión de los estudiantes sobre la dependencia que tienen las decisiones o recomendaciones de la IA en las personas. En la Figura 4.a , podemos observar cómo la opinión de los estudiantes se acercaba a no tener una posición fija o al estar de acuerdo con respecto a la afirmación A1, a ser la opción Nada de Acuerdo la posición mayoritaria en el pos test. El paired t test muestra que existe una diferencia estadísticamente extremadamente significativa en esta métrica entre el pre y el pos test ( $p=0,0006$ ). Estos resultados permiten rechazar la hipótesis H1 (nula) en favor de la alternativa: el taller impacto de forma extremadamente significativa en la opinión de los estudiantes sobre el rol de las personas en las decisiones o recomendaciones que realizan las aplicaciones que utilizan IA.

Con respecto a las afirmaciones A2 y A3, podemos observar que la media reduce sus valores en el pos test, pero la diferencia del paired t test no es estadísticamente significativa en sus casos. En la Figura 4.b, podemos ver que en el caso de A2 las frecuencias en la escala likert, son similares en el pre y pos test. A pesar de no ser estadísticamente significativa la diferencia, vemos un crecimiento del valor Nada de Acuerdo, y un corrimiento en la media. Pero, en base a los valores observados podemos decir que la H2 (nula) se cumple. En el caso de A3, podemos observar en la Figura 4.c, que prácticamente se duplica en el pos test la cantidad de estudiantes que no están de acuerdo (23) con que la afirmación IA *no comete errores*. A pesar de eso, los valores del paired t test, no muestran una diferencia significativa entre ambos grupos, por lo que la H3 (nula) se cumple. En ambos casos, observamos un impacto en los valores de la media, y las frecuencias, a favor de rechazar la hipótesis, pero al mismo tiempo, observamos que las opiniones de los estudiantes ya estaban bastante consolidados en el pre test.



Nivel de acuerdo con la afirmación :1 (Nada de acuerdo) - 5 (muy de acuerdo)

Figura 4. Distribución de las respuestas en los pre y pos test, para cada una de las afirmaciones.

En la Figura 4.d, podemos observar cómo cambia la frecuencia de respuestas del pre al pos test con respecto a la afirmación 4. De contar con solo un estudiante que estaba Nada de acuerdo con la afirmación, pasó a ser considerada por 10 estudiantes en el pos test. El paired t test muestra que existe una diferencia estadísticamente significativa en esta métrica entre el pre y el pos test ( $p=0,0195$ ). Estos resultados permiten rechazar la hipótesis H4 (nula) en favor de la alternativa: el taller impactó de forma estadísticamente significativa en la opinión de los estudiantes sobre la opinión o prejuicio que pueden presentar las recomendaciones o decisiones tomadas por servicios basados en inteligencia artificial.

Con respecto a la afirmación A5, observamos que cerca de la mitad de los estudiantes no tenía una posición tomada en el pre test. Luego del taller, y en base a las respuestas del pos test

observamos que la frecuencia de la respuesta Nada de Acuerdo se cuadruplica. Además, podemos ver que solo un estudiante eligió como respuestas el valor 4 o 5. El paired t test muestra que existe una diferencia estadísticamente extremadamente significativa en esta métrica entre el pre y el pos test ( $p=0,0005$ ). Estos resultados permiten rechazar la hipótesis H4 (nula) en favor de la alternativa: el taller impactó de forma estadísticamente extremadamente significativa en la opinión de los estudiantes sobre pensar en que la Inteligencia Artificial resuelve mejor las tareas que las personas.

### **Análisis cualitativo del curso**

Para conocer las impresiones de los estudiantes a propósito de la experiencia del taller, observaremos la distribución de las respuestas del cuestionario que refieren a la percepción general del taller, al aprendizaje y a la propuesta didáctica. Como este formulario era opcional y anónimo se obtuvieron menos respuestas que con los pre y post test y no es posible cruzarlas con esas. De las 41 respuestas obtenidas, hay 5 que no incluyen ninguno de los aspectos valorativos del taller y, por lo tanto, para el análisis siguiente se considera el universo de las 36 respuestas completas, entendiendo que por su cantidad son representativas del total. Presentamos la información separada por género para indagar en posibles percepciones diferenciadas. Una sola persona expresó identificarse con el género “Otro” y se mostró fuertemente disconforme con el taller, al punto de dejar incompleta la mitad de este cuestionario, mientras que otra persona eligió no responder esta pregunta pero mostró mucha satisfacción. Consideramos que la representatividad de esta información es insuficiente y, por lo tanto, no consideraremos estas categorías al analizar proporciones por género.

En cuanto a la calificación general del taller, la distribución de las respuestas de los estudiantes reflejan un alto grado de satisfacción según lo exhibido por la Figura 5, con una alta concentración por encima de 7. Esto es válido tanto para quienes se identifican con el género masculino como con el femenino.

En cuanto al aprendizaje autopercebido y al interés por seguir aprendiendo, también se refleja un alto nivel de satisfacción tanto para el género masculino como el femenino. Las figuras 6.a y 6.b muestran las respuestas a las preguntas “¿Cuánto aprendiste sobre la temática del taller?” y “¿Te interesaría seguir aprendiendo sobre la temática del taller?”. Las categorías “Mucho” y “Bastante” concentran el 89% de las respuestas (por género: 95% de las respuestas que indicaron femenino y 86% masculino) y el 86% de las respuestas (por género: 85% de las respuestas que indicaron femenino y 93% masculino) para cada pregunta, respectivamente.

En cuanto al ejercicio docente y la perspectiva didáctica, se indagó sobre la percepción de la claridad de las explicaciones del equipo docente y qué tan teórico era el taller. La claridad del

equipo docente fue percibida positivamente (entre “Bastante” y “Mucho” se acumula 89% del total y, separado por género, se reúne el 85% de las respuestas que indicaron género femenino y el 93% que indicaron masculino), mientras que el enfoque didáctico del taller, si bien casi no fue percibido como “muy teórico” (2 respuestas, 6% del total), fue percibido como “bastante teórico” por una cantidad considerable de participantes (34% de las respuestas obtenidas para esta pregunta). Sin embargo, siguen siendo mayoría quienes percibieron el taller como “poco teórico” y “nada teórico” (60% del total, 60% de quienes indicaron género femenino y 64%, masculino), lo cual está alineado con lo expresado en la sección Propósitos del taller y perspectiva didáctica. La distribución de las respuestas a estas preguntas puede verse en las Figuras 7.a y 7.b.

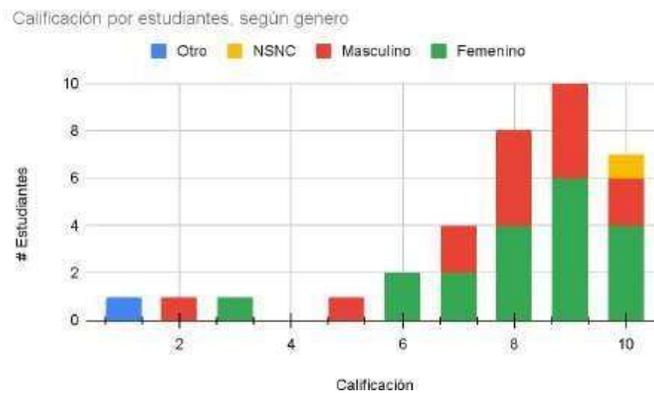


Figura 5. Distribución de la calificación del taller por parte de estudiantes separado por género.

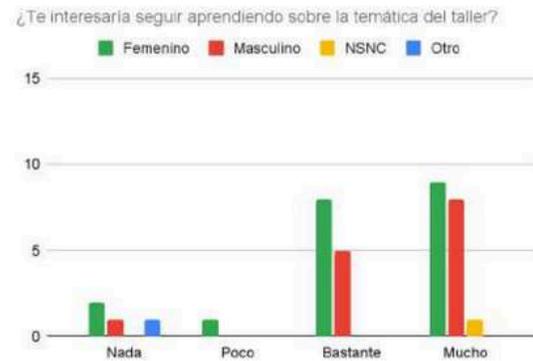
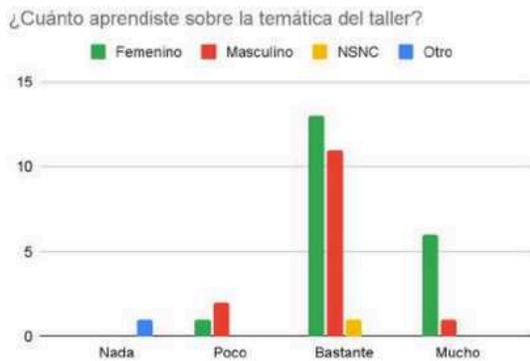


Figura 6.a y Figura 6.b (izq a der): Respuestas del estudiantado a las preguntas referidas a aprendizajes.

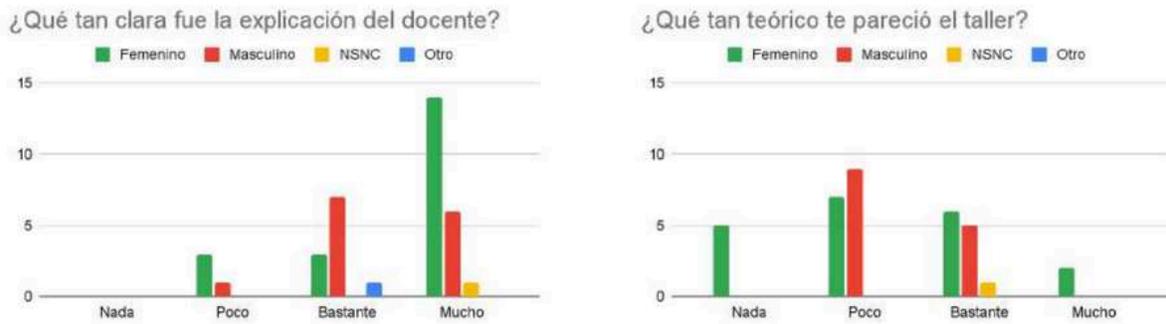


Figura 7.a y Figura 7.b (izq a der): Respuestas del estudiantado a las preguntas referidas a cuestiones didácticas.

### Limitaciones del estudio y trabajo futuro

Si bien los resultados observados son alentadores, refieren a una población pequeña y relativamente homogénea ya que eran estudiantes de cuatro cursos, algunos de ellos de escuelas técnicas. Además, como se usó un identificador no basado en el nombre para mantener el anonimato, debimos descartar datos por no poder asociar pre y post tests cuando este campo estaba mal completado. En ese sentido, podemos afirmar que en el contexto de las JADiCC 2022 la experiencia fue reconocida como valiosa y generó impacto entre los asistentes pero no contamos con evidencia para generalizar estos resultados a poblaciones más amplias. Podemos tomar esta experiencia como punto de partida para ampliar el alcance del taller y evaluar su impacto en otros contextos y escalas.

El dictado de este taller estuvo a cargo de especialistas de la Fundación Sadosky, con lo cual, un punto pendiente es la formación de docentes para implementarlo a mayor escala. En esta línea, parte de las formaciones docentes de la Iniciativa Program.AR -por ejemplo, el curso “La ciudadanía en un mundo atravesado por computadoras” (60hs) o el curso “IA, Internet y microchips: desmitificando en el aula la neutralidad de la tecnología” (40hs, en convenio con INFoD)- abordan estas actividades profundizando en el marco epistemológico y conceptual.

### Conclusiones

En base a los resultados vinculados a los pre y pos test, presentados en la sección Resultados, podemos observar que el taller tuvo un impacto en las opiniones previas que tenían los estudiantes sobre ciertos aspectos relativos a la neutralidad de la inteligencia artificial. Cada una de las hipótesis planteadas en la Introducción fueron vinculadas con una afirmación evaluada en el pre y post test. Como describimos en los resultados, hay una reducción en el valor de la media en cada una de las afirmaciones. En particular, para las hipótesis-afirmaciones H1-A1, H4-A4 y H5-A5 encontramos diferencias estadísticamente significativas al aplicar el paired t test, que permiten rechazar las hipótesis nulas a favor de las alternativas. Es decir, que impacta

principalmente en el nivel de acuerdo sobre las afirmaciones relativas a la no neutralidad de la inteligencia artificial, la influencia de las personas en las recomendaciones brindadas por las herramientas de IA y sobre la superioridad de las soluciones de IA por sobre las personas.

Las actividades del taller estaban pensadas para evidenciar problemas de no neutralidad de las aplicaciones de IA, el impacto de las personas al momento de desarrollar modelos (al momento de definir las clases y elegir los conjuntos de entrenamiento), y ciertos errores o equivocaciones en las decisiones de los modelos, en particular, los sesgos. El desarrollo de estas actividades, claramente impactó en el grado de acuerdo con las afirmaciones anteriormente mencionadas.

En el caso de las hipótesis-afirmaciones H2-A2, H3-A3, podemos observar que la opinión esperada por parte del impacto del taller ya está en parte consolidada en el pre test. Eso nos permite suponer que ya en primera instancia les estudiantes no están en general de acuerdo con la idea de que la IA no comete errores y de que puede ser utilizada para cualquier problema.

La distribución de las respuestas al cuestionario de valoración permite inferir que el taller, además de tener impacto sobre ideas específicas, fue recibido como una experiencia muy positiva e interesante. Además, si bien la mayoría de los estudiantes percibe haber aprendido bastante o mucho, percibieron el taller como una instancia poco o nada teórica. Esto está en línea con la propuesta didáctica planteada, fuertemente basada en una experiencia práctica y motivadora antes que en exposiciones y reflexiones puramente teóricas. Estas conclusiones aplican tanto para las personas que se identificaron como de género masculino como femenino, sin embargo, faltan datos para poder inferir la experiencia de quienes no se identifican con estas categorías.

## **Bibliografía**

- Bitzenbauer, P. (2023). ChatGPT in physics education: A pilot study on easy-to-implement activities. *Contemporary Educational Technology*, 15(3), ep430.
- Carney, M., Webster, B., Alvarado, I., Phillips, K., Howell, N., Griffith, J., ... & Chen, A. (2020). Teachable machine: Approachable Web-based tool for exploring machine learning classification. In *Extended abstracts of the 2020 CHI conference on human factors in computing systems* (pp. 1-8).
- Dabbah J., Garzón M., Gómez M., Martínez M.C. & Martínez López P. E. (2023) Propuesta curricular para la inclusión de las Ciencias de la Computación en la educación obligatoria de Argentina. Fundación Sadosky, Buenos Aires.
- Gómez, M. J. (2020). Aspectos de adquisición de lenguaje en la enseñanza de programación. Universidad Nacional de Córdoba-Facultad de Matemática, Astronomía, Física y Computación. Universidad Nacional de Córdoba. <https://rdu.unc.edu.ar/handle/11086/16051>.
- Lane, D. (2021). Machine learning for kids: A project-based introduction to artificial

intelligence. No Starch Press.

Lee, I., Ali, S., Zhang, H., DiPaola, D., & Breazeal, C. (2021). Developing middle school students' AI literacy. In Proceedings of the 52nd ACM technical symposium on computer science education (pp. 191-197).

Miao, F., & Shiohira, K. (2022). K-12 AI curricula. A mapping of government-endorsed AI curricula.

Su, Y., Lin, Y., & Lai, C. (2023). Collaborating with ChatGPT in argumentative writing classrooms. *Assessing Writing*, 57, 100752.

Tedre, M., Toivonen, T., Kahila, J., Vartiainen, H., Valtonen, T., Jormanainen, I., & Pears, A. (2021). Teaching machine learning in K–12 classroom: Pedagogical and technological trajectories for artificial intelligence education. *IEEE access*, 9, 110558-110572.

Zhai, X. (2023). Chatgpt for next generation science learning. *XRDS: Crossroads, The ACM Magazine for Students*, 29(3), 42-46.

## Reflexiones sobre la Enseñanza de Confiabilidad y Seguridad del Software en Bioingeniería con Programación Basada en Contratos

Jordán F. Insfrán, Javier E. Díaz Zamboni  
{jordan.insfran,javier.diaz}@uner.edu.ar

Laboratorio de Informática y Computación Aplicada  
Facultad de Ingeniería - Universidad Nacional de Entre Ríos

### Resumen

Este trabajo examina cómo la metodología de programación basada en contratos puede mejorar la enseñanza de las características de confiabilidad y seguridad del software en bioingeniería. A través de ejemplos, se explora cómo esta puede transformar la experiencia didáctica del diseño y desarrollo de software biomédico crítico. Desde una mirada docente, se analiza cómo la especificación de contratos podría influir en la construcción de software robusto. Los casos expuestos permiten reflexionar sobre los posibles desafíos y oportunidades al aplicar esta metodología en la enseñanza del diseño de software para dispositivos biomédicos.

Además, se muestra cómo la programación basada en contratos podría aportar a mejorar el desarrollo de software biomédico al expresar los requerimientos como parte del código fuente y posibles estrategias de incorporación a la enseñanza de programación en el área. En efecto, al establecer condiciones precisas para las entradas y salidas de funciones críticas, fortalece la comprensión de la relación entre el software y su entorno médico, resaltando el potencial de la metodología de programación basada en contratos para mejorar la confiabilidad y seguridad del software en bioingeniería.

Aunque los ejemplos analizados no buscan aplicaciones prácticas concretas y reales, actúan como lentes para explorar conceptos fundamentales. Estas reflexiones buscan iniciar un diálogo sobre cómo se podría incorporar la programación basada en contratos en la formación de futuros profesionales en bioingeniería, centrando la atención en cómo esta metodología puede transformar la construcción de software en el ámbito biomédico.

**Palabras claves:** Programación por contratos, sistemas críticos, bioingeniería.

### Introducción

En el campo de la bioingeniería, la creación de software para dispositivos biomédicos es fundamental. Estos dispositivos desempeñan un papel crucial en la atención médica y la vida de los pacientes, y es imperativo garantizar que el software que los controla sea confiable y seguro. La complejidad de este desafío se ve sustancialmente aumentada por la necesidad de cumplir con rigurosas normativas y estándares de seguridad, (IEC 61508-3, 2010; UNE-EN 62304,

2016). Para lograr confiabilidad y seguridad en estos desarrollos, es esencial que los profesionales de la bioingeniería cuenten con una sólida formación en construcción de software, incluyendo el uso de metodologías como la Programación Basada en Contratos (PbC). En este trabajo, exploramos la viabilidad de incorporar la PbC en la enseñanza de la programación en el ciclo básico de una carrera de Bioingeniería, considerando cómo puede contribuir significativamente a la confiabilidad del software en dispositivos críticos.

En dispositivos biomédicos críticos, como los de soporte vital o administración de medicamentos, la confiabilidad es la característica más importante. La confiabilidad se descompone en propiedades vinculadas, como disponibilidad, fiabilidad, seguridad y protección, donde la seguridad es una parte fundamental (Sommerville, 2005, p. 40). Otros autores afirman que la confiabilidad en sistemas de software es la combinación de correctitud y robustez o, en pocas palabras, como la ausencia de *bugs*, (Meyer, 1992).

Por tanto, surge la pregunta de si es factible incorporar la Programación Basada en Contratos desde el inicio de la formación en programación en el ciclo básico de una carrera de Bioingeniería, como una metodología que aporta al desarrollo de habilidades de construcción de software confiable. La PbC permite especificar formalmente las responsabilidades y comportamientos esperados del software a diferentes niveles, lo que proporciona una base sólida para la creación de sistemas más confiables y seguros.

En las próximas secciones, abordaremos dos aspectos fundamentales. En primer lugar, exploraremos cómo se modifica la implementación de una función y se establece su relación con los requerimientos mediante el uso de la PbC en el contexto de la programación estructurada. En segundo lugar, analizaremos la adaptación de la implementación de una clase al aplicar la PbC, especialmente cuando las instancias de la clase desempeñan un papel central en la cumplimiento de requisitos críticos en el paradigma orientado a objetos. Nuestro enfoque se centrará en la relación entre los requisitos que deben cumplirse y el código que define la función o la clase.

La organización del trabajo es la siguiente. En la sección 2 introducimos la PbC, en la sección 3 exploraremos la relación entre la construcción de software crítico en bioingeniería y su relación con la PbC, en la sección 4 analizamos la metodología de la PbC y su utilidad en la enseñanza. En la sección 5 reflexionamos a través de dos casos la incorporación de la PbC. Finalmente, en la sección 6 se analiza brevemente la incorporación del enfoque en una carrera de bioingeniería o similar. Nuestro enfoque se centrará en la estrecha relación entre los requisitos que deben cumplirse y el código que define la función o la clase.

## **Programación por contratos**

El diseño por contrato es un enfoque de desarrollo de programas que implica la inclusión de precondiciones y postcondiciones. Las precondiciones son afirmaciones verificadas antes de la invocación de una función, y todas deben evaluarse como verdaderas. Por otro lado, las postcondiciones son afirmaciones que se comprueban después de ejecutar el cuerpo de la función, y también deben ser todas verdaderas si las precondiciones lo son. Si una precondición no se cumple, no se puede asegurar el cumplimiento de las postcondiciones, ya que los supuestos necesarios para invocar correctamente la función en cuestión no se satisfacen. Dicho esto, cuando se emplean precondiciones y postcondiciones, se establece un contrato entre cada función y el resto del programa, y ambas partes deben cumplir con sus respectivos términos. Este contrato no solo proporciona una especificación precisa de la semántica de una función, sino que también contribuye a comprender su comportamiento y, lo que es aún más crucial, la forma correcta de invocación, (Bourque & Fairley, 2014; Meyer, 1992).

En orientación a objetos la PbC incorpora un concepto adicional: el invariante. El invariante es una afirmación que se asocia a una instancia de una clase. Esta debe evaluarse como verdadera después de la inicialización del objeto, y antes y después de la invocación a cualquier método público. Los invariantes permiten definir formalmente propiedades que deben mantenerse a lo largo del ciclo de vida del objeto.

### **Construcción de software crítico en bioingeniería y PbC**

En el ámbito del desarrollo de dispositivos biomédicos, si el dispositivo es crítico, significa que existen riesgos de fallos no tolerables con probabilidad de ocurrencia elevada, es decir, riesgos que amenazan la vida humana. Para estos riesgos o contingencias el sistema debe diseñarse de modo tal que la ocurrencia del fallo se minimice y, si se produce el fallo, evitar que provoque accidente o daño. Los posibles peligros vinculados al software generalmente se relacionan con deficiencias en la ejecución de servicios del sistema o con el colapso de sistemas de supervisión y seguridad. Los sistemas de supervisión tienen la capacidad de identificar situaciones eventuales de emergencia, como interrupciones en el suministro eléctrico, (Sommerville, 2005, Capítulo 3).

La PbC constituye una metodología que se alinea perfectamente con los objetivos de construcción de software crítico, porque proporciona un enfoque estructurado para especificar las expectativas y los comportamientos del software en términos formales y rigurosos. Esto fortalece la confiabilidad del software, contribuye a prevenir situaciones de riesgo al establecer límites claros para su funcionamiento, y fomenta un enfoque sistemático y cuidadoso en cada etapa del desarrollo.

La capacidad de la metodología de PbC para establecer expectativas precisas y su enfoque en la

correctitud (se hace lo que se pretende) y robustez (se proveen mecanismos para actuar frente a fallos) la convierten en una metodología que vincula de forma directa los requerimientos de software con la implementación. Esto se da tanto a nivel de componentes de subsistemas en diseños modulares (Meyer, 2003; Sommerville, 2005, p. 418), como a nivel del código fuente que implementa un módulo crítico, generalmente a base de un enfoque orientado a objetos, (De Carvalho et al., 2020; Meyer, 1992). Ello permite fortalecer desde la construcción, o programación, las características de seguridad e integridad en el desarrollo de software biomédico crítico mediante estrategias y principios guías en el diseño como los mencionados en la siguiente sección.

### **Metodología de programación basada en contratos en la enseñanza**

La metodología de PbC se enmarca dentro de la enseñanza de métodos formales y la correctitud por construcción (CpC) (Roderick Chapman, 2006). En el marco de CpC, destacan dos principios fundamentales: en primer lugar se debe dificultar la introducción de defectos y, en segundo lugar, se debe facilitar su detección y eliminación temprana del código fuente. Estos principios son esenciales para analizar cómo se enseña la programación. Y se puede afirmar que la PbC se alinea plenamente con ellos, porque promueve el análisis crítico de argumentos y de valores de retorno de funciones, como así también el de atributos y métodos en clases, que garantizan la preservación de estados válidos en los objetos. En sintonía con lo anterior, según SWEBOK-v3, se considera que la PbC mejora la calidad de la construcción de software (Bourque & Fairley, 2014), por lo tanto, consideramos relevante evaluar detenidamente su incorporación a la enseñanza.

Actualmente, la PbC forma parte de la estrategia de enseñanza de la programación desde los primeros años en otras universidades (De Carvalho et al., 2020; E. Kraemer, 2018; E. T. Kraemer et al., 2018). Como caso particular se puede mencionar a la *Innopolis University* (De Carvalho et al., 2020), quienes no solo han incorporado PbC en el primer curso de programación, sino que también comienzan con el paradigma de programación orientado a objetos. Su propósito es desarrollar habilidades de abstracción y de diseño. En nuestro país existen carreras del área de Ciencias de la Computación que tratan el enfoque de PbC como Ingeniería en Computación de la UNTREF (ME, 2008; UNTREF, 2022), Licenciatura en Análisis de Sistemas de la FIUBA (PE-FI-UBA, 2016) o Licenciatura en Ciencias de la Computación de la UNRC (EFQyN-UNRC, 2023), entre otras. Sin embargo, no se observa este tipo de enfoques en carreras de bioingeniería en lo que a metodologías de programación se refiere.

La enseñanza de la programación, aunque debe ser independiente del lenguaje, inevitablemente conduce a la elección de uno. Por lo tanto, es esencial contar con un criterio de selección de lenguaje y una metodología de enseñanza que capacite a los estudiantes para aprender nuevos

lenguajes basándose en los conocimientos adquiridos. En este sentido, es importante destacar que la PbC en sí misma es independiente del lenguaje, lo que significa que se puede aplicar en varios lenguajes de programación. Sin embargo, algunos lenguajes de programación facilitan más que otros la implementación de PbC. Existen varios lenguajes que admiten la PbC, incluyendo Python, Java, Eiffel y C++, entre otros. Algunos de estos lenguajes incorporan la PbC de manera nativa, mientras que otros la admiten a través de módulos o bibliotecas, o bien, como en el caso del lenguaje C, con estrategias basadas en el preprocesamiento del código fuente.

En este trabajo se ha elegido a Python 3 como lenguaje para presentar los casos de la aplicación de PbC. Para este fin se utiliza el módulo *icontract* en su versión 2.6.2 (disponible en el índice de paquetes de Python (PyPI)). Se destaca que Python tiene un amplio uso en la industria del software y en la academia. Además, es un lenguaje de alto nivel que da una comprensión y facilidad de aplicación más accesible a los conceptos de programación en general.

### Análisis de casos

En esta sección se presentan ejemplos y casos de ejemplo propios de la bioingeniería, como modelos simplificados de la realidad que permiten visibilizar la practicidad del enfoque de la PbC. En cada caso se realizan reflexiones sobre el código fuente destacando desafíos y oportunidades inherentes, con énfasis en su potencial impacto en aspectos relacionados a la confiabilidad y seguridad del software. Esto último constituye el elemento motivador que podría emplearse en el aula para reflexionar al respecto.

#### Caso 1: Función que controla la infusión de una sustancia en un paciente

Este caso considera la perfusión controlada de una sustancia, por ejemplo insulina, al cuerpo de un paciente. Por simplicidad, se omiten detalles de la sustancia infundida y solo se considera la dosis a infundir. El requerimiento se muestra en la Tabla 1.

Requerimientos 1
<ul style="list-style-type: none"> <li>La función debe entregar una dosis de sustancia. La dosis siempre es un valor positivo.</li> </ul>

Tabla 1: Requerimiento con precondition.

Código 1
<pre> from icontract import require, errors  INFUSION_COMPLETADA = None VOLUMEN_MEDICAMENTO_DISPONIBLE = 100  @require(lambda dosis: dosis &gt; 0) def infundir_medicamento(dosis): </pre>

<pre> <b>global</b> INFUSION_COMPLETADA, VOLUMEN_MEDICAMENTO_DISPONIBLE <b>if</b> dosis &lt;= VOLUMEN_MEDICAMENTO_DISPONIBLE:     # Aquí se simularía la infusión del medicamento.     INFUSION_COMPLETADA = <b>True</b>     VOLUMEN_MEDICAMENTO_DISPONIBLE -= dosis <b>else</b>:     INFUSION_COMPLETADA = <b>False</b>  dosis = 50 infundir_medicamento(dosis) <b>if</b> INFUSION_COMPLETADA:     print(f"La infusión de {dosis} ml se completó correctamente.") <b>else</b>:     print(f"La infusión no se pudo completar.")  dosis = -10 <b>try</b>:     infundir_medicamento(dosis) <b>except</b> errors.ViolationError <b>as</b> e:     print(f"Error: {e}") </pre>
<p><b>Salida 01</b></p>
<p>La infusión de 50 ml se completó correctamente.  Error: File ~01_ejemplo_funcion_infundir_medicamento_pre.py, line 14 in &lt;module&gt;:  dosis &gt; 0: dosis was -10</p>

Figura 1: Código y salida de función que realiza la infusión de un medicamento.

En la Figura 1 se muestra el código fuente que satisface el requerimiento junto a un código de prueba. En primer lugar se encuentra la importación de las dependencias *require* y *errors* desde el módulo *icontract*. El primero es un decorador que permite indicar una precondition y se la expresa como una función booleana y anónima. Por su parte, *errors* define las excepciones y errores de violación de contrato. La función que se analiza en este caso es *infundir\_medicamento*, que recibe un único parámetro (*dosis*) que establece la cantidad de sustancia a infundir. La precondition de la función establece que la dosis debe ser mayor que cero. El comentario en el cuerpo de la función tiene el propósito de representar una estrategia de infusión cualquiera (pulsada, en rampa, exponencial, u otra). No obstante, esto se omite para dar prioridad al empleo de PbC frente a la implementación concreta que simula la perfusión de la sustancia al paciente.

A continuación, se prueban dos invocaciones: por un lado con una dosis válida ( $dosis > 0$ ) y, por otro lado, con una dosis inválida ( $dosis \leq 0$ ). En la invocación válida, cuando se cumple con la precondition, la función modifica el valor de la variable de estado "INFUSION\_COMPLETADA" con valor verdadero o falso según corresponda. Luego, el programa retorna un mensaje indicando si se logró infundir la dosis al paciente. En la invocación inválida, cuando se incumple con la precondition, la función decorada con *require* lanza una excepción en tiempo de ejecución de tipo *ViolationError* la cual se captura para mostrar el mensaje de error que genera. Este

permite redireccionar el flujo del programa justo en el momento en que se viola la precondición, hacia otra parte donde podrían adoptarse medidas preventivas como, por ejemplo, la activación de una alarma.

Adicionalmente, el decorador *require* admite argumentos opcionales que merecen ser destacados. Por ejemplo, es posible añadir un mensaje personalizado para el caso en que la precondición no se cumpla, como se muestra en el siguiente ejemplo: `@require(lambda dosis: dosis > 0, "La dosis debe ser mayor que cero.")`. También, se ofrece la opción de desactivar la precondición mediante el argumento *enabled*, como en `@require(lambda dosis: dosis > 0, enabled=False)` lo que puede resultar útil en situaciones donde se quiera evitar el costo computacional asociado a su verificación, o cuando se trata de una precondición simbólica que solo puede evaluarse formalmente.

Ahora se supondrá que se vuelve necesario modificar los requerimientos de la Tabla 1 por los de la Tabla 2. Por un lado, ahora la dosis debe ser menor o igual al volumen de medicamento disponible y, por otro, se debe asegurar que la infusión sea completa. Aunque esto se realizaría en el código que simula la infusión, no existe ningún control en relación a la excepcionalidad de que este requerimiento no se cumpla. Si se simula una infusión incompleta en el código de la Figura 1 y la variable `INFUSION_COMPLETADA` es falsa, entonces, no habría lanzamiento de excepción que permita capturar un problema ajeno al código de la función en tiempo de ejecución.

<b>Requerimientos 2</b>
<ul style="list-style-type: none"> <li>● La función debe permitir dosis con valor positivo.</li> <li>● La dosis debe ser menor o igual al volumen de medicamento disponible.</li> <li>● Debe asegurarse que la infusión sea completa retornando verdadero en ese caso.</li> </ul>

Tabla 2: Requerimiento con precondición y postcondición.

El segundo requerimiento implica remover el control de la sentencia condicional de la Figura 1 simplificando el código y el tercer requerimiento constituye una postcondición. La forma de incorporar esto último consiste en añadir la línea `@ensure(lambda result: result == True)` debajo de las precondiciones como se muestra en la Figura 2, previamente importando la dependencia `ensure`. Notar que `ensure` utiliza el argumento `result` en la función anónima. Este es una referencia implícita al valor de retorno de la función y es el que se evalúa en la postcondición.

<b>Código 2</b>
-----------------

```

from icontract import require, ensure, errors

INFUSION_COMPLETADA = None
VOLUMEN_MEDICAMENTO_DISPONIBLE = 100

@require(lambda dosis: dosis > 0)
@require(lambda dosis: dosis <= VOLUMEN_MEDICAMENTO_DISPONIBLE) @ensure(lambda
result: result == True)
def infundir_medicamento(dosis):
    global VOLUMEN_MEDICAMENTO_DISPONIBLE
    # Aquí se simularía la infusión del medicamento.
    VOLUMEN_MEDICAMENTO_DISPONIBLE -= dosis
    se_pudo_completar_la_infusion = False
    return se_pudo_completar_la_infusion

dosis = 50
try:
    INFUSION_COMPLETADA = infundir_medicamento(dosis)
except errors.ViolationError as e:
    print(f"Error: {e}")

```

**Salida 2**

```

Error: File ~02_ejemplo_funcion_infundir_medicamento_pre_y_pos.py, line 8 in <module>:
result == True:
dosis was 50
result was False

```

Figura 2: Código y salida de función que realiza la infusión de un medicamento.

Un cambio adicional introducido en el código de la Figura 2 se relaciona con la incorporación de la asignación intencional a *se\_pudo\_completar\_la\_infusion*, una variable local. Esta asignación se ha diseñado deliberadamente para impedir el cumplimiento de la postcondición, incluso cuando las precondiciones se mantienen válidas. Como resultado, en la salida del Código 2, se produce una violación de contrato por parte de la función, ya que no logra satisfacer la postcondición especificada. Esta intencionalidad tiene un propósito pedagógico importante, ya que ilustra un concepto relevante: el efecto secundario (del inglés, *side effect*). El efecto secundario se produce cuando una función interactúa con elementos externos a ella y, en consecuencia, puede experimentar fallas inesperadas. En el caso estudiado, un ejemplo concreto podría ser la obstrucción de la cánula por donde fluye la solución infundida, lo que impide completar dicho proceso.

Este escenario podría utilizarse para demostrar cómo los parámetros opcionales de las postcondiciones permiten capturar situaciones específicas de manera precisa. Por ejemplo, definiendo 'class SideEffectViolationError(ViolationError): pass' se puede derivar una violación de contrato especializada para representarla. Así, los estudiantes podrían comprender cómo el control de postcondiciones se convierte en un mecanismo valioso para controlar fallas en el

desarrollo de software robusto.

Otro aspecto a observar en el Código 2, es que se ha removido una sentencia condicional del cuerpo de la función “*infundir\_medicamento*” con lo cual se simplificó el algoritmo que satisface la funcionalidad principal del programa. Aquí, la condición del condicional pasó a ser una precondition de la función.

Los códigos anteriores muestran que es posible emplear los mecanismos de manejo de excepciones para capturar las que se producen en tiempo de ejecución/interpretación a través de errores de violación de contrato -*ViolationError*-, y sus derivados -como el ejemplo de *SideEffectViolationError* que deriva de *ViolationError*- admitiendo diferentes alternativas de clasificación de excepciones las cuales pueden ser definidas y gestionadas por el equipo de diseño y desarrollador del software crítico.

### 1.1 Caso 2: Clase que modela una bomba de infusión

El siguiente caso consiste en una clase que modela una bomba de infusión para controlar el hardware encargado de administrar un medicamento a un paciente. Además, permite establecer algunos parámetros e informa resultados en una terminal de consola. Igual que antes, se omiten detalles técnicos por simplicidad. Los requerimientos están en la Tabla 3.

Requerimientos 3
<ul style="list-style-type: none"> <li>• El dispositivo debe admitir los estados OFF, ON e INFUNDIENDO. OFF cuando la interfaz de usuario está activa en modo <i>standby</i> (espera), ON cuando está encendida con el circuito de administración del medicamento energizado, INFUNDIENDO cuando el circuito de administración se encuentra infundiendo el medicamento. El estado inicial debe ser OFF.</li> <li>• La bomba de infusión debe establecer la capacidad del reservorio de medicamento al momento de crearse.</li> <li>• La tasa de infusión se debe poder configurar en cualquier momento con valores de 0,1 a 1500 ml/hora. Cuando se realiza esta configuración, se debe imprimir el mensaje “Tasa configurada: &lt;tasa&gt; ml/hora.” donde &lt;tasa&gt; es el valor de la tasa de infusión configurada.</li> <li>• La bomba de infusión debe poder medir el volumen de medicamento disponible.</li> <li>• Se debe permitir establecer la dosis en ml al momento de infundir el medicamento. Esta dosis no puede ser mayor a la capacidad del reservorio, ni al volumen disponible de medicamento.</li> <li>• La bomba de infusión debe permitir administrar el medicamento de forma controlada. Debe asegurarse que la infusión sea completa retornando verdadero en ese caso.</li> </ul>

Tabla 3: Requerimiento con invariantes, precondition y postcondición.

Se parte de un código fuente sin PbC para el aula propuesto por los docentes, que responde a los requerimientos de la Tabla 3, y la salida de sus pruebas en línea se muestran en la Figura 3. Este código se analizará brevemente y, luego, se expresará el mismo con PbC con el fin de mostrar comparativamente ambos enfoques.

### Código 3

```

Estado = Enum('Estado', ['OFF', 'ON', 'INFUNDIENDO'])
class BombaDeInfusion:
    def init_(self, capacidad_reservorio) -> None:
        self._capacidad_reservorio = capacidad_reservorio # en ml
        self._volumen_medicamento_disponible = 0          # en ml
        self._dosis_infundida = 0                         # en ml
        self._tasa_de_infusion = 0.1                     # en ml/hora
        self._estado = Estado.OFF                        # ON/OFF/INFUNDIENDO

    def encender(self):
        # Aquí iría el código que energiza el hardware de infusión
        self._estado = Estado.ON

    def apagar(self):
        # Aquí iría el código que desenergiza el hardware de infusión
        self._estado = Estado.OFF

    def establecer_e_informar_tasa_de_infusion(self, tasa_de_infusion):
        self._tasa_de_infusion = tasa_de_infusion
        print("Tasa configurada: " + str(self._tasa_de_infusion) + " ml/hora.")

    def agregar_medicamento_al_reservorio(self, volumen_medicamento):
        if self._volumen_medicamento_disponible + volumen_medicamento > self.
capacidad_reservorio:
            raise Exception("El volumen de medicamento excede la capacidad del
reservorio")
        self._volumen_medicamento_disponible += volumen_medicamento

    def devolver_volumen_medicamento_disponible(self):
        return self._volumen_medicamento_disponible

    def infundir_medicamento(self, dosis):
        if dosis <= self._volumen_medicamento_disponible:
            if dosis > 0:
                if self._estado == Estado.ON:
                    try:
                        self._estado = Estado.INFUNDIENDO
                        # Aquí se simularía la infusión del medicamento.
                        self._volumen_medicamento_disponible -= dosis
                        self._dosis_infundida = dosis
                        self._estado = Estado.ON
                        return True
                    except:
                        # Se tratan fallos en la infusión y en la reducción
                        # del volumen de medicamento disponible.
                        raise Exception("Error al infundir medicamento")

```



*Exception("El estado de la bomba no es ON")* debería ser anterior a todos los demás lanzamientos para una mejor organización del código. Además, se observa una falta de uniformidad en el lanzamiento de excepciones ya que el bloque *try-except* interno no se emplea en los otros casos de lanzamientos de excepciones (las declaraciones *raise* están en los bloques *else* de cada condicional). Esta última observación podría devenir del hecho de que el tratamiento de situaciones excepcionales se está realizando en conjunto con la codificación de la funcionalidad principal y es algo que frecuentemente se observa en las prácticas de laboratorio. Si el foco del estudiante se encuentra en programar la funcionalidad principal y, simultáneamente, en el tratamiento de situaciones excepcionales se interrumpe el primer foco de atención. Por otra parte, el bloque *try-except* presenta otro problema: atrapa todas las excepciones impidiendo discriminar el origen del problema en el código que implementa la funcionalidad principal.

Notar el código del método “apagar” en este caso no se indica nada respecto al estado inicial que debe tener el equipo para ser apagado. Aquí, podríamos preguntarnos qué debería hacerse si se decide apagar el equipo mientras se encuentra en el estado INFUNDIENDO. ¿Se debe apagar o se debe indicar primero que debe detenerse la infusión? Esta decisión, y su impacto en los requerimientos, constituye un aspecto pedagógico interesante e importante para motivar discusiones en el aula refiriendo incluso aspectos relacionados a la ética profesional. En efecto, si no hay requerimientos respecto al estado inicial que debe tener el equipo previo a su apagado, ¿por qué el desarrollador/a debería hacer algo más que cambiar el estado a apagado en el método “apagar”?

A continuación se muestra la implementación de la misma clase, pero expresada con la notación de contratos de *icontract*.

#### Código 4

```

from icontract import require, ensure, invariant, errors
from enum import Enum

Estado = Enum('Estado', ['OFF', 'ON', 'INFUNDIENDO'])

estados = [Estado.OFF, Estado.ON, Estado.INFUNDIENDO]

@invariant(lambda self: self._BombaDeInfusion_estado in estados)
class BombaDeInfusion:
    def __init__(self, capacidad_reservorio) -> None:
        self._capacidad_reservorio = capacidad_reservorio # en ml
        self._volumen_medicamento_disponible = 0 # en ml
        self._dosis_infundida = 0 # en ml
        self._tasa_de_infusion = 0.1 # en ml/hora
        self._estado = Estado.OFF # ON/OFF/INFUNDIENDO

```

```

@require(lambda self: self._BombaDeInfusion_estado == Estado.OFF)
def encender(self):
    # Aquí el código que energiza el hardware de infusión
    self._estado = Estado.ON

@require(lambda self: self._BombaDeInfusion_estado == Estado.ON)
def apagar(self):
    # Aquí el código que desenergiza el hardware de infusión
    self._estado = Estado.OFF

@require(lambda self: self._BombaDeInfusion_estado == Estado.ON)
@require(lambda tasa_de_infusion: 0.1 <= tasa_de_infusion <= 1500)
def establecer_e_informar_tasa_de_infusion(self, tasa_de_infusion):
    self._tasa_de_infusion = tasa_de_infusion
    print("Tasa configurada: " + str(self._tasa_de_infusion) + "
ml/hora.")

@require(lambda self: self._BombaDeInfusion_estado == Estado.ON)
@require(lambda volumen_medimento: volumen_medimento > 0)
def agregar_medimento_al_reservorio(self, volumen_medimento):
    if self._volumen_medimento_disponible + volumen_medimento > self.
capacidad_reservorio:
        raise Exception("El volumen de medicamento excede la capacidad del
reservorio")
    self._volumen_medimento_disponible += volumen_medimento

@require(lambda self: self._BombaDeInfusion_estado == Estado.ON)
def devolver_volumen_medimento_disponible(self):
    return self._volumen_medimento_disponible

@require(lambda self: self._BombaDeInfusion_estado == Estado.ON, "El
estado de la bomba no es ON")
@require(lambda dosis: dosis > 0, "La dosis no es mayor que cero")
@require(lambda self, dosis: dosis <=
self._BombaDeInfusion_volumen_medimento_disponible, "La dosis es mayor al
volumen de medicamento disponible")
@ensure(lambda self, result: result == True, "Error al infundir
medicamento")
def infundir_medimento(self, dosis):
    self._estado = Estado.INFUNDIENDO
    self._volumen_medimento_disponible -= dosis
    self._dosis_infundida = dosis
    self._estado = Estado.ON
    return True

@require(lambda self: self._BombaDeInfusion_estado == Estado.ON)
def devolver_dosis_infundida(self):
    return self._dosis_infundida

bomba = BombaDeInfusion(100) # Se ingresa capacidad del reservorio en ml
bomba.encender() bomba.agregar_medimento_al_reservorio(50) # En ml
bomba.establecer_e_informar_tasa_de_infusion(10) # En 10 ml/hora
se_pudo_infundir = bomba.infundir_medimento(30) # Equivale a 3 horas de
tratamiento

```

<pre> <b>if</b> se_pudo_infundir:     print("Se pudo infundir el medicamento")     resultado = bomba.infundir_medicamento(30) # Se intenta infundir 30 ml de     medicamento pero solo quedan 20 ml.     bomba.apagar() </pre>
<p><b>Salida 4</b></p>
<pre> Tasa configurada: 10 ml/hora. Se pudo infundir el medicamento Traceback (most recent call last):   File ~04_ejemplo_clase_bomba_infusion_icontract.py", line 69, in &lt;module&gt;     resultado = bomba.infundir_medicamento(30) # Se intenta infundir 30 ml de medicamento pero solo     quedan 20 ml.     ~~~~~ File ~_checkers.py", line 892, in wrapper result = func(*args, **kwargs)     ~~~~~   File ~_checkers.py", line 631, in wrapper raise violation_error icontract.errors.ViolationError: File ~04_ejemplo_clase_bomba_infusion_icontract.py, line 47 in BombaDeInfusion: La dosis es mayor al volumen de medicamento disponible: dosis &lt;= self._BombaDeInfusion_volumen_medicamento_disponible: dosis was 30 self was &lt;_main_.BombaDeInfusion object at 0x7f6279cb3c50&gt; self._BombaDeInfusion__volumen_medicamento_disponible was 20 </pre>

Figura 4: Código y prueba en línea de clase que controla una bomba de infusión con PbC.

Notar que los problemas del código anterior sin PbC se han resuelto. Destacan la la verificación del estado del equipo antes de ejecutar cada método, el orden y uniformidad en el lanzamiento de excepciones en el método "*infundir\_medicamento*" -el estado ON se verifica primero, y se remueve el bloque *try-except*-, y una simplificación importante de este último porque el cuerpo de dicho método es solo la funcionalidad principal: infundir una dosis de un medicamento a un paciente a una tasa dada.

Queda una cuestión muy relevante para discutir en las clases en relación al último código: ¿qué debería hacerse en caso de que se lance una excepción durante el proceso de infusión? En el código 4 no existe un bloque *try-except* interno que trate situaciones específicas dando lugar actividades de práctica reflexivas para pensar estrategias de atención de tales situaciones con excepciones definidas por el programador/a.

También se debe resaltar la incorporación del nuevo decorador: *invariant*. En este caso, el decorador se coloca sobre la declaración de la clase y permite definir un predicado que debe cumplirse luego de la inicialización del objeto y antes y después de la ejecución de cada método público que, para la clase *BombaDeInfusion*, establece que la bomba de infusión debe encontrarse en un estado válido en todo momento. Por supuesto, existen otras posibilidades para definir el invariantes y esta es solo una posibilidad.

### **Posible secuencia didáctica para incorporar PbC en la currícula**

Para integrar eficazmente la Programación por Contratos (PbC) en la currícula de bioingeniería, proponemos una secuencia didáctica articulada en dos cursos del ciclo básico de Bioingeniería: Fundamentos de Programación (FP) y Programación Avanzada (PA). En FP, enfocado en programación estructurada, se introducirán pre y postcondiciones en funciones, aplicando la PbC en problemas de diseño. En PA, la aplicación se ampliará a clases, abordando desafíos más complejos en el desarrollo de software biomédico. Los equipos docentes de las materias básicas pueden proponer cursos optativos para aplicar la PbC en contextos específicos, fomentando el análisis de contratos en casos de estudio biomédicos.

Esta secuencia busca consolidar los fundamentos del diseño de software, relacionar requisitos con el desarrollo en bioingeniería y cultivar habilidades para diseñar sistemas seguros mediante la aplicación de la PbC en la construcción de software biomédico a lo largo de la formación académica.

### **Conclusiones**

Se consideró en forma general y con ejemplos prácticos las posibilidades que ofrece la PbC como metodología para enseñar y trabajar la incorporación de las características de confiabilidad y seguridad en software crítico en carreras de bioingeniería.

Se valoraron aspectos sobre la relación entre la especificación de requerimientos y el código fuente donde se expresan las funcionalidades requeridas mediante la notación de contratos. Aquí, se resaltó cómo la metodología ofrece espacios de reflexión sobre la detección en tiempo de programación de requerimientos faltantes y la necesidad, desde el punto de vista de la ética profesional, de dar aviso de estas situaciones que podrían poner en riesgo la vida de un paciente.

Destacamos la versatilidad de la metodología de PbC porque permite su adopción en diferentes niveles de la currícula universitaria en forma gradual. Se presentó una posible propuesta de incorporación que ofrece flexibilidad a través de cursos obligatorios y optativos, en contextos de bioingeniería. Lo anterior permitiría cultivar habilidades prácticas esenciales desde el inicio de la formación académica.

Creemos que es importante tener en cuenta que una implementación exitosa de la PbC requeriría de un equilibrio cuidadoso en el diseño de las actividades teórico-prácticas, empleando la metodología para potenciar el aprendizaje alrededor de los conceptos esenciales correspondientes a cada espacio curricular.

### **Referencias**

- Bourque, P., & Fairley, R. E. (Eds.). (2014). SWEBOK: Guide to the software engineering body of knowledge (Version 3.0). IEEE Computer Society.
- De Carvalho, et al. (2020). Teaching Programming and Design-by-Contract. En M. E. Auer & T. Tsiatsos (Eds.), *The Challenges of the Digital Transformation in Education* (Vol. 916, pp. 68-76). Springer International Publishing.
- EFQyN-UNRC. (2023, noviembre 14). Plan de Estudios de Licenciatura En Ciencias De La Computación de la UNRC.
- IEC 61508-3. (2010). IEC 61508-3:2010. Functional safety of electrical/electronic/programmable electronic safety-related systems—Part 3: Software requirements (International Standard 61508-3; 2.0).
- Kraemer, E. (2018). Teaching the Design-by-Contract Concept in a Software Engineering Course Using RESOLVE. *ACM SIGSOFT Software Engineering Notes*, 43(3), 18-18.
- Kraemer, E. T., Sitaraman, M., & Hollingsworth, J. (2018). An Activity-Based Undergraduate Software Engineering Course to Engage Students and Encourage Learning. *Proceedings of the 3rd European Conference of Software Engineering Education*, 18-25.
- ME. (2008). *Res. N° 447 del Ministerio de Educación de la Nación*. <https://untref.edu.ar/uploads/Programas/programas-carreras/Ing.%20en%20Computacion%20-%20Plan%202007%20-%20RM%20447-08.pdf>
- Meyer, B. (1992). Applying «design by contract». *Computer*, 25(10), 40-51.
- PE-FI-UBA. (2016). *Plan de Estudios (PE) de la Licenciatura en Análisis de Sistemas de la FI-UBA*.
- Roderick Chapman. (2006). Correctness by construction: A manifesto for high integrity software. En T. Cant (Ed.), *10th Australian Workshop on Safety Related Programmable Systems (SCS '05)* (Vol. 55). Australian Computer Society, Inc.
- Sommerville, I. (2005). *Ingeniería del software* (M.-R. Miguel, Ed.; 7.ª ed.). Pearson Educación.
- UNE-EN 62304. (2016). *UNE-EN 62304:2007/A1:2016 Software de dispositivos médicos*. AENOR. <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0056471>
- UNTREF. (2022). *Res. C.S. N° 013/22 de la Universidad Nacional de Tres de Febrero*.

## **Formar profesores en Ciencias de la Computación para la educación obligatoria. Un aporte y un desafío desde la universidad**

Virginia Brasesco, Fernando Bordignon, Cecilia Sagol, Javier Kullock  
{virginia.brasesco, fernando.bordignon, cecilia.sagol, avier.kullock}@unipe.edu.ar

Universidad Pedagógica Nacional

### **Resumen**

En los últimos años ha surgido y se ha desplegado en distintos países un movimiento de alfabetización en Ciencias de la Computación sobre los distintos niveles que componen la educación obligatoria. En la Argentina, en particular, se están desarrollando e implementando modificaciones en los planes de estudio de las provincias (con distintos grados de avances) tendientes a incorporar tales saberes al currículum. En el marco de este escenario, es imperioso redefinir las propuestas de formación docente en relación a los saberes digitales, en particular a la computación. En esta oportunidad y con vocación de colaborar en la formación docente, presentamos, desde la Universidad Pedagógica Nacional (UNIPE), una nueva carrera de Profesorado en Informática. En este sentido compartimos su plan de estudios y características particulares, que entendemos que colaboran en ponerlo a la vanguardia en relación a demandas vigentes de incorporación de saberes en Ciencias de la Computación en el currículum de la educación básica.

**Palabras claves:** Ciencias de la Computación, Educación Básica, Formación del Profesorado.

### **Introducción**

En los últimos años ha surgido y se ha desplegado en distintos países un movimiento de alfabetización en Ciencias de la Computación (CC), principalmente con foco en la resolución de problemas y su programación.

En el marco de la sociedad del conocimiento han surgido necesidades de nuevas alfabetizaciones en estudiantes y docentes, como por ejemplo en medios, en lenguajes audiovisuales, en finanzas, entre otras. En esta línea el aprendizaje de la computación se percibe como un conjunto de conocimientos y habilidades que permiten comprender e interactuar de una manera más profunda con las tecnologías digitales y la informacionalización<sup>1</sup> de casi cualquier aspecto de la vida. Cada día que pasa somos conscientes que las CC están ayudando a promover importantes y acelerados cambios en el conocimiento humano, la percepción de nuestro entorno y en las costumbres que configuran nuestro desarrollo cotidiano. En este sentido,

---

<sup>1</sup> Informacionalización es la tendencia asociada al trabajo de uso de las tecnologías digitales como principal medio de trabajo y a la generación de bienes informacionales como principal producto.

la computación es una tecnología diferente, particular en sus efectos, dado que como personas nos permite extender nuestras capacidades cognitivas. Como señala en particular, el investigador Zukerfeld: en la actualidad son tres los elementos, propios del paradigma informacional (Castells, 2001), que están traccionando nuestras formas de producir: informacionalización, plataformización<sup>2</sup> y automatización (Zukerfeld, 2020).

Esta tendencia da cuenta del reconocimiento de que la computación es parte de los saberes fundamentales, a la par de las "tres R" clásicas de la alfabetización (lectura, escritura y aritmética). En particular, la programación es una alfabetización sobre la que es posible construir otras actividades y conocimientos, de la misma manera como hemos hecho con la escritura y la escritura tradicional (Vee, 2017). Así, tener una mirada de la alfabetización sostenida en contextos históricos, sociales y conceptuales nos estaría ayudando a comprender la programación informática como un importante fenómeno de comunicación y no simplemente como una nueva habilidad tecnológica, ya que la programación es una infraestructura; porque funciona como la base de todas nuestras comunicaciones digitales (ibid).

Esta realidad ha sido reconocida y ha generado preocupación en las políticas públicas de diversos países, que, desde hace más de una década, han impulsado cambios en la currícula de la educación básica obligatoria, donde principalmente han integrado la disciplina de CC como un nuevo saber a construir para los estudiantes. Además, estas acciones, para las políticas educativas de algunos países, se perciben cómo estratégicas para mejorar sus instituciones democráticas y también construir una mayor soberanía sobre temas relacionados con las tecnologías digitales.

De esta manera, en los últimos años se ha avanzado en el desarrollo e implementación de políticas educativas en relación a la enseñanza de los conceptos básicos de las Ciencias de la Computación con foco en la escuela obligatoria. Así, gobiernos de distintos países, ya sea desarrollados o en vías de desarrollo, han reformulado sus políticas educativas y han incorporado saberes en torno a las CC al currículum escolar. Por ejemplo, países tales como India, Israel, Brasil, Estonia, Costa Rica, Australia, Vietnam, Reino Unido y Nueva Zelanda ya han adoptado políticas educativas en tal sentido (European Education and Culture Executive Agency, 2022; Borchard y Roggi, 2017; K-12 Computer Science Framework Steering Committee, 2016; Fundación Sadosky, 2013; Bell, 2014; Furber, 2012)

En particular, en la República Argentina se viene realizando una serie de discusiones y cambios en las políticas públicas educativas tendientes a proponer y promover la implementación de diversas estrategias para que saberes propios de las CC se introduzcan en los planes de estudio

---

<sup>2</sup> Plataformización es la tendencia del trabajo a estar atravesado por plataformas en línea de Internet.

de los niveles inicial, primario y secundario. En el año 2015, en el marco del Consejo Federal de Educación se declaró de importancia estratégica para el país la enseñanza de la programación durante la escolaridad obligatoria (Consejo Federal de Educación, 2015). Al año siguiente, el proyecto Nueva Escuela Secundaria, de la Ciudad Autónoma de Buenos Aires, realizó cambios en los contenidos de la materia Tecnologías de la Información y añadió contenidos específicos de pensamiento computacional y programación de computadoras. Luego, en el año 2018, el Consejo Federal de Educación aprobó un documento denominado Núcleos de Aprendizajes Prioritarios (NAPs) para Educación Digital, Programación y Robótica, que contiene una serie de temas y objetivos a incorporarse a los niveles Inicial, Primario y Secundario del país en relación a las Ciencias de la Computación (Consejo Federal de Educación, 2018). Los nuevos NAPs representan un potencial avance desde las políticas públicas, ya que abordan cuestiones referentes al desarrollo de nuevos saberes digitales, pero, a la vez, suponen un conjunto importante de desafíos educativos para su implementación real. Siguiendo esas definiciones, algunas jurisdicciones fueron agregando apartados a sus diseños curriculares, en los distintos niveles de la educación obligatoria, siguiendo los avances descritos en los NAP nacionales.

Para llevar adelante efectivamente, tales decisiones de política pública, se necesitan otros elementos del esquema educativo, por ejemplo: diseños curriculares, materiales de estudio, docentes capacitados y lineamientos y materiales didácticos para la enseñanza y el aprendizaje. Estos elementos deben ser creados, investigados e implementados para construir conocimiento pedagógico en esta área. Así, en consonancia con las políticas públicas educativas mencionadas y con la finalidad de formar profesores de informática, que desde sus conocimientos y prácticas dialoguen con este nuevo currículum emergente en Ciencias de la Computación, se presenta un nuevo diseño curricular para la formación de profesores.

En la siguiente sección presentaremos un resumen de la carrera "Profesorado en Informática" de la Universidad Pedagógica Nacional. En las secciones siguientes se encuentra un breve racconto de las materias divididas por áreas temáticas: La programación y su didáctica, Infraestructura Tecnológica y su didáctica, Tecnología, Sociedad y Ciudadanía Digital y Ciencia de Datos. Finalmente el artículo cierra con algunas reflexiones.

### **Presentación de la carrera**

En el año 2020 la Universidad Pedagógica Nacional junto a la Fundación Sadosky diseñaron un plan de estudios correspondiente a un nuevo modelo de Profesorado en Informática, que fue puesto en funcionamiento en la UNiPE en el año 2021 en sus sedes Metropolitana y Pilar.

El objetivo principal de la carrera es formar profesores en Informática con una base académica de calidad en el ámbito de la Universidad Pública para cubrir las demandas de los sistemas

educativos jurisdiccionales. A su vez, también, se busca construir un espacio de reflexión investigación y producción de conocimiento sobre el alcance de la tecnología y su impacto en la sociedad; presentar los fenómenos computacionales que forman parte de la vida cotidiana, problematizándolos desde sus aspectos técnicos y teóricos más relevantes y cómo éstos se relacionan; reconocer y valorar la importancia del conocimiento de las Ciencias de la Computación para la formación ciudadana y, finalmente, que los egresados puedan diseñar propuestas de clase superadoras, interdisciplinarias que abordan situaciones problema más complejos y vinculados con su entorno.

De acuerdo a lo establecido en los Lineamientos generales para la formación docente aprobados por el Consejo Interuniversitario Nacional (Res. CE N° 787/12) y por el Consejo Universitario, la estructura curricular del Profesorado está organizada en cuatro campos: el Campo de la Formación General, el Campo de la Formación Pedagógica, el Campo de la Formación Disciplinar Específica y el Campo de la Formación en la Práctica Profesional Docente.

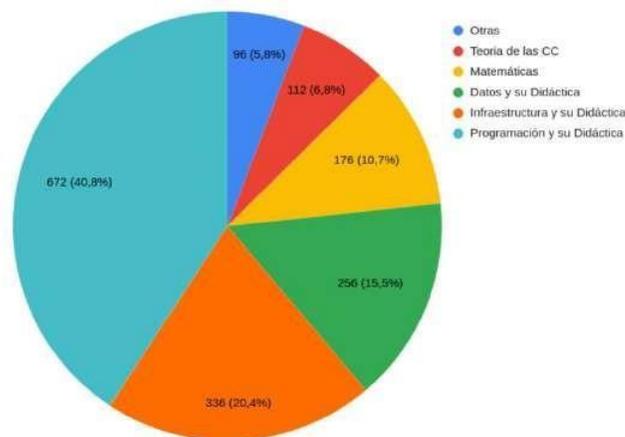


Figura 1. Distribución de horas del Campo de Formación Disciplinar Específico. Fuente: elaboración propia

El Profesorado en Informática tiene una duración de 4 (cuatro) años y una carga horaria total de 2.856 horas. De las cuales más del 57% (1.648 horas) son parte del total del Campo de la Formación Disciplinar Específica; y dentro de estas horas más del 40% son materias de Programación y su didáctica, ver figura 1.

La siguiente figura 2 muestra la organización de las materias del Profesorado en Informática agrupadas por año:

Materias		Materias		
Primer año	Mundo moderno I	48 hs.	Seminario tipo I	48 hs.
	Programación I	128 hs.	Programación III	128 hs.
	Matemática discreta	96 hs.	Didáctica de la programación	96 hs.
	Pedagogía	48 hs.	Psicología Educacional	48 hs.
	Mundo moderno II	48 hs.	Práctica profesional II	48 hs.
	Programación II	128 hs.	Seminario tipo II	48 hs.
	Organización y arquitectura de computadoras	64 hs.	Programación IV	64 hs.
	Didáctica general	48 hs.	Sistemas operativos	64 hs.
	Práctica profesional I	48 hs.	Teoría de la computación	48 hs.
			Sociología de la educación	48 hs.
		Práctica profesional III	48 hs.	
Tercer año	Seminario tipo III	48 hs.	Política educacional	48 hs.
	Redes informáticas y telecomunicaciones	64 hs.	Ciencia de datos e inteligencia artificial	80 hs.
	Didáctica de la programación II	96 hs.	Tecnología, sociedad y ciudadanía digital	64 hs.
	Bases de datos	48 hs.	Materia electiva I	48 hs.
	Historia de la educación	48 hs.	Inglés nivel III	48 hs.
	Inglés nivel 1	48 hs.	Residencia pedagógica I	96 hs.
	Seminario optativo tipo 1	48 hs.	Problemáticas de la educación secundaria	48 hs.
	Objetivos digitales interactivos	48 hs.	Didáctica de la programación IV: <small>Ciencia de datos y tecnología, sociedad y ciudadanía digital</small>	64 hs.
	Didáctica de la programación III: Infraestructura tecnológica	96 hs.	Introducción a la ingeniería del software	48 hs.
	Probabilidad y estadística	80 hs.	Materia electiva II	48 hs.
	Práctica profesional IV	48 hs.	Seminario optativo tipo II	48 hs.
	Inglés nivel II	48 hs.	Residencia pedagógica II	96 hs.
			Actividad científica, cultural y/o académica	8 hs.

Figura 2 - Plan de estudios del Profesorado en Informática. Fuente: elaboración propia.

La formación disciplinar específica se divide en siete áreas: 1) Programación, 2) Infraestructura tecnológica (hardware, sistemas operativos y redes informáticas), 3) Datos (bases de datos, ciencia de datos e inteligencia artificial), 4) Fundamentos matemáticos de las CC, 5) Didáctica de las Ciencias Computacionales, 6) Tecnología y Sociedad y Ciudadanía Digital y 7) Teoría de los sistemas de cómputo.

La propuesta de estudios en informática le da importancia a la programación, y la pone de relevancia dentro de las materias específicas de su formación. Esto se enmarca dentro de la teoría de utilizar la programación como herramienta para enseñar conceptos computacionales

más complejos, como por ejemplo la ciencia de datos, la inteligencia artificial o el diseño de sistemas de software complejos. Así, se espera que las y los egresados de la formación, tengan una sólida noción de un abanico de lenguajes computacionales y dominio de distintos entornos de programación y nociones de algoritmia.

Dada la especificidad con que cuenta la enseñanza de la disciplina y el desafío de construirla en esta instancia, el área Didáctica de las Ciencias Computacionales aborda la didáctica escolar de los diversos contenidos que se abordan en la carrera.

Por otra parte, el área Datos apunta a comprender de qué manera los sistemas informáticos actuales manejan grandes volúmenes de información, cómo realizan búsquedas veloces entre esos enormes volúmenes y cómo se procesan esos datos para que puedan ser viables las múltiples aplicaciones de la inteligencia artificial.

Asimismo, en el área Fundamentos Matemáticos de las Ciencias Computacionales se trabajan algunos saberes propios de la matemática que son base fundamental de la disciplina informática así como de su comprensión y análisis.

El área de Tecnología, Sociedad y Ciudadanía Digital complementa una visión que es transversal a toda la carrera, en la que se problematiza la tecnología y se la analiza desde la perspectiva de las distintas relaciones sociales que la atraviesan y a las que ésta contribuye a formar.

Finalmente, el área Teoría de los sistemas de cómputo aborda, desde un enfoque teórico, aspectos fundamentales de la disciplina que hacen a un entendimiento más profundo de los modelos de cómputo y las técnicas de desarrollo, a pesar de que exceden lo que se abordará como contenidos en la enseñanza escolar. El objetivo del área, a diferencia de las anteriores que transitan temas que se trabajarán explícitamente en el aula, es consolidar la formación profesional del docente y fortalecer su entendimiento de la disciplina.

Las materias electivas contribuyen a la formación general del profesor/a. En general, se orientan hacia el desarrollo de un carácter más disciplinar y en ese caso sirven para brindar una formación en temas novedosos o que reflejen el estado del arte de la disciplina. También pueden tener una orientación más didáctica: en este caso sirven para introducir enfoques novedosos, experimentales u otros que se consideran valiosos y no tengan lugar definitivo en el resto del plan de estudios.

En relación a la implementación y desarrollo de la nueva carrera, podemos observar que si bien el primer año, que se desarrolló en tiempo de aislamiento por la pandemia COVID19, fue dictado en modalidad virtual, la carrera está diseñada para desarrollarse en modalidad presencial. Así, a partir del año 2022 todas las materias empezaron a dictarse presencialmente con algunas pocas

excepciones por motivos de distancia de los y las docentes que residen en otras provincias, en las que se recurre a la modalidad sincrónica virtual en modo seminario.

Asimismo, para expandir el aula de clases, la universidad cuenta con un campus virtual donde cada materia organiza el material de sus cursadas (bibliografía, calendario, entregas de trabajos, etc) en aulas virtuales, sumando acciones de coordinación, comunicación y de trabajo colaborativo que ocurren durante cada semana. Esto va formando además, un corpus de materiales didácticos que son importantes en esta propuesta fundacional, innovadora y pionera.

En relación a la infraestructura y equipamiento de apoyo a la enseñanza y el aprendizaje, la universidad cuenta con un importante surtido de computadoras portátiles en ambas sedes, las cuales son utilizadas en las materias, y pueden ser solicitadas por el estudiantado para seguir practicando fuera de la clase.

A continuación se desarrollarán aspectos principales en relación al enfoque, objetivos y contenidos de una selección de áreas y materias clave de la formación disciplinar.

### **La programación y su didáctica**

Las cuatro materias de programación con las que cuenta el profesorado y las dos materias de didáctica de la programación se enfocan en formar docentes con conocimientos profundos de la programación y la algoritmia.

Es interesante remarcar que no se requiere que los/las estudiantes que ingresan tengan conocimientos previos en el área de programación (así como en las otras mencionadas). Durante las cursadas se trabaja a partir de los conceptos básicos, considerando la construcción desde el inicio del instrumental de programación apropiado. Es importante, ya que se cumple con la doble asignación de trabajar desde esos conceptos poniendo en juego en la práctica la didáctica específica que luego se profundizará en las materias del área correspondiente.

En la materia Programación 1, el enfoque está en los conceptos de programación imperativa y también las prácticas de depuración del código. Esta materia utiliza lenguajes de programación por bloques y sin computadoras. Se utiliza la plataforma Pilas Bloques en el primer tercio de la materia y luego se continúa con Zamba (Brassesco et al., 2021), una plataforma creada por docentes de la UNiPE para poder trabajar con problemas de entrada y salida de texto, y la traducción del código en bloques a lenguaje textual. Hacia el fin del cuatrimestre, las y los estudiantes son capaces de modularizar la estrategia de solución a los problemas planteados utilizando funciones y procedimientos con y sin parámetros; detectar errores en código propio o ajeno utilizando técnicas de depuración; analizar y distinguir dos o más soluciones como implementaciones un mismo problema.

La materia Programación 2 está pensada como una continuidad con la materia Programación 1. En este sentido, el objetivo mínimo de Programación 1 es que sean capaces de dar una estrategia de solución modularizada a un problema, pudiendo o no codificar todas sus partes. En Programación 2 se busca que las y los estudiantes programen en un lenguaje textual y utilicen técnicas de verificación de código basada en especificación semi formal. El lenguaje elegido en estas primeras cohortes fue Python debido a la disponibilidad de material de consulta libre y en español que circula en internet. En esta materia se hace foco en la utilización de tipos de datos abstractos y su relación con la noción de clase y objeto. Las estructuras de datos utilizadas en esta materia son las secuencias de tipo listas, pilas y colas.

En Programación 3 se introducen nociones de algoritmia sobre estructuras de datos, como por ejemplo: búsqueda binaria en arreglos, operaciones sobre árboles de búsqueda (búsqueda, inserción, borrado y modificación) y algoritmos de ordenamiento. En esta materia mediante el uso de punteros y referencias las y los estudiantes codifican sus propias estructuras de datos en forma de nuevas clases, dependiendo el problema y la complejidad temporal deseada para resolverlo.

La materia Programación 4, centrada en la programación orientada a objetos, profundiza en las propiedades de código como modularización, cohesión, ocultamiento de información y su modificabilidad. Utilizando teoría de grafos, se resuelven problemas tradicionales, por ejemplo, recorridos (en anchura y en profundidad), caminos mínimos, circuito euleriano vs. hamiltoniano. Además a lo largo de la materia se introduce la noción de clases de problemas P, NP y NP Completo y la diferencia entre algoritmos exactos y heurísticas.

Como se puede ver el nivel de profundidad de los temas de programación y algoritmia es elevado, y similar al grado de estudio en carreras como ingenierías y licenciaturas en informática que excede por supuesto los contenidos que estos docentes tendrán que dictar cuando entren en funciones en el sistema educativo básico. Si bien en la escolaridad obligatoria, los temas de programación 3 y 4 no están incluidos, lo que se busca con este contenido es reforzar los conocimientos y nuestros egresados y egresadas puedan reconocer la resolución de situaciones problemáticas tradicionales dentro del campo disciplinar, así como la capacidad de abordarlas a partir de estrategias de trabajo actualizadas tanto dentro como fuera del aula, siempre basado en la participación, la cooperación y la construcción colectiva del conocimiento.

Buscamos que las y los profesores puedan no sólo utilizar las herramientas y plataformas existentes, sino que se vuelvan creadores de nuevas herramientas didácticas. En este sentido en las materias de Didáctica 1 y 2, se analiza con una mirada socio histórica la construcción de la disciplina, mientras que se diseñan distintas actividades y secuencias didácticas para la enseñanza de la programación en la escuela. Brassesco (2023) analiza el diseño de la materia

de Didáctica 1 y su implementación en el ámbito de la UNIPE.

### **Infraestructura Tecnológica y su didáctica**

En el área de infraestructura tecnológica encontramos 5 materias, 4 disciplinares y una de didáctica específica. Estas materias son: Organización y Arquitectura de Computadoras, Objetos Digitales Interactivos, Sistemas Operativos, Redes Informáticas y Telecomunicaciones y Didáctica de la Programación III.

En el dinámico y en constante evolución campo de la Informática, la formación de futuros profesores no sólo se trata de adquirir conocimientos técnicos, sino también de desarrollar una comprensión profunda de los fundamentos que sustentan las tecnologías modernas y su impacto en la sociedad. En este contexto, las materias que conforman el área de infraestructura tecnológica desempeñan un papel fundamental al proporcionar a los estudiantes una base sólida para comprender y abordar los desafíos presentes y futuros en el ámbito de la informática.

Organización y Arquitectura de Computadoras se erige como un pilar esencial en la formación de profesionales de la informática, ya que brinda a los estudiantes una visión detallada de cómo los componentes de hardware interactúan para ejecutar aplicaciones y procesar información. Esta materia no sólo explora las complejidades de los circuitos y la estructura de las computadoras, sino que también trasciende a cuestiones más amplias, como la eficiencia energética y la escalabilidad de los sistemas, que son vitales en un mundo cada vez más consciente de la sostenibilidad. Buscamos que en la materia se den debates respecto a elecciones de componentes en líneas de producción de dispositivos a nivel nacional, tomando por ejemplo el caso de la fabricación de computadoras para el plan Conectar Igualdad desarrollado por el Ministerio de Educación de Argentina.

En el núcleo de cualquier sistema informático se encuentran los Sistemas Operativos, piezas de software que gestionan los recursos y brindan una interfaz para que los usuarios interactúen con las máquinas. La comprensión profunda de cómo funcionan estos sistemas es esencial para garantizar un rendimiento óptimo, la seguridad y la confiabilidad de las aplicaciones y servicios informáticos. En sintonía con la discusión en torno a la creación de netbooks destinadas a fines educativos, el diálogo persiste en torno a la concepción de un sistema operativo de código abierto. Este sistema operativo se plantea con el objetivo de proporcionar una variedad de recursos de aprendizaje adecuados para las múltiples realidades de las aulas en toda la República Argentina, se analiza el caso Huayra GNU/Linux (<https://huayra.educar.gob.ar/>) y en la materia las y los estudiantes construyen sus propios sistemas pensando en sus futuros estudiantes.

Los Objetos Digitales Interactivos (ODI) abren un fascinante mundo de posibilidades al permitir a

los estudiantes explorar aspectos de la computación física. De manera tradicional, el diseño, el desarrollo y la generación de prototipos de dispositivos electrónicos digitales estuvo exclusivamente en manos de profesionales de disciplinas afines a la ingeniería electrónica. Sin embargo, en los últimos años, se observa una emergencia de una capa de abstracción para el diseño, una baja de costos de componentes y una buena disponibilidad de las herramientas y los elementos necesarios para el desarrollo de proyectos de electrónica digital programables. En este contexto, entendemos que un objeto digital interactivo es un artefacto tangible digital que, por un lado, posee un componente físico determinado por su estructura mecánica y elementos eléctricos y electrónicos asociados y, por otro lado, tiene asociado un software que permite controlar y dar “vida” a las distintas operaciones para las que fue creado. Así, es interactivo dado que a partir de una serie de sensores y actuadores puede interactuar con su entorno en base a instrucciones de programa que le indican cómo hacerlo.

Desde la educación, entendemos a los ODIs como “nuevos objetos para pensar”, asociados a la formación de los estudiantes. Esto se sustenta en que las computadoras se han constituido como aliadas de importancia para llevar adelante las propuestas de tipo constructoras. Son herramientas adecuadas para el desarrollo y generación de prototipos de objetos para pensar, debido a su característica proteica (como observará el profesor Seymour Papert), es decir a su capacidad para cambiar continuamente su naturaleza, apariencia y comportamiento. Nuestra práctica, que incluye computación física, se centra en el uso de placas Arduino (de software y hardware abierto) dado que resultan sencillas de utilizar. Esto se debe al alto nivel de abstracción de sus interfaces y a una gran comunidad en línea que da soporte y apoyo de forma desinteresada a quien lo requiera. En términos de aprendizaje, estos artefactos digitales proveen experiencias educativas caracterizadas por un piso bajo (fácil acceso) y un techo alto (muchas posibilidades de desarrollo). El curso busca un acercamiento al trabajo interdisciplinario, donde además de promover el desarrollo de habilidades técnicas, al diseñar y construir ODIs que respondan a situaciones problemáticas, los estudiantes disponen de un ambiente de trabajo que habilita un diálogo potencial con diferentes disciplinas, a partir de las situaciones que emergen durante el diseñar y construir prototipos de ODIs.

La materia de Redes Informáticas y Telecomunicaciones capacita a los estudiantes para diseñar infraestructuras de comunicación, conectando dispositivos y personas a través de redes locales y globales, impulsando la interconectividad en el mundo actual. Durante la pandemia de COVID-19, las redes informáticas ganaron relevancia. ENACOM Argentina garantizó acceso gratuito a páginas educativas en colaboración con empresas celulares para asegurar continuidad educativa. El plan Conectar Escuelas busca brindar acceso a internet en aulas para todos los estudiantes. En Uruguay, el plan Ceibal ofrece conectividad y una intranet para respaldar la educación nacional. En esta materia entonces no sólo se analiza el aspecto técnico de las redes,

sino también su relevancia en la construcción de una sociedad más democrática.

Finalmente, la Didáctica de la Programación III desempeña un papel crucial al abordar la pedagogía de la enseñanza de la programación. No se trata sólo de transmitir conocimientos, sino de fomentar la habilidad de enseñar de manera efectiva, adaptándose a las necesidades de los estudiantes y promoviendo la resolución creativa de problemas. Esta materia refleja la importancia de formar a futuros educadores capaces construir y analizar herramientas para la enseñanza de redes de telecomunicaciones e infraestructura.

En su conjunto, estas materias no solo proporcionan un conocimiento técnico sólido, sino que también incitan a la reflexión sobre la historia, la evolución y el impacto social y económico de la informática en la sociedad moderna. Al preparar a los estudiantes con una comprensión integral de las tecnologías informáticas, desde sus cimientos hasta sus aplicaciones prácticas y su influencia cultural, estas materias contribuyen de manera significativa a la formación de profesionales capacitados y con una visión holística en el campo de la informática.

### **Tecnología, Sociedad y Ciudadanía Digital**

En esta área se encuentra un único curso que tiene por objetivo ayudar a desarrollar una comprensión sobre una serie de saberes en relación a los efectos socio técnicos de las tecnologías digitales en el cotidiano de los ciudadanos.

En términos de contenidos la materia se construye sobre un conjunto de ejes principales: a) ciencia, técnica y tecnología b) las tecnologías digitales como nuevos elementos ordenadores de vida cotidiana y c) construcción de ciudadanía en la era digital.

Así, temas como la relación tecnología-sociedad y las distintas formas de determinismos comienzan el debate en el curso. También el reconocimiento de la necesidad de desarrollo de una soberanía científica y tecnológica (y hasta tecno pedagógica). En el segundo eje, se abordan temas en relación al poder y control con tecnologías digitales en un mundo donde la informacionalización, la automatización y la plataformización son elementos principales que traccionan nuestro presente desde casi cualquier dimensión. Finalmente, en construcción de ciudadanía en la era digital, se abordan temas en relación a los datos personales, su captura y comercialización, ética y algoritmos, riesgos en la red, convivencia digital, voto electrónico, entre otros.

### **Ciencia de Datos**

En el área de ciencia de datos encontramos tres materias que se encuentran vinculadas directamente con temáticas muy actuales: Bases de datos, Ciencia de Datos e Inteligencia Artificial y Didáctica de la Programación IV.

Estas disciplinas emergen como fundamentales en un mundo cada vez más orientado hacia la automatización, el análisis predictivo y la toma de decisiones basada en datos. Las primeras dos materias (Bases de datos, Ciencia de Datos e Inteligencia Artificial) brindan a las y los estudiantes herramientas y técnicas avanzadas para analizar grandes conjuntos de datos, descubrir patrones, generar modelos predictivos y extraer conocimientos valiosos. En la materia inicial de Bases de datos se presentan las diferencias entre los distintos modelos, y se trabaja con consultas SQL en bases relacionales, sentando las bases para trabajar con conceptos y herramientas más avanzados en la siguiente materia que se centra en la exploración de grandes volúmenes de datos. Asimismo, al trabajar sobre la intersección de la Ciencia de Datos y la Inteligencia Artificial con el ejercicio de la ciudadanía y la moral tecnocientífica vigente, se busca despertar una ciudadanía crítica. *"Se trata de dotar a las y los estudiantes de herramientas para que formulen sus propias preguntas críticas, incisivas, incluso suspicaces. Que su reflejo sea dudar de las promesas mágicas de la tecnología: de la actual y de la futura, aprendiendo a rechazarla cuando lo que se pierde sea más que lo que se gana, pero sin por eso dejar de utilizarla cuando lo crean conveniente."* (Schapachnik, F., y Bonello, M. B., 2022, p. 179).

La Didáctica de la Programación IV, hace foco en la intersección de Ciencia de Datos, Tecnología, Sociedad y Ciudadanía Digital. Introduce a las y los cursantes para que puedan repensar el impacto de las tecnologías digitales en la transformación pedagógica del sistema escolar desde un posicionamiento crítico. A partir de los contenidos propuestos se pretende que puedan analizar experiencias educativas que incluyan las tecnologías atendiendo a transformaciones digitales superadoras, reconociendo su impacto en los materiales didácticos y su integración al currículum. Pueden estudiarse propuestas didácticas basadas en herramientas de visualización y manipulación de datos, modelos amigables de aprendizaje automático, proyectos de análisis de datos, etc. Se espera que se identifiquen áreas de vacancia y propongan estrategias de abordaje de dichos contenidos para destinatarios específicos.

Esta área curricular pretende que las y los estudiantes releven de manera comparada las propuestas didácticas y recursos para abordar el impacto del uso de la tecnología digital en la sociedad. Se propone un enfoque que incorpore a este análisis la dimensión disciplinar técnica (del área y del resto de la carrera) para completar la explicación de cómo y por qué se dan estos fenómenos. En este sentido, se priorizan actividades y recursos que pongan en práctica la idea de que las CC permiten comprender mejor el mundo en el que vivimos. Por ejemplo, los impactos reales del uso de las ciencias de datos en términos de percepción y comprensión de la realidad, acceso a la información, construcción de discursos de odio, etc.

En conjunto, estas materias juegan un papel esencial en la configuración del futuro de la informática y su impacto en la sociedad. Tenemos claro que nuestra misión se enfoca en formar profesionales de la educación con sólidos conocimientos en Ciencias de la Computación, más que profesionales que van a ejercer la disciplina en ambientes de diseño y desarrollo empresarial. Esta situación nos obliga a repensar muchas de las estrategias y materiales educativos que diseñamos y/o seleccionamos; es un desafío que tenemos por delante y que asumimos en beneficio de tratar de colaborar con políticas públicas educativas que permitan un desarrollo más justo e igualitario.

### **Consideraciones Finales**

Las CC son una disciplina consolidada, que cuentan con una vasta bibliografía y una comunidad científica e industrial que soporta y continúa realizando aportes y avances constantemente.

En los últimos años hemos visto cómo la pandemia por COVID-19 y la aparición de herramientas con aprendizaje automático (como *ChatGPT*) han acelerado y revolucionado el mundo de la educación y la forma de ejercer nuestra ciudadanía, entre otras áreas de la vida social. Vemos cómo constantemente se busca informatizar la relación con el estado, muchas veces sin comprender las implicancias de dicho "avance".

La incorporación de una materia en la escolaridad obligatoria, en el marco de la cual a partir del aprendizaje de la programación se pueden problematizar distintos aspectos de la sociedad en la que vivimos, es una oportunidad para formar una ciudadanía crítica y comprometida. La República Argentina ya inició el proceso y se ve reflejado el cambio de curricula en distintas jurisdicciones en cada uno de los niveles de la escolaridad obligatoria.

La formación integral de nuestro estudiantado en los aspectos disciplinares de las CC, con una mirada constante sobre la relación de la tecnología con la sociedad, nos permite formar profesoras y profesores que desarrollarán sus prácticas profesionales reconociendo sus derechos y deberes ciudadanos, respetuosos de la dignidad humana y responsables de sus actos profesionales.

Confiamos que la profundidad en el abordaje de los conceptos computacionales, les permitirá una rápida adaptación al uso y comprensión de nuevas tecnologías, problematizando sobre su impacto en diferentes contextos; tomando decisiones sobre aspectos de planificación, coordinación y evaluación de procesos de enseñanza y de aprendizaje, y sobre la elaboración y actualización de contenidos.

Por último, la implementación de un nuevo profesorado en un área tan cambiante en cuanto a la tecnología utilizada, implica la creación de materias con contenido que puede cambiar de un

cuatrimestre a otro. Nos impulsa a continuar el proceso de mejora sobre el dictado y el material utilizado de forma periódica. De esta manera, el profesorado se enmarca en un proceso de investigación/acción, de trabajo iterativo en permanente revisión, análisis y ajuste como un laboratorio dinámico y vivo en el que docentes y estudiantes construyen saberes entre las ciencias, la actualidad y el contexto.

## Referencias

- Bell, T. (2014). Establishing a nationwide CS curriculum in New Zealand high schools, en *Communications of the Association for Computing Machinery*, 57, pp. 28–30.
- Borchard, M. y Roggi, I. (2017). Ciencias de la computación en los sistemas educativos de América Latina, Unesco.  
<https://unesdoc.unesco.org/ark:/48223/pf0000372138>
- Brassesco, M. V. (2023). Introducción a la didáctica de la programación para un profesorado universitario en Informática. *Simposio Argentino de Educación en Informática (SAEI) en las Jornadas Argentinas de Informática (JAIIO 2023)*.  
<https://ojs.sadio.org.ar/index.php/JAIIO/issue/current>
- Brassesco, M. V., Del Dago, G., y Urso, M. (2021). Zamba: Una plataforma de aprendizaje como soporte a conceptos computacionales. *Jornadas Argentinas de Didáctica de las Ciencias de la Computación*, 315.  
<https://jadicc2021.program.ar/wp-content/uploads/2022/03/actas-jadicc-2021.pdf>
- Castells, M. (2001). Epílogo: Informacionalismo y la Sociedad Red. En Pekka Himanen. *La ética hacker y el espíritu de la era de la información*, 110-124. Destino.
- Consejo Federal de Educación (2015) Res 263/15. Resoluciones CFE, 2015. Consejo Federal de Educación (2018) Res 343/18. Resoluciones CFE, 2018.
- Echeveste, M. (2022). La búsqueda por el sentido a aprender a programar y su relación con la forma en la que se enseña en las escuelas. *Simposio Argentino de Educación en Informática (SAEI) en las Jornadas Argentinas de Informática (JAIIO 2022)*.  
<https://publicaciones.sadio.org.ar/index.php/JAIIO/article/download/410/345/>
- Fundación Sadosky (2013) CC – 2016. Una propuesta para refundar la enseñanza de la computación en las escuelas Argentinas.  
<http://www.fundacionsadosky.org.ar/wp-content/uploads/2014/06/cc-2016.pdf>
- Fundación Sadosky (2023). Aprendizajes y desafíos para la enseñanza de las Ciencias de la Computación en las escuelas. La iniciativa Program. AR de Argentina  
[https://fundacionsadosky.org.ar/wp-content/uploads/2023/07/ProgramAR\\_CAF\\_10.pdf](https://fundacionsadosky.org.ar/wp-content/uploads/2023/07/ProgramAR_CAF_10.pdf)
- Furber, S. (2012) Shut down or restart? The way forward for computing en *UK schools. Technical Report The Royal Society*, London.
- European Education and Culture Executive Agency (2022) Informatics education at school in Europe.  
<https://eurydice.eacea.ec.europa.eu/publications/informatics-ducation-school-europ>

K-12 Computer Science Framework Steering Committee (2016) The K-12 Computer Science Framework. ACM.

Schapachnik, F., y Bonello, M. (2022). *Ciencias de la Computación en la escuela: Guía para enseñar mucho más que a programar*. Siglo XXI Editores.

Vee, A. (2017) *Coding Literacy. How Computer Programming Is Changing Writing*. The MIT Press

Zukerfeld, M. (2020). Bits, plataformas y autómatas. Las tendencias del trabajo en el capitalismo informacional en *LAT Revista Latinoamericana de Antropología del Trabajo*, 4(7), 1-50

## **Mejorando la enseñanza en situaciones de ACSC síncronas basadas en texto mediante reconocimiento de emociones**

Germán Lescano<sup>1,2</sup>, Daniela Missio<sup>1</sup>, Rosanna Costaguta<sup>1</sup>  
gelescano@unse.edu.ar, daniela\_mmi@yahoo.com.ar, rosanna@unse.edu.ar

<sup>1</sup> Instituto de Investigación en Informática y Sistemas de Información  
Universidad Nacional de Santiago del Estero

<sup>2</sup> Consejo Nacional de Investigaciones Científicas y Técnicas

### **Resumen**

En una experiencia de Aprendizaje Colaborativo Soportado por Computadora (ACSC) se asume la existencia de un grupo de estudiantes que trabajan juntos mediados por computadora con el objetivo de aprender. Se reconoce que el ACSC promueve el aprendizaje activo y la motivación de los estudiantes por su propio aprendizaje, incrementa la autoestima en los participantes, al mismo tiempo que promueve el desarrollo de habilidades sociales y de comunicación.

Muchos factores pueden afectar negativamente el funcionamiento de un grupo impidiendo que la experiencia de ACSC sea exitosa. Uno de esos factores son las emociones. Se reconoce que la cognición y la emoción están relacionadas; pueden influir sobre el desempeño de una persona y juegan un rol esencial en la atención y en la memoria, en el juicio, en la toma de decisiones, y en la resolución de problemas creativos. Los estudiantes que se encuentran en estado de ansiedad, enojados, o molestos pueden tener dificultades para aprender y no se desempeñan adecuadamente. Existen investigaciones que demuestran que niveles apropiados de emociones positivas facilitan el aprendizaje al mejorar la motivación, la tolerancia al riesgo en la toma de decisiones, el esfuerzo y la persistencia.

En las clases presenciales el docente tiene acceso a la información que brindan las señales afectivas sobre la situación de sus alumnos ya que puede observar cuando están aburridos y cambiar la estrategia de enseñanza, o llamarles la atención. En las experiencias de ACSC síncronas basadas en texto, esta información se pierde debido a que el medio utilizado no la transmite, lo que ocurre por ejemplo con las expresiones faciales. Esto dificulta la labor de un docente. Además, la ausencia de información afectiva no solo afecta al docente en el proceso de enseñanza y a los estudiantes en sus procesos de aprendizaje, también afecta al clima emocional y al desempeño del grupo.

En este trabajo se describe un modelo que reconoce emociones en tiempo real en situaciones de ACSC síncronas basadas en texto y permite dar soporte a la enseñanza en estos tipos de entornos. El modelo fue validado con experiencias reales de ACSC en asignaturas vinculadas con ciencias de la computación. Los resultados proporcionados por el modelo son prometedores.

**Palabras Claves:** Aprendizaje colaborativo soportado por computadora, Comunicación síncrona, Reconocimiento automático de emociones en texto, Soporte para la enseñanza, Ciencias de la Computación.

## Introducción

En el Aprendizaje Colaborativo Soportado por Computadora, en adelante ACSC, los estudiantes se agrupan para lograr un objetivo académico empleando computadoras para dar soporte a los procesos colaborativos. Colocar a los alumnos en grupos no garantiza que aprendan, sino más bien es necesario que trabajen correctamente (Soller, 2001).

Muchos factores pueden incidir en que el proceso de aprendizaje colaborativo no sea el apropiado, tales como, los procesos de negociación, los estilos de participación, las opiniones divergentes, la mediación de la tecnología, la tarea, los conflictos, las emociones, entre otros (García-prieto et al., 2003; Heerdink et al., 2013; Järvenoja & Järvelä, 2013; Jones & Issroff, 2005; Kreijns et al., 2003; Nam, 2014).

En estos tipos de entornos, la comunicación cara-a-cara disminuye considerablemente o incluso pueden estar ausentes debido a la mayor preferencia por usar medios textuales tales como los correos electrónicos, los mensajes de texto, herramientas de chats, foros, etc. Los medios electrónicos frecuentemente limitan la cantidad de intercambios no verbales entre los miembros del equipo, de modo que el éxito y la efectividad de los equipos distribuidos, es mucho más sensible a la habilidad de los miembros en el uso de tecnologías para comunicarse y colaborar.

Tener en cuenta estos aspectos es particularmente importante sobre todo en asignaturas en las que se reconoce la necesidad de incrementar la tasa de alumnos que aprueban y la continuidad con sus estudios, como es el caso de los cursos introductorios en ciencias de la computación. Se ha encontrado que los alumnos que se encuentran en estados emocionales positivos tienen mayores oportunidades de terminar satisfactoriamente los cursos introductorios en ciencias de la computación (Bennedsen & Caspersen, 2008). Por este motivo, hacer un seguimiento de los grupos de aprendizaje de estudiantes de las carreras vinculadas con Ciencias de la Computación puede ser relevante para contribuir con el incremento de las tasas de alumnos que aprueban sus exámenes finales.

Es importante que en los grupos colaborativos el docente pueda realizar el seguimiento del proceso e intervenir cuando sea necesario. Este seguimiento realizado por los docentes en situaciones de ACSC se puede efectuar a través del análisis de métricas. Debido a que el manejo adecuado de las emociones resulta fundamental en los procesos de comunicación, para promover en los alumnos habilidades para gestionar sus emociones, es de fundamental importancia que el docente pueda tener acceso a esta información

afectiva cuando se trabaja en entornos mediados por computadora. El docente podría inferir emociones de sus alumnos al analizar, por ejemplo, los comportamientos de los mismos. Sin embargo, en grupos numerosos este tipo de seguimiento es un gran trabajo para el docente. Por este motivo, una solución que automáticamente reconozca emociones en los alumnos puede ser de gran ayuda para los docentes.

En este trabajo se propone un modelo que permite reconocer estados emocionales en los alumnos que interaccionan en situaciones de ACSC síncronas basadas en texto, a fin de proporcionar al docente esta información afectiva de modo que pueda emplearla para realizar ajustes en su proceso de enseñanza. El modelo fue validado con experiencias reales de ACSC en asignaturas vinculadas con ciencias de la computación. La validación realizada al modelo arroja resultados prometedores.

### **Antecedentes**

En general, distintos métodos fueron empleados para analizar emociones en entornos de ACSC, algunos cuantitativos y otros cualitativos (Capdeferro & Romero, 2012). De acuerdo con los autores anteriormente mencionados, los enfoques que predominan como fuente de análisis de las emociones son los auto-reportes, los archivos de logs generados por los alumnos, y los mensajes generados en las plataformas de aprendizaje. Según (Turner et al., 2003) también se aplica el recuerdo estimulado, los reportes en línea, el muestreo de experiencias, y las entrevistas.

Desde un punto de vista holístico, (Wosnitza & Volet, 2005) reconocen tres tipos de enfoques para analizar datos sobre emociones en entornos de aprendizaje: los de tipo *snapshot* que permiten medir las emociones inmediatamente antes o después del proceso de aprendizaje; los que estimulan el recuerdo para analizar emociones una vez que concluye el proceso de aprendizaje; y los que permiten medirlas durante el proceso de aprendizaje.

### **Enfoques tipo Snapshot**

En esta categoría se incluyen los cuestionarios diseñados de manera específica para estudiar emociones. Generalmente estos cuestionarios se responden a través de escalas de tipo Likert, una escala psicométrica, que especifica el nivel de acuerdo o desacuerdo manifestado por una persona con respecto a declaraciones presentadas en los cuestionarios. Estas herramientas no permiten tener acceso directo a las emociones que se desencadenan durante el proceso de aprendizaje, lo cual significa que el proceso de activación emocional es desconocido. Otra desventaja de este enfoque es que se basan en el auto-reporte.

Ejemplos de esta categoría son: el cuestionario de emociones académicas (Pekrun et al., 2002); la escala de test de ansiedad (Pintrich et al., 1991); el cuestionario para analizar fuentes de frustración en entornos de ACSC (Capdeferro & Romero, 2012); el cuestionario para evaluar sentimientos hacia el entorno de aprendizaje (Gunawardena, 1995); y el cuestionario para evaluar sentimientos (entusiasmo, excitación, exaltación, ánimo, frustración, enojo, irritación) (Levin et al., 2010).

### **Enfoques que estimulan el recuerdo**

Estos enfoques se basan en presentar al alumno un recuerdo de una actividad de aprendizaje reciente (por ejemplo, mediante audio o video) y generar una reflexión sobre episodios seleccionados que se espera contengan actividad emocional. La selección se puede basar en indicadores visibles de la actividad emocional, o en elegir directamente aquellos en los que se considera que la activación emocional tomó lugar.

La ventaja de los enfoques que estimulan el recuerdo es que no interrumpen el proceso de aprendizaje y proveen la oportunidad de sondear el proceso de evaluación, aunque se base en datos de auto-reporte. Ejemplos de esta categoría son: el enfoque de viñeta y el paradigma de incidentes críticos (Heerdink et al., 2013).

Un software para viñetas, conocido como Bubble Dialogue es analizado por (Jones & Issroff, 2005). Estos investigadores señalan que este software está siendo usado para evaluar y soportar el desarrollo de habilidades sociales y emocionales en poblaciones de niños con necesidades particulares, tales como dificultades de comportamiento y emocional, y niños con autismo. Los usuarios pueden representar los pensamientos de un personaje, así como su voz, lo cual permite a los participantes, a través de sus personajes, revelar sus pensamientos y las visiones que pueden estar ocultas o ser difíciles de acceder.

### **Enfoques que miden emociones durante el proceso de aprendizaje**

En esta categoría se agrupan los métodos que además de realizar el análisis del proceso de aprendizaje, las reflexiones y las experiencias, también permiten realizar un análisis en función del tiempo. Estos métodos subsanan las desventajas de los métodos de tipo *snapshot* al poder medir las emociones durante el proceso de aprendizaje, incluso en algunos casos posibilitan tener acceso al proceso de activación emocional.

(Wosnitza & Volet, 2005) reconocen cuatro métodos que entran en esta categoría. El primero se basa en la observación de expresiones faciales. Una herramienta muy usada para analizar las expresiones faciales es el Sistema de Codificación de Acciones Faciales (FACS) desarrollado por (Ekman & Friesen, 1978). Para este caso, la evaluación y el proceso de activación emocional no puede ser visible pero las emociones pueden ser observadas en las expresiones faciales. El segundo método consiste en el análisis de contenido basado en texto. Este método tiene la ventaja de no interrumpir el proceso de aprendizaje. Sin embargo, el análisis basado en texto presenta como limitación que la emoción sólo puede ser observada, aunque los elementos de evaluación emocional y su proceso de activación puedan estar disponibles, dejando en manos del investigador que infiera el proceso de desarrollo de las emociones. Otra desventaja es que el análisis del contenido está inevitablemente afectado por la interpretación del investigador. El tercer método se basa en pedir a los alumnos que mantengan un diario emocional donde escriben sobre sus experiencias emocionales del día. Este instrumento puede ser usado para investigar emociones en

procesos de aprendizaje más largos. Se reconoce como limitación de este método, el hecho de basarse en el auto-reporte. Si bien este instrumento no provee un acceso directo a las emociones que se desencadenan, se reconoce que el proceso de medición es más cercano al proceso que lo desencadena comparado con la técnica del *snapshot*. El cuarto método está relacionado con el muestreo de la experiencia y el tiempo, lo que generalmente involucra completar un cuestionario online a la mitad de una actividad. El cuestionario suele ser breve y presenta al alumno varias situaciones durante el proceso de aprendizaje. Una desventaja asociada con este método es que el proceso de activación y evaluación emocional permanece limitado y puede necesitar ser inferido a partir de otras fuentes de información. Otra desventaja es que interrumpe el proceso de aprendizaje dependiendo del auto-reporte de los datos.

Respecto a los enfoques que miden emociones durante el proceso de aprendizaje, se hicieron algunas contribuciones en dos modalidades, desde el punto de vista computacional, tendientes a facilitar el reconocimiento automático: métodos basados en el análisis de señales bio-fisiológicas y otros basados en el análisis del comportamiento motor. Los métodos del primer tipo permiten reconocer emociones en base al análisis de las señales eléctricas que producen el cerebro, el corazón, los músculos y la piel. Ofrecen la ventaja de que, al ser señales involuntarias, posibilitan medir los estados afectivos de manera continua y reconocerlos tan pronto como ocurren. La desventaja más importante de estos métodos es que son invasivos, dificultando la experiencia del usuario (Feidakis, 2016). Además, requieren de equipamiento especializado que generalmente es costoso. Los métodos del segundo tipo permiten reconocer emociones analizando comportamiento y cambios físicos en el cuerpo de quien comunica la experiencia emocional. Estos métodos ofrecen la ventaja de poder evaluar estados afectivos a partir del uso de dispositivos generalmente disponibles para los alumnos, como ser cámaras, micrófonos, mouse y teclado (Feidakis, 2016).

Los antecedentes encontrados respecto al reconocimiento de emociones en entornos de ACSC pueden agruparse bajo el uso de herramientas de *awareness* emocional o visualización emocional. Las herramientas que proporcionan *awareness* emocional grupal permiten a los alumnos reconocer sus propias emociones de manera que puedan evaluarlas y regularlas conforme a las necesidades del proceso colaborativo (Daradoumis et al., 2013). Estas herramientas en principio no señalan al alumno qué realizar, sino más bien se basan en un enfoque reactivo al colocar el foco de control en las manos del alumno para que pueda interpretar la información que se le provee y opere con tal información para mejorar la colaboración (Miller & Hadwin, 2015). Sin embargo, existen antecedentes que proponen la posibilidad de realizar recomendaciones a los alumnos, de manera complementaria, para ayudarlos a regular sus emociones (Argueda et al., 2015).

### **Enfoque Propuesto**

El presente trabajo se asienta, por un lado, sobre el reconocimiento de la importancia que las emociones detentan sobre los aprendizajes de los estudiantes, y por otro, en la amplia experiencia de los autores como docentes en situaciones de ACSC, particularmente usando la aplicación

COLLAB. COLLAB da soporte a la comunicación, la cooperación, la coordinación y la colaboración de los estudiantes y docentes cuando desarrollan actividades de aprendizaje colaborativo basadas en texto (Lescano & Costaguta, 2018). Esta aplicación actualmente funciona desde servidores de la FCEyT-UNSE (<http://chat.fce.unse.edu.ar/chat/web/index.php>) proporcionando un entorno propicio para la interacción y el trabajo en equipo de grupos de aprendizaje, y dejando registro de las interacciones textuales de los grupos.

Por lo dicho, en este trabajo se propone un modelo para reconocer emociones en situaciones de ACSC síncronas basadas en texto aplicando técnicas de aprendizaje de máquina tradicionales. Para efectuar la construcción sistemática del modelo se aplicó la metodología CRISP-DM.

A fin de emplear minería de datos en este trabajo se ejecutó un trabajo experimental con el objetivo de adquirir datos de interacciones de alumnos en situaciones de ACSC empleando herramientas síncronas basadas en texto. Para generar los datos se solicitó la colaboración a profesores universitarios relacionados con carreras en Ciencias de la Computación. Participaron de esta experiencia profesores pertenecientes a dos universidades de Argentina (Universidad Nacional de Santiago del Estero y la Universidad Católica de Santiago del Estero) y dos universidades de Colombia (Corporación Universitaria Comfacauca y la Institución Universitaria Centro de Estudios Superiores).

En estas experiencias de ACSC, los docentes decidieron el criterio para formar los grupos y las actividades que debían realizar los mismos. Las actividades fueron de naturaleza abierta y relacionadas con el desarrollo de producciones escritas (desarrollo de trabajos prácticos). Participaron 143 estudiantes que usaron la aplicación *COLLAB* para comunicarse con sus compañeros de grupo a fin de realizar las actividades asignadas. En estas experiencias realizadas, los docentes intervinieron cuando era necesario, por ejemplo, cuando los grupos no trabajaban. Generalmente, las tareas asignadas tuvieron una semana como plazo de realización.

### **Comprensión de los datos**

La primera etapa en la metodología CRISP-DM consiste en analizar los datos disponibles. De acuerdo con los datos recolectados de las situaciones de ACSC, el 70.27% de los grupos estuvo integrado por tres alumnos, el 16.22% por dos integrantes, el 10.81% por cuatro integrantes, y el 2.7% por cinco integrantes.

El conjunto de interacciones fue analizado aplicando análisis de contenido siguiendo la metodología de Krippendorff (Krippendorff, 2004). El análisis fue analizado por una misma persona aplicando test-retest. Para realizar este test-retest, el etiquetado realizado inicialmente fue comparado por otro etiquetado realizado por la misma persona seis meses después. El análisis consistió en asignar a cada interacción una etiqueta que indica la emoción manifestada en la interacción. Para asignar la etiqueta se tuvo en cuenta el esquema de clasificación de emociones basada en la teoría de control-valor planteada por Pekrun (Pekrun, 2014). El conjunto de interacciones contabiliza un total

de 7166 interacciones. La confiabilidad del etiquetado fue de  $\alpha = 0.86$ .

Pekrun agrupa las emociones relevantes para el aprendizaje en cuatro grupos: a) Emociones de logro. Están relacionadas a las actividades de logro y al éxito o fallo que se derivan de este tipo de actividades. Estas emociones son el disfrute del aprendizaje; la esperanza o el orgullo relacionado al éxito; y la ansiedad o vergüenza asociado al fracaso, y la desilusión. Las emociones de logro están siempre presentes en las configuraciones académicas, especialmente cuando la importancia del éxito o fracaso se hace saber a los estudiantes; b) Emociones epistémicas. Son generadas por situaciones cognitivas. En esta categoría se tiene como emociones la sorpresa ante una nueva tarea; la curiosidad, la confusión y la frustración ante obstáculos; el disfrute cuando el problema es resuelto. Se considera que las emociones epistémicas son importantes cuando las tareas de aprendizaje no son rutinarias; c) Emociones de tópico. Están relacionadas con los tópicos presentados en la clase. Las emociones en esta categoría son el agrado y el desagrado. Las emociones positivas y negativas hacia el tópico pueden influir en el interés de los estudiantes respecto a los materiales de aprendizaje; y d) Emociones sociales. Se vinculan con las relaciones que se establecen entre los alumnos y entre los alumnos y docentes. En esta clase se tiene las emociones de amor, simpatía, compasión, rebeldía, envidia, ansiedad social, enojo y rebeldía.

Del total de interacciones analizadas, sólo en el 25% fue posible inferir presencia de emociones. Esto puede deberse, a priori, a las características propias del grupo de estudiantes participantes en la experimentación (estudiantes avanzados de carreras de informática). Quizás también pueda tratarse de estudiantes que lograron aprender buenas prácticas de interacción con otros durante sus procesos de formación. En la Figura 1 se muestra el porcentaje de interacciones que responden a cada tipo de emoción.

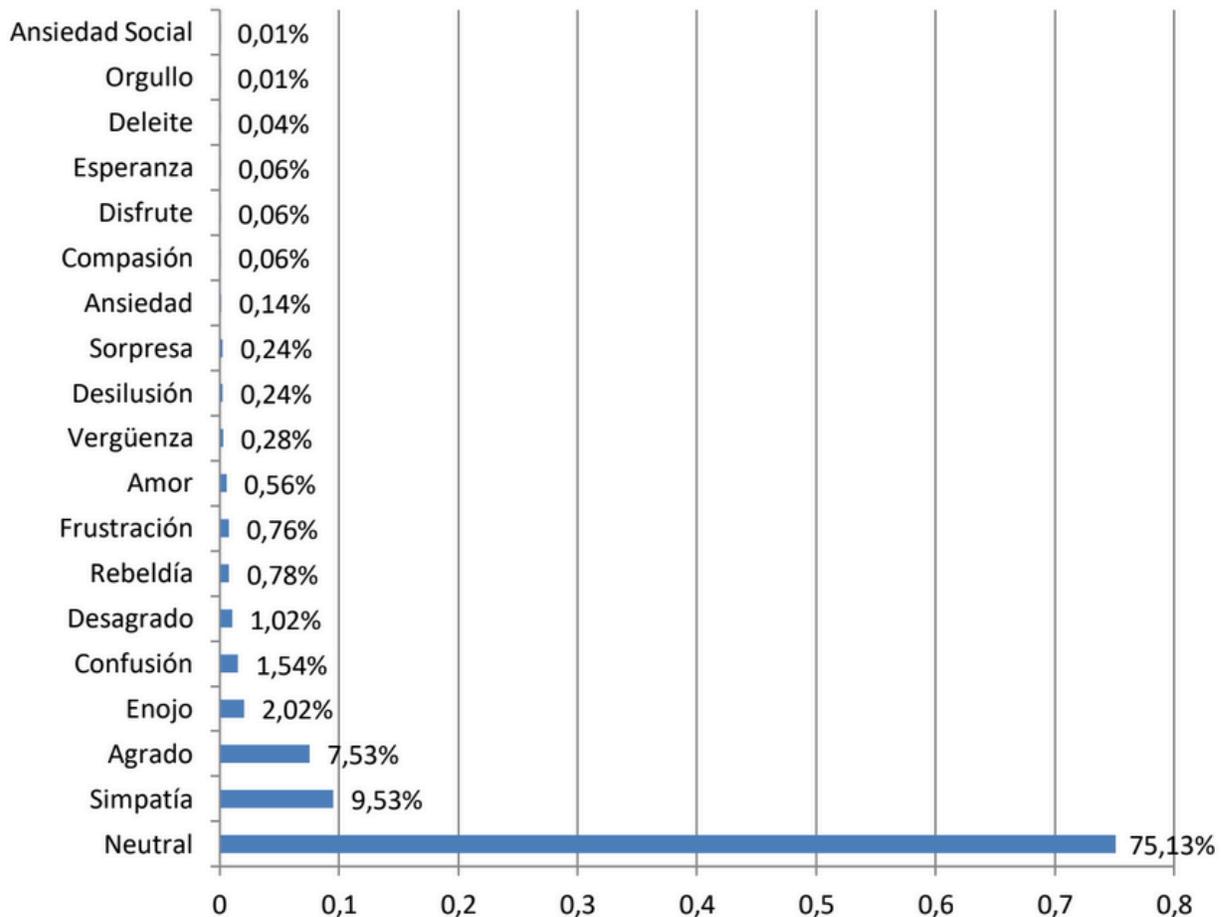


Figura 1. Porcentaje de emociones reconocidas en las interacciones de ACSC recolectadas.

### Preparación de los datos

A fin de poder descubrir patrones en datos textuales es necesario transformarlos para poder aplicar técnicas de minería de datos. La hipótesis fundamental que se considera es que la forma en que los alumnos se expresan permite reconocer emociones. Por esta razón, la característica principal que se tiene en cuenta son las palabras. Se supone que la co-ocurrencia de palabras pueden ser indicadores de manifestación de emociones.

Como parte del proceso de preparación de los datos se procedió a la *tokenización* de los textos, se eliminaron *stop-words*, se obtuvieron los lemas de cada palabra, y se reemplazaron la presencia de emoticones por códigos debidamente definidos. Si bien las palabras en sí son características importantes en el procesamiento de lenguaje natural, adicionalmente se consideraron otras características. Por un lado, el manejo de la negación. Para el manejo de la negación se tuvo en cuenta la propuesta de (Padmaja et al., 2014) que sugiere definir una cantidad  $n$  de manera tal que cada vez que ocurra una negación se nieguen las  $n$  palabras que aparecen a continuación de la negación.

	<b>Actividades Relacionadas a la Tarea</b>		<b>Actividades Sociales</b>
<b>Ejecución</b>	<ul style="list-style-type: none"> <li>• Intercambio información (TaskExch)</li> <li>• Realización interrogantes (TaskQues)</li> </ul>	de  de	<ul style="list-style-type: none"> <li>• Saludos (SociGree)</li> <li>• Apoyo o Soporte Social (SociSupp)</li> <li>• Resistencia Social (SociResi)</li> <li>• Comprensión Compartida (SociUnd+)</li> <li>• Pérdida de Comprensión Compartida (SociUnd-)</li> </ul>
<b>Coordinación/Regulación</b>	<ul style="list-style-type: none"> <li>• Planeación de la tarea (MTaskPlan)</li> <li>• Monitoreo del progreso de la tarea (MTaskMoni)</li> <li>• Evaluaciones positivas del progreso de la tarea (MTaskEvl+)</li> <li>• Evaluaciones negativas del progreso de la tarea (MTaskEvl-)</li> </ul>		<ul style="list-style-type: none"> <li>• Planear la colaboración (MSociPlan)</li> <li>• Monitorear el proceso grupal (MSociMoni)</li> <li>• Evaluaciones positivas del proceso grupal (MSociEvl+)</li> <li>• Evaluaciones negativas del proceso grupal (MSociEvl-)</li> </ul>
<b>Otras</b>	<ul style="list-style-type: none"> <li>• Observaciones técnicas neutrales (TechNeut)</li> <li>• Observaciones técnicas positivas (TechPosi)</li> <li>• Observaciones técnicas negativas (TechNega)</li> </ul>		<ul style="list-style-type: none"> <li>• Otras/ sin sentido (Other)</li> </ul>

Tabla 1. Esquema de clasificación de actividades colaborativas propuesta por (Janssen et al., 2012)

Otra característica considerada fue el tipo de actividad colaborativa que refleja una interacción. Para considerar el tipo de actividad colaborativa se tuvo en cuenta la clasificación propuesta por (Janssen et al., 2012). Estos autores distinguen entre actividades orientadas a la tarea y actividades sociales. A su vez, considera que las actividades pueden estar involucradas con las fases de ejecución, las fases de regulación o monitoreo, y otras no importantes. En la Tabla 1 se muestra una tabla de doble entrada en la cual se muestra la clasificación de las actividades según sean orientadas a la tarea o sociales, y al momento en que se producen (ejecución, monitoreo u otra). En la Figura 2 se muestra la distribución de las emociones por tipo de actividad colaborativa.

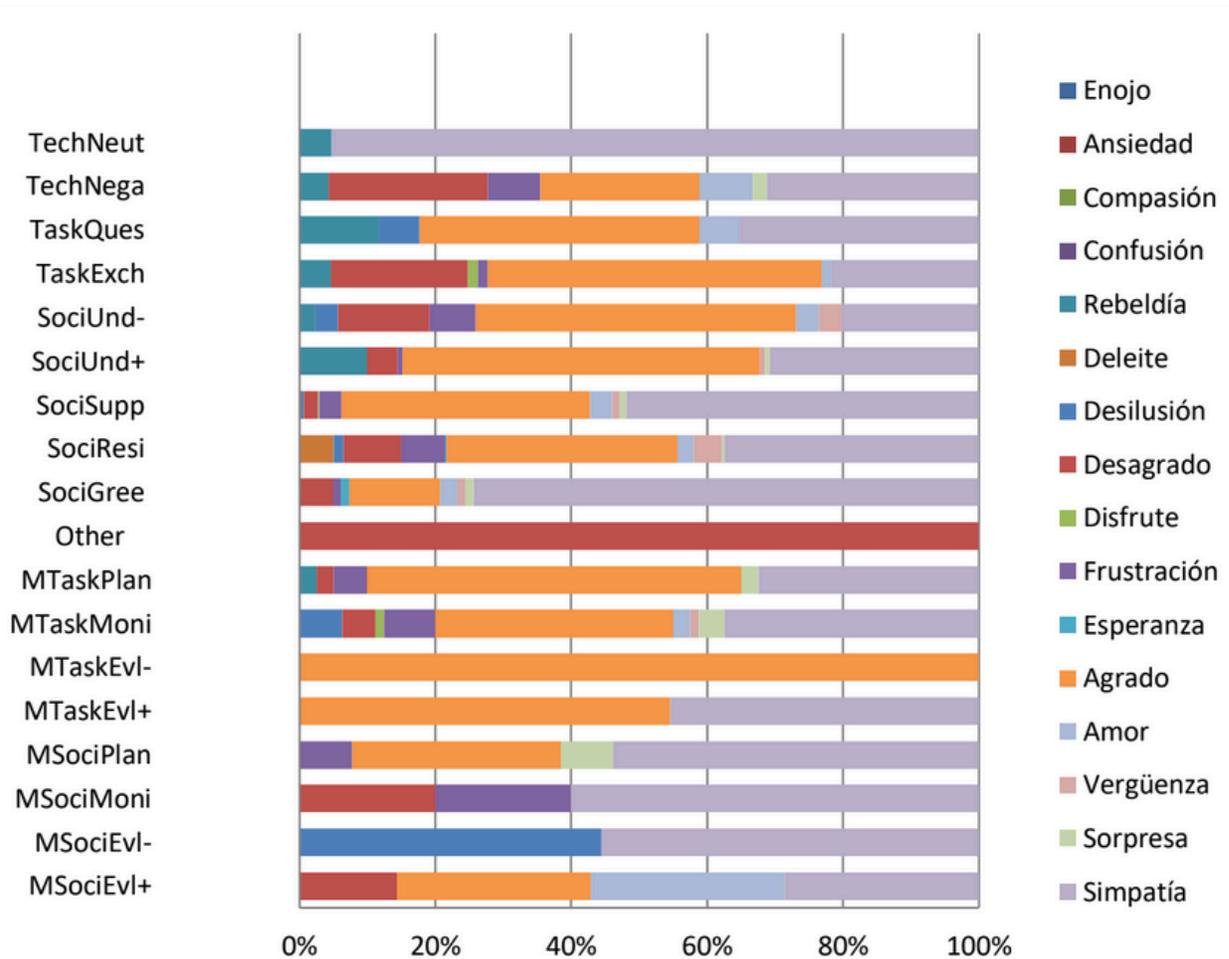


Figura 2. Distribución de las emociones por tipo de actividad colaborativa

Otras características empleadas son considerar el hecho de si en la interacción se formulan expresiones que reflejan problemas de tecnología, problemas de comunicación, o de alejamiento de objetivos. Estas tres características son de tipo dicotómicas y surgen de la revisión bibliográfica porque coincide con factores involucrados en la generación de emociones (Capdeferro & Romero, 2012; Pekrun & Linnenbrink-Garcia, 2012; Robinson, 2013).

### Modelado

En este trabajo se evaluaron tres modelos construidos cada uno con una técnica de minería de datos. Las técnicas de minería de datos seleccionadas fueron: regresión logística, máquinas de vector de soporte, y redes neuronales. La selección de estas técnicas radica en el hecho de que son apropiadas para trabajar con datos de tipo textuales.

Las Tablas 2, 3 y 4 detallan los distintos valores que se utilizaron como parámetros para los modelos de regresión logística, la máquina de vector de soporte (SVM) y el perceptrón multicapa, respectivamente. En la Tabla 4, respecto al parámetro capas, por ejemplo, el valor (5) indica la cantidad de neuronas en la capa oculta. En ese caso la red tiene tres capas: la capa de entrada, la

capa oculta (con 5 neuronas) y la capa de salida. En el caso de la configuración de capas (15, 20) significa que tiene dos capas ocultas, una con 15 neuronas y la otra con 20 neuronas. Por lo tanto, esa red está constituida por 4 capas (la de entrada, las dos ocultas y la capa de salida).

Parámetro	Valores considerados
C	(0.1, 0.5, 1.0,1.5)
Tolerancia	(0.001, 0.1,1.0)

Tabla 2. Valores usados como parámetros para evaluar los modelos generados con regresión logística

Parámetro	Valores considerados
C	(0.1, 0.5, 1.0,1.5)
Tolerancia	(0.001, 0.01,0.1, 1.0)
Kernel	(lineal, sigmoide, rbf)

Tabla 3. Valores usados como parámetros para evaluar los modelos generados con SVM

Parámetro	Valores considerados
Capas	(5,),(10,),(15,), (5,10),(15,20))
Cantidad de iteraciones	500
Tolerancia	(0.001, 0.1,1.0)
Función de activación	Tangente hiperbólica, función logística, función relu, función identidad
$\lambda$	(0.001, 0.1, 1)

Tabla 4. Valores usados como parámetros para evaluar los modelos generados con el perceptrón multicapa

### Evaluación del Modelo Propuesto

En esta sección se presenta la evaluación de los modelos planteados. La evaluación es una etapa en el proceso de descubrimiento de conocimiento y tiene por objetivo verificar el desempeño del modelo para la tarea por el cual fue construido. Las métricas utilizadas para realizar la evaluación son la *precisión*, el *recall* y el *F1*.

Dado la escasa cantidad de interacciones recolectadas para cada tipo de emoción (Figura 1), en la evaluación se hizo foco en aquellas emociones que se reconocieron al menos en 0.5% de las interacciones. La Tabla 5 muestra los resultados obtenidos.

## Discusión

A partir de la Figura 1 se observa que las emociones que tienden a expresarse cuando se emplean medios de comunicación síncronos en situaciones de ACSC son: la simpatía, el agrado, el enojo, la confusión, el desagrado, la rebeldía y la frustración. A partir de los resultados obtenidos se puede observar que en el caso de los estudiantes universitarios de carreras de informática hay una tendencia a reflejar pocas emociones (25%) en las interacciones correspondientes a sesiones de ACSC. La escasez de la manifestación de emociones en interacciones basadas en texto está en concordancia con el hecho de que este tipo de canales no logra reflejar la información afectiva, que en las comunicaciones cara-a-cara se manifiestan a través de los gestos, el tono de voz, posturas, entre otros.

Respecto al modelo construido para reconocer emociones, los mejores resultados se obtuvieron con el de regresión múltiple, sin manejo de la negación y aplicando muestreo aleatorio. Sin embargo, si se considera un muestreo estratificado, el modelo SVM en el cual no se considera el manejo de la negación, presenta un mejor desempeño. Dado que la distribución de las emociones muestra que algunas emociones tienen más probabilidad de ocurrencia que otra, se debería optar por los resultados obtenidos con el muestreo estratificado puesto que este muestreo asegura que el entrenamiento haya sido realizado contemplando todas las clases.

Emoción	Situación Experimental	Precisión			Recall			F1		
		RLM	SVM	RN	RLM	SVM	RN	RLM	SVM	RN
Amor	I	0.00	0.33	0.00	0.00	0.17	0.00	0.00	0.22	0.00
	II	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	III	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Frustración	I	0.33	0.14	0.20	0.06	0.06	0.06	0.11	0.09	0.10
	II	0.38	0.23	0.00	0.13	0.13	0.00	0.19	0.17	0.00
	III	0.50	0.50	1.00	0.06	0.19	0.06	0.11	0.27	0.12
Rebeldía	I	0.50	0.43	0.36	0.35	0.35	0.29	0.49	0.39	0.31
	II	0.50	0.30	0.60	0.16	0.24	0.12	0.24	0.27	0.20
	III	0.50	0.40	0.44	0.29	0.24	0.24	0.37	0.30	0.31
Desagrado	I	0.50	0.45	0.45	0.23	0.23	0.23	0.31	0.30	0.30
	II	0.45	0.33	0.38	0.19	0.19	0.12	0.27	0.24	0.18
	III	0.40	0.50	0.40	0.09	0.14	0.09	0.15	0.21	0.15
Confusión	I	0.55	0.45	0.45	0.55	0.67	0.55	0.55	0.54	0.45
	II	0.39	0.34	0.44	0.20	0.24	0.15	0.26	0.28	0.23
	III	0.42	0.45	0.31	0.24	0.30	0.15	0.31	0.36	0.20
Enojo	I	0.50	0.45	0.45	0.37	0.35	0.40	0.43	0.39	0.42
	II	0.36	0.33	0.36	0.37	0.29	0.39	0.36	0.31	0.37

	III	0.42	0.43	0.42	0.23	0.23	0.23	0.30	0.30	0.30
Agrado	I	0.66	0.71	0.66	0.81	0.83	0.80	0.73	0.77	0.72
	II	0.60	0.63	0.57	0.68	0.69	0.69	0.63	0.66	0.62
	III	0.54	0.59	0.53	0.66	0.57	0.68	0.59	0.58	0.60
Simpatía	I	0.70	0.73	0.71	0.81	0.77	0.82	0.75	0.75	0.76
	II	0.60	0.61	0.59	0.80	0.78	0.81	0.68	0.68	0.68
	III	0.57	0.53	0.57	0.75	0.79	0.76	0.65	0.63	0.65

Tabla 5. Resultados obtenidos por la experimentación. Situaciones experimentadas: I. Sin manejo de la negación y aplicando muestreo estratificado; II. Con manejo de la negación y aplicando muestreo aleatorio; y III. Con manejo de la negación y aplicando muestreo estratificado

Se puede apreciar que para las emociones en las que se tiene un poco más de datos, el modelo aprendió a reconocer la emoción de una manera aceptable. Por ejemplo, para el modelo SVM (Tabla 5) la precisión para reconocer la emoción de simpatía es del 70% y es capaz de reconocer el 73% de todas las interacciones en las que se manifiesta esta emoción. Se debe recordar que la emoción de simpatía tuvo una frecuencia de 7.53% en los datos recolectados.

Entre las emociones que tienden a expresarse se observó que la frustración y la confusión están presentes. Sobre estas emociones el docente tiene que prestar atención e intentar sacar a los alumnos de dicho estado emocional. Por un lado, se asocia a la frustración con las tasas de abandono en la cursada de asignaturas (Capdeferro & Romero, 2012). Sin embargo, en (Baker et al., 2010) se sugiere diferenciar si la fuente de la frustración se atribuye a un evento externo o bien es parte natural de una actividad de procesamiento cognitivo, en este último caso puede ser no necesario la intervención externa. Por otro lado, la confusión puede estar correlacionada positivamente con el aprendizaje si se la administra productivamente, así, por ejemplo, algunos alumnos cuando están confundidos les atribuyen a sus bajas habilidades por lo cual requieren ser alentados, y otros cuando están confundidos se sienten motivados porque consideran a la situación como un desafío que deben superar (Baker et al., 2010). La forma en que regule el docente la frustración del alumno dependerá de las características de este último.

Las emociones influyen en el alumno en sus procesos motivacionales, estrategias de aprendizaje, recursos cognitivos, en la autorregulación y en el logro académico (Lan et al., 2023; Pekrun et al., 2002; Tahirbegi, 2023), de ahí la importancia de que el docente pueda tener acceso a la información emocional y reconocer aquellas emociones que afecten el trabajo de los alumnos, para sugerir cambios o intervenir apropiadamente en las situaciones necesarias. El docente, conociendo la información emocional de los grupos colaborativos que monitorea, puede: buscar dar autonomía al estudiante procurando que pueda autorregular su aprendizaje, permitir reconocer el valor de los temas abordados en clase y en las tareas colaborativas, proporcionar retroalimentación oportuna, y promover interacciones que afiancen los procesos colaborativos. Interviniendo el docente adecuadamente en estas dimensiones, creemos que sería posible mejorar las tasas de retención y

de aprobación de estudiantes en ciencias de la computación. Como futura línea de trabajo se buscará verificar la afirmación anterior.

### Conclusiones

Las emociones son un tipo de información que los docentes deben tener en cuenta para realizar ajustes en sus procesos de enseñanza a fin de aumentar las posibilidades de aprendizaje en los alumnos. En este trabajo se desarrolló y evaluó un modelo que permite reconocer emociones en situaciones de ACSC síncronas que emplean herramientas de comunicación textuales. Este modelo se basa en considerar la forma en que se expresan los alumnos cuando interactúan, el tipo de actividad colaborativa que representa a la interacción, y la consideración de indicadores tales como la presencia de dificultades de comunicación, dificultades con la tecnología, y el uso de expresiones que hagan inferir la existencia de incongruencias en objetivos.

El modelo construido demostró buenas capacidades para reconocer emociones y podrá reconocer una mayor cantidad de ellas si se dispone de datos suficientes para entrenarlo. La incorporación de este modelo en entornos que soportan el ACSC resulta ser una innovación educativa que permitiría realizar aportes sustantivos a la enseñanza en grupos colaborativos al posibilitar a los actores involucrados ser conscientes de las emociones. Como línea de trabajo futuro se plantea elaborar propuestas que, utilizando el modelo desarrollado, permitan usar los datos de información afectiva para contribuir tanto al proceso de enseñanza como al de aprendizaje. También se reconoce que sería interesante extender este estudio a experiencias de aprendizaje colaborativo presencial en la que se puedan tomar como entrada impresiones de los propios docentes y estudiantes, puesto que en general, la presencialidad es más rica en manifestaciones emocionales que los entornos mediados por computadora.

### Referencias

- Argueda, M., Xhafa, F., Daradoumis, T., & Caballe, S. (2015). An ontology about emotion awareness and affective feedback in elearning. *International Conference on Intelligent Networking and Collaborative Systems*, 156–163.
- Baker, R. S. J. d., D’Mello, S. K., Rodrigo, M. M. T., & Graesser, A. C. (2010). Better to be frustrated than bored: The incidence, persistence, and impact of learners’ cognitive-affective states during interactions with three different computer-based learning environments. *International Journal of Human Computer Studies*, 68(4), 223–241.
- Bennedsen, J., & Caspersen, M. E. (2008). Optimists have more fun, but do they learn better? On the influence of emotional and social factors on learning introductory computer science. *Computer Science Education*, 18(1), 1–16.
- Capdeferro, N., & Romero, M. (2012). Are Online Learners Frustrated with Collaborative Learning Experiences? *The International Review of Research in Open and Distance Learning*, 13(2), 26–44.
- Daradoumis, T., Argueda, M., & Xhafa, F. (2013). Building Intelligent Emotion Awareness for Improving Collaborative e-Learning. *5th International Conference on Intelligent*

- Networking and Collaborative Systems*, 281–288.
- Ekman, P., & Friesen, W. V. (1978). *Facial action coding system: A technique for the measurement of facial movement*. Consulting Psychologists Press.
- Feidakis, M. (2016). Chapter 11 - A Review of Emotion-Aware Systems for E-Learning in Virtual Environments. In S. Caballé & R. Clarisó (Eds.), *Formative Assessment Learning Data Analytics and Gamification* (Intelligen). Academic Press.
- Garcia-prieto, P., Bellard, E., & Schneider, S. C. (2003). Experiencing Diversity, Conflict, and Emotions in Teams. *Applied Psychology: An International Review*, 52(3), 413–440.
- Gunawardena, C. N. (1995). Social Presence Theory and Implications for Interaction and Collaborative Learning in Computer Conferences. *International Journal of Educational Telecommunications*, 1(2/3), 147–166.
- Heerdink, M. W., Kleef, G. A. Van, Homan, A. C., & Fischer, A. H. (2013). On the Social Influence of Emotions in Groups: Interpersonal Effects of Anger and Happiness on Conformity Versus Deviance. *Journal of Personality and Social Psychology*, 105(2), 262–284.
- Janssen, J., Erkens, G., Kirschner, P. A., & Kanselaar, G. (2012). Task-related and social regulation during online collaborative learning. *Metacognition and Learning*, 7(1), 25–43.
- Järvenoja, H., & Järvelä, S. (2013). Regulating Emotions Together for Motivated Collaboration. In *Affective Learning Together. Social and Emotional Dimensions of Collaborative Learning*. (pp. 162–181). Routledge.
- Jones, A., & Issroff, K. (2005). Learning technologies: Affective and social issues in computer-supported collaborative learning. *Computers and Education*, 44(4), 395–408.
- Kreijns, K., Kirschner, P. A., & Jochems, W. (2003). Identifying the pitfalls for social interaction in computer-supported collaborative learning environments: a review of the research. *Computers in Human Behavior*, 19(3), 335–353.
- Krippendorff, K. (2004). *Content Analysis: An Introduction to Its Methodology*. SAGE Publications.
- Lan, G., Zhao, X., & Gong, M. (2023). Motivational intensity and willingness to communicate in L2 learning: A moderated mediation model of enjoyment, boredom, and shyness. *System*, 117.
- Lescano, G., & Costaguta, R. (2018). COLLAB: Conflicts and Sentiments in chats. *XIX International Conference on Human Computer Interaction (Interacción 2018)*.
- Levin, D. Z., Kurtzberg, T. R., Phillips, K. W., & Lount, R. B. (2010). The role of affect in knowledge transfer. *Group Dynamics: Theory, Research, and Practice*, 14(2), 123–142.
- Miller, M., & Hadwin, A. (2015). Scripting and awareness tools for regulating collaborative learning: Changing the landscape of support in CSCL. *Computers in Human Behavior*, 52, 573–588.
- Nam, C. W. (2014). The effects of trust and constructive controversy on student achievement and attitude in online cooperative learning environments. *Computers in Human Behavior*, 37(August 2014), 237–248.

- Pekrun, R. (2014). *Emotions and Learning. Practices Series-24*. International Academy of Education & International Bureau of Education.
- Pekrun, R., Goetz, T., Titz, W., & Perry, R. (2002). Academic emotions in students' self-regulated learning and achievement: a program of qualitative and quantitative research. *Educational Psychologist, 37*(2), 91–105.
- Pekrun, R., & Linnenbrink-Garcia, L. (2012). Academic emotions and student engagement. In S. L. Christenson, A. L. Reschly, & C. Wylie (Eds.), *Handbook of research on student engagement* (pp. 259–282). Springer-Verlag.
- Pintrich, P. R., Smith, D. A. F., Garcia, T., & McKeachie, W. J. (1991). *A manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ)*.
- Robinson, K. (2013). The interrelationship of emotion and cognition when students undertake collaborative group work online: An interdisciplinary approach. *Computers and Education, 62*(March 2013), 298–307.
- Soller, A. (2001). Supporting Social Interaction in an Intelligent Collaborative Learning System. *International Journal of Artificial Intelligence in Education, 12*(1), 40–62.
- Tahirbegi, D. (2023). Exploring emotion regulation in small ensemble contexts; three cases from higher music education. *Learning, Culture and Social Interaction, 42*.
- Turner, J. C., Meyer, D. K., & Schweinle, A. (2003). The importance of emotion in theories of motivation: Empirical, methodological, and theoretical considerations from a goal theory perspective. *International Journal of Educational Research, 39*(4–5), 375–393.
- Wosnitza, M., & Volet, S. (2005). Origin, direction and impact of emotions in social online learning. *Learning and Instruction, 15*(5), 449–464.

## Relevamiento de conocimientos previos de programación en el nivel universitario

Gonzalo Pablo Fernández<sup>1</sup>, Cecilia Martínez<sup>2</sup>, Pablo E. "Fidel" Martínez López<sup>1,3,4</sup>  
gpfernandez@dc.uba.ar, cecimart@gmail.com, fidel.ml@gmail.com

<sup>1</sup> Universidad Nacional de Quilmes

<sup>2</sup> Universidad Nacional de Córdoba

<sup>3</sup> Centro de Investigación y Desarrollo en Informática Aplicada -  
Universidad Nacional de Hurlingham

<sup>4</sup> Fundación Sadosky

### Resumen

En los últimos años se le ha dado un papel preponderante a la enseñanza de la programación en los niveles obligatorios de la educación. Han surgido también numerosas alternativas para aprender a programar de forma autodidacta. Como resultado de estos procesos, el público que ingresa a la universidad puede tener conocimientos previos de programación heterogéneos. Las carreras universitarias de informática suelen asumir conocimientos nulos sobre programación pero eso está cada vez más alejado de la realidad.

En este trabajo nos proponemos relevar la población de estudiantes universitarios que cursan la materia *Introducción a la Programación* en la *Universidad Nacional de Quilmes*, materia que fue pensada originalmente para enseñar conceptos básicos de programación a estudiantes sin ningún conocimiento previo en el área. Desarrollamos un test diagnóstico para indagar experiencias previas en programación, así como para evaluar algunos conceptos respecto a construcciones sintácticas de los lenguajes de programación.

Nuestros hallazgos muestran la enorme heterogeneidad del público, teniendo estudiantes que aprendieron conceptos de programación en distintas etapas (en la escuela, con cursos en línea, de forma autodidacta), con distintos lenguajes de programación (tanto en texto como en bloques) y además, con percepciones muy distintas respecto a cuánto saben realmente.

**Palabras clave:** Enseñanza de la programación; Nivel universitario; Conocimientos previos

### Introducción

La incorporación de contenidos de computación, y sobre todo, programación en niveles cada vez más bajos de escolarización es un fenómeno global, motivado tanto por las nuevas investigaciones sobre Pensamiento Computacional (según las cuales, las habilidades que se aprenden al estudiar Computación son útiles para cualquier persona y no sólo para quienes se van a dedicar profesionalmente a la informática (Wing (2006), Denning and Tedre (2019))) como

por la creciente brecha entre la demanda de profesionales de la informática y la escasa oferta de mano de obra calificada para satisfacerla.

En Argentina en particular, la iniciativa Program.AR dedica grandes esfuerzos por llevar la enseñanza de la programación y otros temas relacionados a las Ciencias de la Computación a las escuelas primarias y secundarias, a través de capacitación docente, generación de contenido y otras actividades relacionadas. Por otro lado se desarrollaron numerosas propuestas de capacitación brindadas por el gobierno nacional en forma de cursos de programación, orientados principalmente a generar mano de obra calificada en poco tiempo. Entre estas propuestas se destaca Argentina Programa por su alcance nacional.

Enseñar a programar a estudiantes universitarios requiere abordar simultáneamente un conjunto de saberes complejos (Jenkins (2002)). Entre ellos, la sintaxis y la semántica de un lenguaje de programación son dos de los pilares sobre los que se construyen los demás conocimientos. La semántica se refiere a la expresión de las ideas que se intentan realizar a través del lenguaje de programación. La sintaxis incluye las reglas de construcción de las cadenas de símbolos que la computadora procesará para ejecutar la tarea encomendada. Ya en 1976, Ben Shneiderman proponía usar un enfoque espiralado para enseñar a programar distinguiendo claramente el aspecto sintáctico del semántico (Shneiderman (1977)).

En efecto, diversos estudios muestran que el alumnado que aprende programación presenta dificultades en comprender y construir las reglas sintácticas (Dijkstra (1989), Winslow (1996)). Xie et al. proponen que las habilidades necesarias para aprender a programar son 4 aunque dos de ellas están relacionadas con la sintaxis (la capacidad para leerla y entenderla y la capacidad para escribirla) y las otras dos están relacionadas con la semántica (la capacidad para identificar construcciones semánticas y la capacidad para usarlas correctamente según la situación) (Xie et al. (2019)).

Respecto al problema de la sintaxis, se ha realizado mucha investigación en torno a cómo superar las barreras iniciales que presentan la sintaxis compleja de los lenguajes de programación, sobre todo en los niveles educativos iniciales. El principal avance gira en torno a los entornos de programación por bloques que, por la forma en que se escriben los programas, impide cometer errores sintácticos. Sin embargo, poco se ha investigado sobre tales cuestiones en el nivel universitario (Abdul-Rahman and du Boulay (2014)). Ahora que es cada vez más usual que las personas aprendan a programar en los primeros niveles educativos (o incluso, que aprendan a través de cursos o de forma autodidacta) se vuelve imprescindible poner el foco en los primeros años de la educación universitaria.

En nuestra práctica como docentes de *Introducción a la Programación* hemos observado a través

de los años que los estudiantes tienen dificultades para construir las nociones asociadas a las reglas sintácticas del lenguaje, sobre todo en las referentes a la distinción entre comandos y expresiones. Nos hemos propuesto establecer cuáles son los saberes previos de sintaxis de programación de los estudiantes de *Introducción a la Programación* en el primer año de una carrera universitaria.

En la sección 2 introducimos el marco teórico sobre el que trabajaremos, tanto en cuanto a antecedentes como en cuanto a definiciones. En la sección 3 describimos el estudio realizado, comenzando con la población sobre la que se realizó, siguiendo con el detalle del instrumento administrado y finalizando con el procesamiento realizado sobre los datos recolectados. En la sección 4 analizamos los resultados obtenidos en las distintas preguntas. Finalmente, resumimos nuestras conclusiones en la sección 5.

## **Marco Teórico**

### **Antecedentes**

Gómez et al. realizan un estudio en el que comparan a dos grupos de estudiantes de primaria. Uno de los grupos había tenido experiencia previa programando en un entorno de programación por bloques y el otro no. Aunque la cantidad de errores sintácticos en ambos grupos fue similar, el grupo que había tenido experiencia previa con bloques tuvo mejores notas en los exámenes, lo que sugiere que la experiencia con bloques les ayudó a desarrollar ciertas habilidades aunque no les facilitó la comprensión de la sintaxis (Gomez et al. (2019)).

Joanna Goode realiza una investigación para indagar sobre los saberes tecnológicos de los ingresantes a la Universidad de los Ángeles en función de datos demográficos. Los datos recogidos en encuestas y entrevistas muestran que a mayor acceso y a mejor posición social, mayores son los saberes de tecnología – entre ellos conceptos centrales de programación. Particularmente describe que aquellos estudiantes que pudieron cursar materias de programación en el secundario han podido borrar las diferencias de origen socio-económico y construir saberes digitales de computación (Goode (2010)).

### **Definiciones Teóricas**

Como ya se mencionó en la Introducción, la sintaxis consiste en el conjunto de reglas que establecen la correcta estructura de los símbolos y construcciones que conforman un programa. Cuando hablamos de estructura sintáctica nos referimos al Árbol de Sintaxis Abstracta (AST), que es una representación del código en forma de árbol donde cada nodo es una de las posibles construcciones y sus hijos son las partes de dicha construcción, sin importar qué símbolos las describen en el texto que conforma el programa. Si bien el AST no es parte explícita de los

contenidos impartidos por la materia, es una noción que es posible construir a partir de estudiar la estructura sintáctica de cada herramienta del lenguaje y se espera que luego de trabajar con programas los estudiantes perciban esta estructura de forma al menos implícita y que los ayude a comprender mejor la estructura de las ideas que están expresando mediante el programa.

### **Metodología**

Para responder a nuestra pregunta de investigación sobre cuáles son las nociones previas de AST de los estudiantes de primer año de programación se realizó un test que permite identificar y cuantificar estas nociones. El experimento se llevó a cabo durante la cursada del primer semestre de 2023.

### **Población**

Nuestros estudiantes asisten a la *Universidad Nacional de Quilmes* (UNQ) . Se trata de una universidad pública ubicada a 15 km de la ciudad de Buenos Aires . Hay 3 carreras en la Universidad que tienen como materia obligatoria a *Introducción a la Programación* (InPr): Tecnicatura en Programación Informática (TPI), Licenciatura en Informática (LI) y Licenciatura en Bioinformática (LB). La TPI está casi por completo contenida dentro de la LI y por eso es común que quienes se anotan a la LI se anoten también a la TPI. El 50 % se ha inscripto sólo en la TPI, el 34 % solo en la LI, y el 14 % en ambas. Un estudiante es de la LB.

La materia en la que se realiza el estudio (InPr) es del segundo semestre. Originalmente fue pensada para introducir las bases de la programación y que sea el primer contacto con elementos de programación, utilizando el lenguaje *Gobstones* (Martínez López et al. (2012)), un lenguaje diseñado específicamente para enseñar a programar. InPr aborda conceptos básicos de programación tales como programas, procedimientos, repetición simple y alternativa condicional, además de sumar parámetros, repetición condicional, funciones, variables y tipos de datos simples (registros y listas), e incorpora nociones de documentación y estilo. Tras un cambio de plan se incorporó una materia previa, *Elementos de Programación y Lógica* (EPyL), como correlativa, pero sin articulación entre los grupos docentes. En esta materia decidieron incluir nuevamente, entre otros temas, los conceptos básicos de programación: programas simples, procedimientos, repetición simple y alternativa condicional, a pesar de que los mismos ya se abordaban en la materia siguiente. Se utiliza el lenguaje QDraw, un lenguaje teórico que fue construido específicamente para esta materia como simplificación de las ideas de Gobstones. Debido a que no hay un número suficiente de computadoras para que los estudiantes puedan realizar sus prácticas, nunca se pensó en realizar una implementación en computadoras de un intérprete o compilador de QDraw que permita utilizarlo en máquina. Por esas razones, los ejercicios se resuelven en papel y no hay forma automática de validar la corrección de un

programa.

La materia InPr se cursa en 7 comisiones de entre 30 y 40 personas cada una. Una de las comisiones es virtual y se dejó fuera del estudio porque el objetivo de esta investigación era analizar los procesos de aprendizaje y se pensó dificultoso de observar estos procesos de manera virtual. Del total de 219 estudiantes que se inscribieron a la materia, participaron del estudio y dieron su consentimiento 157 personas.

### **Recolección de la información**

Para recuperar información sobre las nociones previas de programación se elaboró un test de saberes previos que incluyó una sección inicial de datos demográficos y una segunda parte con las preguntas temáticas. Así el test consistía en dos grandes momentos.

Una primera serie de preguntas indagaban sobre las condiciones socio-educativas previas del estudiantado (colegio secundario al que asistieron, experiencias previas de programación, asistencia a cursos de programación, etc).

El segundo momento consistía en una serie de actividades que para su resolución requerían poner en juego los conceptos que estamos analizando en este estudio a saber: correcta interpretación de una estructura jerárquica, percepción de estructura sobre determinados conceptos cotidianos, correcta interpretación de la estructura sintáctica de un cuerpo de texto (ya sea código o lenguaje coloquial y la relación entre ambos), clasificación sintáctica de porciones de código en distintos contextos y relación de los mismos con su semántica correspondiente, entre otros.

El instrumento fue administrado el primer día de clase. El test ofreció 7 actividades, de las cuales seleccionamos 2 de ellas por ser las más relevantes. Las actividades elegidas se detallan a continuación.

**Huevo y Gallina.** Esta actividad intenta recuperar la noción de indentación que indica la ubicación de acciones de programación en relación a otras. Se presentan varias imágenes que, se supone, representan índices de un libro. Se les pregunta en cuáles la sección “La gallina” es subsección de la sección “El huevo”. Entre las distintas opciones varían el orden de las secciones, la indentación relativa entre las secciones y la numeración de las mismas. Esperamos que quienes no saben programar elijan más aquellas opciones que priorizan la numeración de las secciones por sobre la indentación de las mismas. En la figura 1 se muestran las posibles opciones entre las cuales elegir.

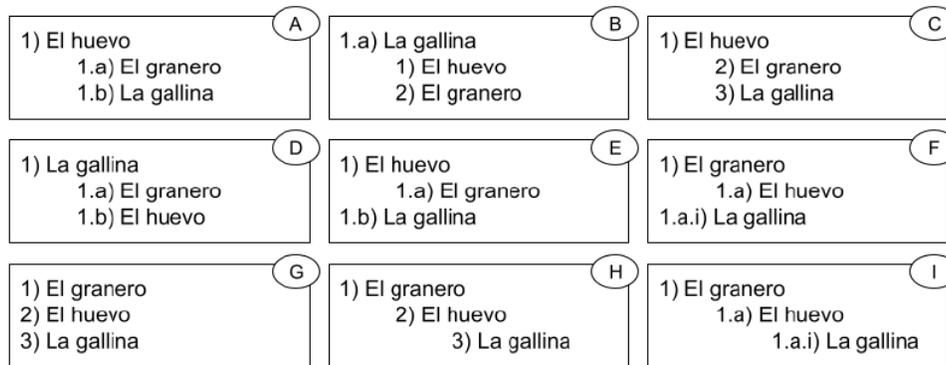


Figura 1. Propuestas de posibles índices presentadas para la pregunta **Huevo y Gallina**.

**Cantidad o Acción.** El objetivo de esta actividad es recuperar nociones sobre sintaxis y sobre semántica inferidas a partir de fragmentos de código. Se presentan varios fragmentos de código y en cada uno se les pregunta si cierta palabra dentro del código corresponde a una cantidad, a una acción o a otra cosa (en cuyo caso se esperaba que digan a qué otra cosa) y que justifiquen su respuesta. Hay varios indicios (algunos sintácticos y otros semánticos) en cada fragmento que indican una u otra opción y van variando entre los distintos fragmentos. Estos indicios son:

- la categoría sintáctica del nombre de la subtarea (que puede ser un verbo o un sustantivo),
- el significado del nombre de la subtarea (es decir, la semántica del nombre),
- su ubicación dentro del código (que puede ser la ubicación de un argumento o de un cuerpo) y
- el hecho de que inicie o no con una letra mayúscula.

En la figura 2 se muestran cada uno de los fragmentos de código que se pedía clasificar.

Esperamos que quienes no saben programar usen los dos primeros indicadores para determinar si la subtarea en cuestión es una cantidad (es decir, una expresión) o una acción (es decir, un comando), ya que están relacionados al nombre elegido. El uso de mayúsculas no debería influir demasiado porque es un indicador utilizado solamente en la materia: los comandos en Gobstones empiezan con mayúsculas y las expresiones con minúsculas. En cuanto a la ubicación, quienes ya saben programar deberían reconocerlo, sobre todo si conocen lenguajes como C o Java en los que se utiliza la misma sintaxis de cuerpos encerrados entre llaves. Sin embargo, también hay que considerar que, siendo que la sintaxis es casi idéntica a la de QDraw (el lenguaje que ya usaron en la materia correlativa), tiene sentido que presten atención a la estructura de código y sepan que el nombre es solamente un texto sin significado que no tiene ningún efecto en la ejecución. En este experimento se descartaron varios resultados ya que las

explicaciones dadas mostraron que no comprendieron la consigna, por lo que el total baja a 141 participantes.

<pre>repetir (3) {   comerUnaManzana() }</pre>	<pre>repetir (<u>CantidadDeManzanas</u>()) {   ComerUnaManzana() }</pre>
<pre>repetir (<u>comerUnaManzana</u>()) {   cantidadDeManzanas () }</pre>	<pre>repetir (ComerUnaManzana()) {   <u>CantidadDeManzanas</u> () }</pre>
<pre>repetir (3) {   <u>Manzana</u> () }</pre>	<pre>repetir (<u>contarCantidadDeManzanas</u>()) {   ComerUnaManzana () }</pre>
<pre>repetir (ContarCantidadDeManzanas()) {   <u>manzana</u> () }</pre>	<pre>repetir (<u>ContarCantidadDeManzanas</u>()) {   ComerUnaManzana () }</pre>
<pre>repetir (<u>Manzana</u>()) {   cantidadDeManzanas () }</pre>	<pre>repetir (ComerUnaManzana()) {   <u>contarCantidadDeManzanas</u> () }</pre>
<pre>repetir (<u>ComerUnaManzana</u>()) {   cantidadDeManzanas () }</pre>	<pre>repetir (ComerUnaManzana()) {   <u>cantidadDeManzanas</u> () }</pre>

Figura 2. Fragmentos de código que se pide clasificar en la pregunta **Cantidad o Acción**.

### Estrategias de análisis

Una primera aproximación a los datos fue a través de estadística descriptiva. Las respuestas del estudiantado se cruzaron en una planilla de cálculo buscando frecuencias y correlaciones. Se identificaron ítems relativos a experiencias previas de aprendizaje de programación con (cursos, materias del secundario, lenguajes, etc).

**Huevo y Gallina.** A cada una de las posibles opciones se les asignó un puntaje positivo o negativo para cada uno de los posibles aspectos (*Numeración e Indentación*) que pudieron haber tenido en cuenta al momento de elegir aquellas opciones que cumplían con la consigna. Así, las opciones que en base a la numeración respetaban la consigna, tuvieron un puntaje positivo para el aspecto *Numeración* mientras que aquellas que en base a la numeración no respetaban la consigna tuvieron un puntaje negativo en tal aspecto. A partir de estos puntajes se calculó el porcentaje de preferencia para cada aspecto como la proporción de opciones elegidas de entre aquellas tales que, según el aspecto, respetaban la consigna.

Si bien la indentación no es una característica exigida por la computadora para que el programa funcione, poner en juego procesos de indentación sería un indicador de que les estudiantes han comprendido la estructura sintáctica del código, de manera que puedan ordenarlo para hacerlo accesible, en primer lugar a ellos mismos en un momento posterior y en segundo lugar a otros programadores.

**Cantidad o Acción.** Se intentó inferir en base a la respuesta qué indicador usaron para decidir. Se tuvieron en cuenta primero los 4 indicadores antes mencionados. El indicador *Posición*

establece que si la palabra está entre llaves debe ser un comando y si está entre paréntesis debe ser una expresión. El indicador *Mayúsculas* establece que si la palabra empieza con mayúsculas debe ser un comando y si no una expresión. El indicador *Categoría* establece que si la primera palabra es un verbo debe nombrar un comando y si es un sustantivo debe nombrar una expresión. Para estos primeros 3 indicadores podríamos asociar la elección “Acción” con ser un comando y la elección “Cantidad” con ser una expresión. Sin embargo, según el indicador *Categoría*, las variantes de “Manzana” deberían ser consideradas como “Cantidad” por ser sustantivos. El cuarto indicador es *Semántica* que hace referencia al significado de la oración. Las variantes de “contarCantidadDeManzanas” deberían ser interpretadas como cantidades según este indicador ya que, si bien parece estar describiendo una acción, lo que en definitiva resolvería algo con ese nombre es calcula una cantidad. También hay un detalle con las variantes de “Manzana” ya que teniendo en cuenta el indicador semántico, la respuesta no debería ser ninguna de las dos.

El solapamiento de los indicadores *Categoría* y *Semántica* en las variantes de “Manzana” y “contarCantidadDeManzanas”, nos llevaron a definir un segundo análisis en el que mantuvimos *Posición* y *Mayúsculas* como indicadores y, en lugar de los últimos dos, un tercer indicador basado en el nombre (*Nombre*) que busca incluir ambos aspectos sin ambigüedades. Según este indicador, las variantes de “ComerManzana” serían acciones, las variantes de “cantidadDeManzanas” serían cantidades y las variantes de “Manzana” no serían ninguna de las dos. En el segundo análisis se descartaron las preguntas respecto a las variantes de “contarCantidadDeManzanas”.

Para las opciones en las que la respuesta fue “Otro” se determinó qué indicadores se observaron en base a las justificaciones dadas. Con esta información se calculó cuál fue el indicador preferido por cada participante en base a cuál fue tenido en cuenta en más respuestas que los otros. En caso de empate, consideramos que le participante no tiene preferencia por ninguno.

Finalmente se llevó a cabo un análisis de preferencia definitiva en el que se distinguió a aquellas personas que parecían estar usando siempre el mismo indicador para basar su elección. Decimos que una persona tiene preferencia definitiva por cierto indicador si todas sus respuestas son coherentes con el criterio de tal indicador.

## **Resultados**

A continuación analizamos los resultados obtenidos. Primero observamos las respuestas dadas por los participantes en el primer momento del estudio, donde se les preguntó por su contexto socio-económico así como sobre sus conocimientos previos en programación. Luego estudiamos las respuestas para las dos actividades del segundo momento, a partir de las cuales podemos

inferir sus intuiciones respecto a la indentación del código y a la correcta identificación de elementos en un fragmento de código, respectivamente.

### **Contexto previo**

Un primer hallazgo es que el 50 % del estudiantado cursó materias de programación en el nivel secundario. Respecto de la formación previa de nuestros estudiantes en programación se pudo reconstruir a partir de la encuesta que el 20 % fue a un colegio técnico o con orientación en informática. El 64 % no fue a colegio técnico o con orientación en informática pero igual tuvo alguna materia relacionada (informática, computación, tics, etc.) en su escuela secundaria.

En Argentina, las escuelas técnicas y con orientación en informática ofrecen como materias obligatorias tres materias de Programación, Aplicación a las Nuevas Tecnologías (que incluye base de datos, normas de seguridad, etc.), y Formación en Ambientes del Trabajo. En tanto en la formación general básica informática incluye generalmente habilidades que permiten usar programas, o tecnologías de la información y la comunicación. Entre ellas se incluyen el uso de procesadores de textos o planillas de cálculos. Sin embargo, al preguntarles si habían aprendido algún lenguaje de programación (ya sea de texto o de bloques) en la escuela, menos del 15 % respondió de forma afirmativa. Sí observamos una mayor cantidad de gente que aprendió a programar a través de cursos o de forma autodidacta. Un 10 % del estudiantado encuestado mencionó haber realizado (o estar realizando en el momento) el curso Argentina Programa.

A pesar de haber aprobado EPyL, más de la mitad (55 %) aseguró no saber programar. El 30 % aseguró que sí sabe aunque al preguntarles qué lenguajes conocían mencionaron lenguajes que no son de programación (como html o css). Otro 10 % dijo saber pero destacó que sabe muy poco o sólo lo básico. Sólo 4 mencionaron QDraw.

Podemos observar que los antecedentes formales académicos no son un indicador válido de los conocimientos previos, pues a pesar de haber tenido materias de programación en la secundaria y de haber cursado una materia previa con contenidos de programación, los estudiantes manifiestan no saber esos conocimientos.

Asimismo, otra observación es que pareciera que los estudiantes no identifican las nociones previas a poder programar como parte de saberes de programación. Ellos mencionan que “no saben programar” a pesar de haber logrado en las experiencias previas construcciones sencillas donde se usan los principales conceptos. Dos hipótesis nos permitirían explicar este fenómeno. Por un lado la confianza – o falta de confianza – en sus posibilidades de programar. Esto se evidencia en la baja proporción de estudiantes que aseguraron saber programar (con un rotundo “no” como respuesta) o que dieron respuestas tímidas como “sí, pero poco” o “recién estoy

empezando”.

Por otro lado, la idea de que programar construcciones sencillas en Scratch u otros entornos didácticos “no es programar”. Esto se evidencia por un lado en el hecho de que sólo 4 personas hayan mencionado el lenguaje QDraw en la respuesta a los lenguajes de programación conocidos y por otro en la aparente nula relación entre haber usado entornos de programación por bloques y saber programar. De las 76 personas que mencionaron haber utilizado algún entorno de programación por bloques, 36 aseguraron no saber programar (casi la mitad) y sólo 20 (27 %) aseguraron que sí, aunque todos conocían además algún lenguaje textual industrial.

Esta cuestión fue observada por Colleen Lewis al comparar dos grupos de estudiantes aprendiendo a programar. Uno de los grupos aprendió con Logo y el otro con Scratch. Al indagar las percepciones de los estudiantes al finalizar el curso quienes habían aprendido a programar con Logo (un lenguaje basado en texto) aparentaban tener mayor confianza en sus habilidades que quienes aprendieron con Scratch (un lenguaje basado en bloques) (Lewis (2010)).

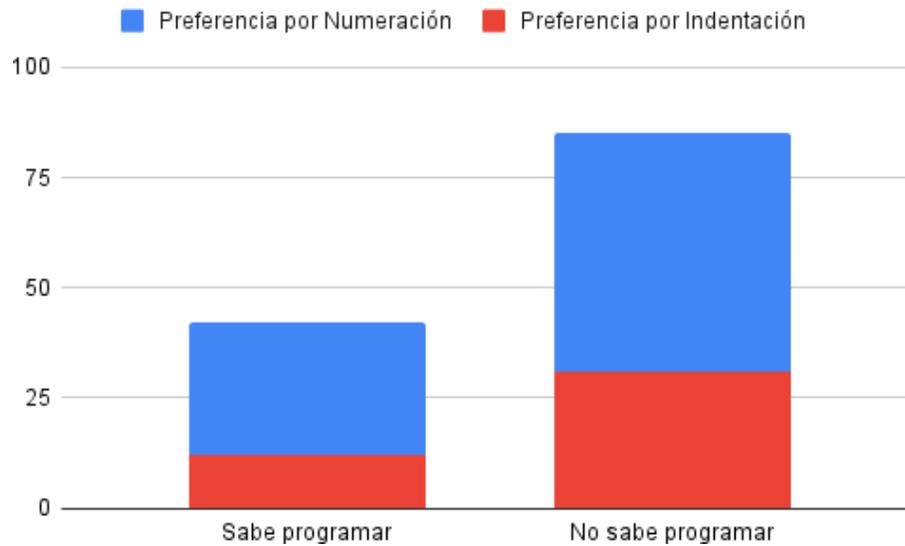
### **Indentación**

Respecto a la noción de indentación observamos que la mayor parte del estudiantado le dio mayor peso al aspecto semántico que al sintáctico cuando tuvieron que determinar cuál era la jerarquía implicada en la actividad del huevo y la gallina. Al considerar el porcentaje de estudiantes que eligió en base a la numeración o a la indentación, la relación fue 70 % a favor de la numeración contra 30 % de la indentación.

En una segunda revisión nos enfocamos en las opciones que favorecían Numeración pero contradecían Indentación (las opciones E y F) y las que hacían lo contrario (C y H). Comparamos entonces cuántas personas eligieron E y F pero no C y H y cuántas eligieron C y H pero no E y F. Observamos en este caso que más del 30 % eligió las dos de numeración y no las dos de indentación y sólo un 5 % hizo la elección inversa: eligió las dos de indentación y no las dos de numeración.

Estos datos nos permiten deducir que la noción de jerarquía sintáctica por indentación no parece estar interiorizada como un elemento a tener en cuenta para determinar la jerarquía de la estructura.

Al cruzar los datos de preferencia con las respuestas a la pregunta de si sabían programar vemos que no parece haber correlación. Como se puede observar en la figura 3, pareciera que la numeración sigue siendo el aspecto más elegido sin importar el conocimiento previo en programación.



**Figura 3.** Cantidad de personas que manifestaron preferencia por alguno de los indicadores *Numeración* o *Indentación*, agrupadas según si afirmaron o negaron saber programar.

### Clasificación

Respecto a la noción de clasificación de fragmentos, en ambos análisis se observó que el indicador más elegido fue *Posición*. En el primer análisis con 4 indicadores, el segundo indicador más elegido fue *Categoría*. En el segundo análisis con 3 indicadores, el indicador *Nombre* fue casi tan elegido como el indicador *Posición*. En ambos casos el indicador *Mayúsculas* fue el menos preferido. Los porcentajes de preferencia por cada indicador se pueden apreciar en las figuras 4 y 5 respectivamente.

La elección del indicador *Posición* por sobre el indicador *Nombre* nos hace pensar que estas personas conocen el funcionamiento de los lenguajes de programación en los que los nombres son ignorados por la computadora cuando el código se ejecuta mientras que las personas que eligieron *Nombre* por sobre *Posición* podrían no saber cómo es la estructura sintáctica de los lenguajes de programación y se guiaron por lo que sí entendían en el fragmento presentado. Sin embargo también podríamos pensar que este segundo grupo eligió en base al nombre porque esperan que el código use nombres declarativos. En el caso de la preferencia definitiva, observamos que 25 % usó siempre la posición para elegir, 10 % eligió siempre en base al nombre y menos de 1 % (una única persona) eligió siempre en base a las mayúsculas.

Esto nos muestra que la gran mayoría no tiene un criterio claro para decidir cuando algunos indicadores contradicen a otros. También vemos que la proporción que elige en base a la

posición ignorando totalmente los demás indicadores es bastante mayor a la que tiene esa preferencia con el nombre, a pesar de que en el análisis anterior estos dos grupos parecían ser de similar dimensión. Esperando encontrar alguna correlación entre conocimientos

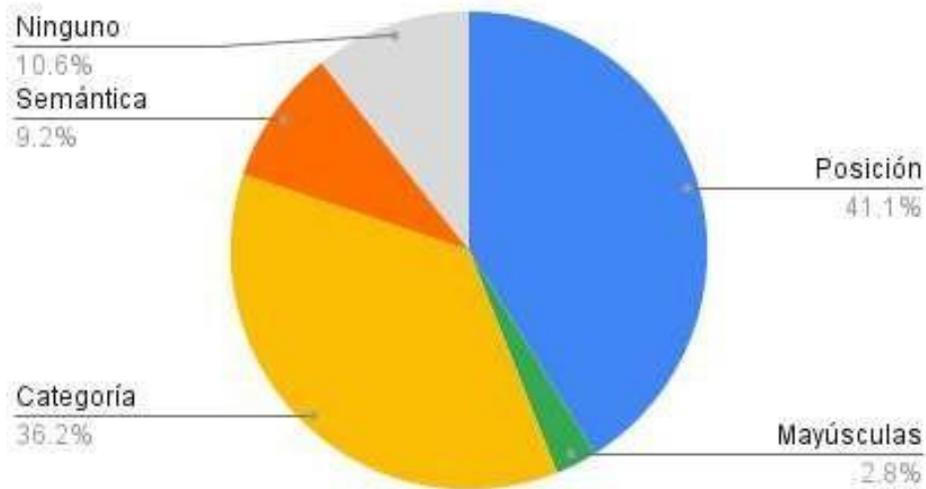


Figura 4. Porcentaje de participantes que manifestaron preferencia por cada uno de los cuatro indicadores en el primer análisis.

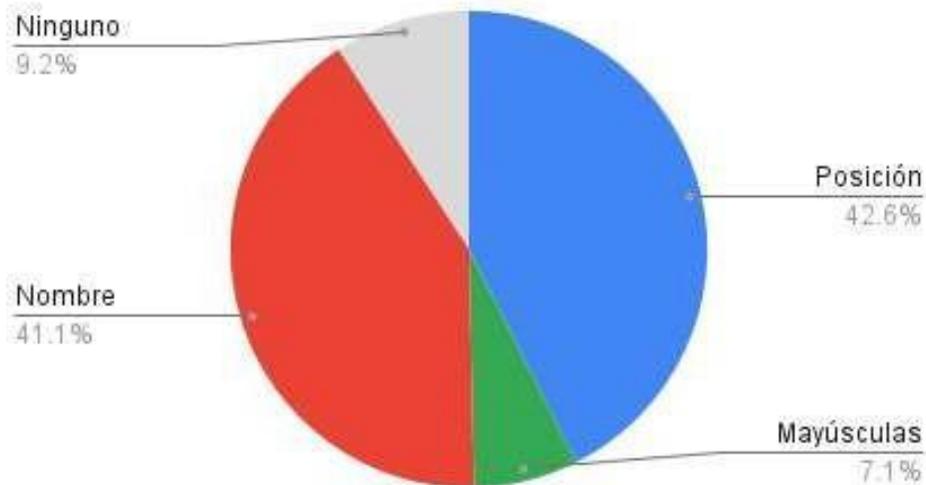


Figura 5. Porcentaje de participantes que manifestaron preferencia por cada uno de los tres indicadores en el segundo análisis.

previos en programación y la predilección por el indicador *Posición*, cruzamos estos datos con las respuestas a la pregunta de si sabían programar. Se tuvieron en cuenta para esto sólo las respuestas que fueron afirmativas o negativas (ignorando las dudosas). Este análisis se puede apreciar en la figura 6.

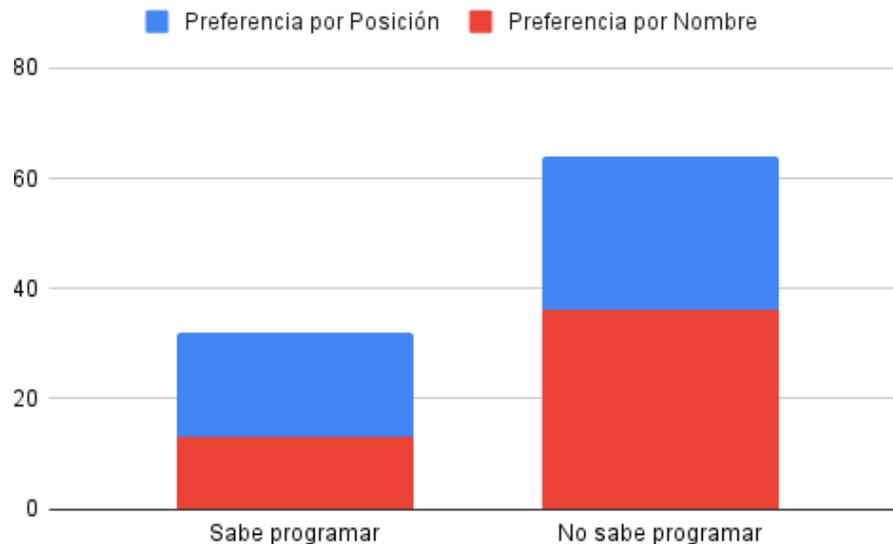


Figura 6. Cantidad de personas que manifestaron preferencia por alguno de los indicadores *Posición* o *Nombre*, agrupadas según si afirmaron o negaron saber programar.

Observamos que no parece haber correlación entre autopercepción de que saben programar desde antes y haber tenido más en cuenta la posición (es decir la sintaxis) que el nombre (es decir la semántica). Como el gráfico anterior, podría explicarse si suponemos que quienes saben programar esperan que el código use nombres declarativos y por eso hay una gran proporción que le da prioridad al nombre, aunque creemos que la explicación puede ser en realidad que la respuesta a la pregunta de si saben programar no haya sido del todo honesta. Volviendo a recordar que todos los participantes cursaron la materia EPyL, quienes negaron saber programar podrían haber interiorizado ciertos conceptos que ahora están poniendo en juego. En otras palabras, sí aprendieron a programar (o al menos, aprendieron algunos conceptos) pero no perciben los saberes adquiridos como habilidades relacionadas a la programación.

## Conclusiones

Se realizó un relevamiento de los conocimientos previos en programación a estudiantes que cursan la materia Introducción a la Programación en la Universidad Nacional de Quilmes. Aunque la materia fue pensada asumiendo nulos (o muy escasos) conocimientos previos en

programación, como docentes empezamos a notar cada vez más que no es esa la situación. El objetivo de este trabajo de investigación no es en este momento plantear propuestas para realizar modificaciones en el dictado de la materia sino comprender el contexto inicial para luego avanzar en las modificaciones que sean pertinentes.

Un primer hallazgo es que el grupo es muy heterogéneo en cuanto a conocimientos previos de programación. Teniendo en cuenta únicamente la respuesta a la pregunta “¿Sabés programar?” ya observamos una gran disparidad. Sin embargo, en un análisis más detallado de las respuestas, observamos que no es suficiente con separar al curso entre quienes saben y quienes no saben programar ya que encontramos una gran cantidad de puntos intermedios (gente que dice que no sabe programar a pesar de que utilizó algún entorno de programación por bloques, gente que dice que no sabe programar pero identifica correctamente los elementos de programación, etc).

Recordando que todos los participantes tuvieron que cursar la materia correlativa EPyL, observamos que los contenidos impartidos no alcanzan para que los estudiantes perciban que saben programar, aunque hay varios indicios que nos muestran que sí parecen haber adquirido varios saberes relacionados a la disciplina. Esto puede deberse a que no esté clara la definición de programar y, por lo tanto, no consideren que todo lo que aprendieron en EPyL sea programar.

En cuanto a los saberes previos en programación, saber programar (o creerlo) no parece ser un factor determinante al momento de realizar las actividades. Independientemente de si sabían o no programar, la numeración de las secciones fue el aspecto más elegido para decidir la jerarquía entre ellas. En el caso de la clasificación en comandos y expresiones, se ven similares proporciones de personas que deciden en base a la posición y que deciden en base al nombre. Incluso entre quienes afirmaron saber programar, hay una importante proporción que presta atención al nombre, a pesar de que este no sea determinante para la correcta ejecución del código.

Queda como trabajo futuro reunir al plantel docente de la materia para replantear ciertos aspectos de su dictado teniendo en cuenta los resultados obtenidos así como también articular mejor la transición desde la materia correlativa EPyL. Este problema se abordará en otra etapa del trabajo.

## **Bibliografía**

Abdul-Rahman, S.-S. and du Boulay, B. (2014). Learning programming via worked-examples: Relation of learning styles to cognitive load. *Computers in Human Behavior*, 30:286–298.

Denning, P. J. and Tedre, M. (2019). *Computational Thinking*. The MIT Press Essential

Knowledge Series. MIT Press.

- Dijkstra, E. W. (1989). On the cruelty of really teaching computing science. *Communications of the ACM*, 32(12):1398–1404.
- Gomez, M. J., Moresi, M., and Benotti, L. (2019). Text-based programming in elementary school: A comparative study of programming abilities in children with and without block-based experience. *Proceedings of the 2019 ACM Conference on Innovation and Techno-logy in Computer Science Education*, pages 402–408.
- Goode, J. (2010). Mind the gap: The digital dimension of college access. *The Journal of Higher Education*, 81(5):583–618.
- Jenkins, T. (2002). On the difficulty of learning to program. *3rd Annual LTSN-ICS Conference, Loughborough University*.
- Lewis, C. M. (2010). How programming environment shapes perception, learning and goals: Logo vs. scratch. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, pages 346–350.
- Martínez López, P. E., Bonelli, E. A., and Sawady O'Connor, F. A. (2012). El nombre verdadero de la programación. una concepción de enseñanza de la programación para la sociedad de la información. *Anales del 10º Simposio de la Sociedad de la Información (SSI'12), dentro de las 41 Jornadas Argentinas de Informática e Investigación Operativa (JAIO'12)*, pages 1–23.
- Shneiderman, B. (1977). Teaching programming: A spiral approach to syntax and semantics. *Computers and Education*, 1(4):193–197.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.
- Winslow, L. E. (1996). Programming pedagogy - a psychological overview. *SIGCSE Bulletin*, 28(3):17–22.
- Xie, B., Loksa, D., Nelson, G. L., Davidson, M. J., Dong, D., Kwik, H., Hui Tan, A., Hwa, L., Li, M., and Ko, A. J. (2019). A theory of instruction for introductory programming skills. *Computer Science Education*, 29(2):205–253.

## **La Accesibilidad Web: una perspectiva de formación profesional en las Ciencias de la Computación<sup>1</sup>**

Javier Díaz, Alejandra Schiavoni, Paola Amadeo, Ivana Harari, Ana M. Ungaro  
{jdiaz, ales, pamadeo, iharari, anaungaro}@linti.unlp.edu.ar

Laboratorio de Investigación en Nuevas Tecnologías Informáticas  
Facultad de Informática. Universidad Nacional de La Plata

### **Resumen**

Pensar el proceso de formación implica analizar y reflexionar desde una perspectiva socio-histórica situada a la disciplina y a los destinatarios. Desde esta mirada se hacen evidentes las áreas de vacancia y la necesidad de plantear innovaciones respecto de los avances de la sociedad del conocimiento y de los problemas complejos que la Informática tiene que abordar desde una universidad pública comprometida con la sociedad en virtud de trabajar por una educación como un Derecho. La Accesibilidad Web es una temática compleja que conjuga aspectos informáticos, legales, sociales, morales, educativos y éticos. Dentro del desarrollo informático la accesibilidad está ligada a todas las áreas, desde los contenidistas, diseñadores, comunicadores hasta los evaluadores de calidad de software. Es una temática transversal por lo que su abordaje debe ser integral. En este artículo se plasman las actividades de formación sobre Accesibilidad Web que se llevan a cabo en la Facultad de Informática y cuyos destinatarios pertenecen a diferentes grupos de actores de la sociedad. Se describen los objetivos y tareas desarrollados en cada caso, teniendo siempre como eje el reconocer la relevancia de trabajar la inclusión y accesibilidad.

**Palabras claves:** Accesibilidad Web, Inclusión, estrategias de formación, docentes y estudiantes.

### **Introducción**

¿Qué es educar en la Universidad? Este interrogante inicial abre la puerta para reflexionar acerca de la enseñanza en términos de inclusión y derechos. Una pregunta que parece obvia pero invita a reflexionar sobre los procesos en particular y el interés por pensar y repensar el sentido de las estrategias institucionales y pedagógicas que contribuyen a construir una universidad que reciba a las diversidades de sujetos que quieren estudiar y formarse en ella, como también a pensar en la dimensión epistemológica que significa incluir nuevos problemas disciplinares.

Interpelar lo obvio, lo que muchas veces no se cuestiona, intenta transparentar y reconocer los

---

<sup>1</sup> El presente artículo adhiere al uso de lenguaje inclusivo, respetando los derechos de las personas a ser nombradas según su elección.

procesos formativos que trascienden las fronteras de las propuestas formales de la educación para poder incluir nuevas dimensiones de trabajo, análisis, conocimiento y problemas que cuestionan y reconfiguran la enseñanza planteada en el currículum de las carreras de informática.

Esta mirada, que complejiza, nos ayuda a entender de manera espiralada cómo el espacio y proceso de formación de estudiantes y profesionales es un desafío permanente de actualización disciplinar. El proceso educativo y la formación profesional es una práctica continua y se va transformando en un contexto que la atraviesa, moldea y lo constituye, recorriendo diferentes espacios de formación que van desde el grado al posgrado, desde lo formal a instancias informales de capacitación abarcando la heterogeneidad de destinatarios.

Pensar el proceso de formación implica analizar y reflexionar desde una perspectiva socio-histórica y situada a la disciplina y a los destinatarios. Desde esta mirada se hacen evidentes las áreas de vacancia y la necesidad de plantear innovaciones respecto de los avances que la sociedad del conocimiento nos pone como desafíos, y nos invita a pensar en virtud de problemas complejos que la Informática como disciplina tiene que abordar desde una universidad pública comprometida con la sociedad en virtud de trabajar por una educación como un Derecho.

La Accesibilidad Web es una temática compleja que conjuga tanto aspectos puramente informáticos como aspectos legales, sociales, morales, educativos y éticos. El perfil de un profesional informático incluye incumbencias que abordan distintas áreas como la investigación, auditoría, y desarrollo de sistemas informáticos. Pero todos estos conocimientos no cobran sentido si la propuesta es compleja de utilizar y excluye a potenciales usuarios debido a condicionamientos arbitrarios.

En la Facultad de Informática de la Universidad Nacional de La Plata, desde hace tiempo venimos con un trabajo sostenido en torno a la accesibilidad que ha permitido construir una perspectiva que coloca la inclusión como un paradigma que significa construir oportunidades para todos y todas (Fernando Tauber, 2020).

Recientemente en una publicación, César Barletta plantea que el concepto de accesibilidad en la educación superior se refiere a la eliminación de barreras físicas, comunicativas y tecnológicas que pueden limitar o dificultar la participación plena de los sujetos en el proceso educativo. Busca garantizar que todos puedan acceder a la educación superior en igualdad de condiciones y disfrutar de una experiencia educativa inclusiva (Barletta, 2023, p. 131).

Desde estos planteos iniciales es que nos interesa, en el marco de este artículo plantear los aspectos normativos y legales que colocan a las instituciones de educación superior en el

compromiso de trabajar desde las inclusión y accesibilidad resguardando el derecho a la educación y por otro claro reconocer acciones y propuesta concretas de trabajo pedagógicos que pujan por aportar a la construcción de una ciencia informática que dialogue con las demandas y necesidades de una sociedad diversa y heterogénea.

### **Contexto**

En términos normativos, la Ley 24.521 sobre la Educación Superior de la Nación Argentina, que las universidades, los institutos universitarios y los institutos de educación superior deben respetar, estipula que se deben tener políticas de inclusión educativa tomando medidas para dar iguales oportunidades y posibilidades a las personas con discapacidad, y que puedan acceder al sistema sin discriminaciones de ningún tipo (Ley 24.521, 1995). A su vez, el espíritu del Estatuto de la propia Universidad Nacional de La Plata, que desde su preámbulo la define como una institución pública y gratuita de educación superior, que se debe ofrecer abierta e inclusiva para toda la sociedad estableciendo políticas que tiendan a facilitar el ingreso, permanencia y egreso de los sectores más vulnerables de la sociedad (Estatuto de la Universidad Nacional de La Plata, 2009). También indica, la ejecución de políticas con el objeto principal de propender al mejoramiento constante de la calidad de vida de los integrantes de la comunidad universitaria, a la vez que garantiza la efectiva igualdad de oportunidades para el acceso a la educación superior.

A nivel de accesibilidad digital, la Argentina cuenta con un marco legislativo desde el año 2010 cuando se aprobó por unanimidad la Ley Nacional 26.653 sobre accesibilidad de la información en las páginas web. La misma establece que los sitios Web que dependen directa e indirectamente del Estado Nacional deben ser accesibles, para que toda persona pueda acceder a sus contenidos sin excepciones, con independencia de sus capacidades perceptuales y motrices, garantizando igualdad de oportunidades y trato (Ley 26.653, 2010), cumpliendo con las pautas de *Web Content Accessibility Guidelines (WCAG) W3C* para el contenido Web (Web Content Accessibility Guidelines [W3C], 2008). A su vez, en enero de 2019 en la provincia de Buenos Aires, se promulgó la Ley Provincial 15.115, que establece la obligatoriedad del cumplimiento de normas y requisitos de accesibilidad a la información en el diseño de páginas web, en el marco de la ley nacional mencionada anteriormente (Ley 15.115, 2019). La accesibilidad es un proceso que es transversal a todas las etapas de desarrollo de software, sin embargo la adopción de un producto de software libre requiere de validaciones para luego corregir los aspectos que sean necesarios.

El tema de Accesibilidad Web está siendo estudiado por un grupo de investigadores del LINTI, Laboratorio de Investigación en Nuevas Tecnologías Informáticas, desde hace más de 20 años, analizando las normas existentes y llevando a cabo tareas en conjunto relacionadas con la

sensibilización y concientización de la problemática actual de la persona con discapacidad en la Web. En el año 2010 se creó, dentro de la Facultad de Informática, la Dirección de Accesibilidad con el objetivo de trabajar por una Facultad que incluya, integre y sea accesible a toda la ciudadanía, y también con la finalidad de ofrecer igualdad de condiciones a toda la comunidad educativa. Entre todas las actividades que se llevan a cabo, se vienen desarrollando proyectos de extensión acreditados por la Universidad Nacional de La Plata, en el que participan varias escuelas secundarias de la ciudad y la región. Desde el año 2011, la Dirección de Accesibilidad dicta cursos abiertos y gratuitos en el marco de la Secretaría de Extensión de la Facultad, trabajando aspectos relacionados con la introducción de accesibilidad digital y la generación de materiales accesibles. También, se llevan a cabo charlas a docentes, nodocentes y estudiantes para concientizar sobre la importancia del tema. Además, se dictan cursos de postgrado y de capacitación para otras instituciones, que se explican en la siguiente sección.

Todos los procesos de formación que se vienen impartiendo tienen como eje el reconocer la relevancia de trabajar la inclusión y accesibilidad, entendiendo que la enseñanza es la herramienta para plantear nuevos problemas que desafían la formación profesional.

#### **Formación integral en Accesibilidad Web Perspectivas conceptuales:**

En el marco de este apartado vamos a dar cuenta de prácticas concretas desde las cuales trabajamos la inclusión y accesibilidad para ello es interesante sumar algunos aportes conceptuales más, en ese sentido reconocemos que el planteamiento y andamiaje de la accesibilidad se configura desde distintos enfoques, abordajes y acciones. Las políticas de acceso y “calidad” para una educación inclusiva deben formar parte de un planteo estratégico para favorecer la disminución de la desigualdad económica y social de las personas más desfavorecidas de la sociedad (Gross Martinez, 2016, pp. 1-17).

También comprendemos que la accesibilidad, constituye un elemento fundamental en la calidad de vida de las personas con discapacidad, siendo un medio imprescindible para la integración y participación en la vida pública, y un presupuesto ineludible para el ejercicio de derechos fundamentales, como la libertad de circulación, de comunicación y de expresión, entre otros (Tejada Ríos, 2021, p. 93).

Haciendo foco en la problemática desde la cual trabajamos es importante destacar que la accesibilidad en la educación superior no se limita únicamente a aspectos físicos y tecnológicos, sino que también implica una mentalidad y cultura inclusiva en las instituciones educativas (Barletta, 2023, p. 132).

Tomando estos aportes es que desde diferentes espacios académicos e institucionales desplegamos prácticas educativas desde esta perspectiva.

## Formación en el Ingreso El Ingreso

El “Taller de Inserción a la Vida Universitaria” (TIVU) es parte de la estrategia de ingreso a las carreras de la Facultad de Informática de la UNLP. Este espacio es de carácter obligatorio para las carreras de la Facultad de Informática, en el mismo se aborda la problemática de ser estudiante en una universidad pública como la UNLP y en particular en la Facultad de Informática.

A partir de pensar las prácticas instituyentes e instituidas, como también los procesos de cambio e innovación que llevan adelante los sujetos involucrados permite por un lado comprender, pero también proyectar y fundamentalmente construir lo nuevo (Ungaro, Harari, Amadeo, Alman, 2023, p. 133).

Desde este enfoque ser estudiante implica un proceso de construcción de los sujetos-estudiantes. En el TIVU la intención es pensar la disciplina desde el sentido social de los futuros informáticos con el propósito de abordar las dimensiones éticas y axiológicas del proceso. Los “recién llegados” (Arendt, 2005), comienzan a transitar un espacio institucional que está poblado de experiencias nuevas con efectos formativos, son tramados por la lógica disciplinar, cultura institucional y pedagógica que se explicitan en prácticas educativas novedosas.

Esta mirada compleja permite considerar a las diversas áreas del conocimiento y perspectivas para pensar una institución que incluya sujetos diferentes, diversos y heterogéneos. Con esta premisa, el TIVU busca tender puentes con los sujetos involucrados, a partir de la organización de talleres donde se abordan problemáticas propias del “ser estudiante en la facultad”. El trabajo en taller permite construir un espacio de formación y socialización que promueve el encuentro entre docentes y alumnos en un diálogo horizontal. Es aquí donde la potencia de la comunicación, entendida como construcción de subjetividad y sentido que adquiere valor al constituirse en un espacio de diálogo pedagógico permanente.

El TIVU se desarrolla en 4 encuentros y su carga horaria es de 16 horas. Sus destinatarios son los ingresantes a todas las carreras de la Facultad que en el 2023 rondarán los 2000 estudiantes. Los encuentros abordan diferentes problemáticas relacionadas con la subjetividad de ser estudiante en la universidad, sintéticamente podemos mencionar como ejes centrales a: el marco histórico institucional de la UNLP y de la Facultad, por otro lado desde el relato de experiencias estudiantiles los modos de participación y ser estudiante en Informática, también se aborda el trabajo del informático a partir de relatos de graduados y por último se trabaja la problemática de género y derechos humanos y accesibilidad. Cabe destacar que los contenidos seleccionados buscan, a través de un trabajo participativo y grupal.

En este sentido, pensar una universidad que incluya requiere reconocer que hay saberes no

disciplinarios, pero si académicos que hacen a la cultura institucional y que es necesarios que se constituyan en un habitus, para que los ingresantes puedan transitar de manera fluida por la institución, por la carrera y por su proceso formativo que inician con el ingreso. El diálogo e intercambio con el otro es importante no sólo por la afiliación a un grupo sino también porque sin diálogo no hay educación (Freire, 1970).

### ***Formación en Grado***

En la Facultad de Informática se incorporó esta temática en el plan de estudios de las carreras que se dictan en la institución en el año 2002, sólo tres años después de que se publicaran las primeras normas WCAG (WCAG 1.0, 1999), las que son reconocidas como el estándar de facto a nivel internacional en cuanto a accesibilidad web. Los conceptos relacionados a este tema fueron introducidos a través de la asignatura Diseño Centrado en el Usuario, con el objetivo de que los futuros profesionales comiencen a tomar conciencia sobre la importancia de considerar una tecnología más inclusiva. En esta asignatura que es optativa de 5to.año de las carreras de Licenciatura en Informática y Licenciatura en Sistemas de la facultad, se abordan también cuestiones de usabilidad como un factor importante para acercar las implementaciones informáticas a todas las personas, más allá de sus condiciones, de sus conocimientos y de su edad. Esto va en concordancia con la posterior creación en el año 2009 de la “Red Interuniversitaria Latinoamericana y del Caribe sobre Discapacidad y Derechos Humanos”, que en uno de sus principios proclama incorporar a la currícula de disciplinas universitarias y asignaturas, contenidos formativos referidos a la discapacidad desde la perspectiva de los derechos humanos (Méndez, Krause Arriagada, García, Solano Meneses, Cabeza, y Araguas, 2023).

### ***Formación en Postgrado***

Desde el año 2020 se viene dictando un curso de postgrado en la Facultad de Informática, sobre Usabilidad y Accesibilidad en el ámbito Informático; es virtual y sincrónico y otorga créditos para el Doctorado en Ciencias Informáticas. Está destinado a profesionales provenientes de diferentes carreras, siendo la mayoría del área informática. El curso trabaja sobre la premisa de que los verdaderos avances tecnológicos se dan cuando la tecnología mejora la calidad de vida de las personas y se encuentra al alcance de las mismas, cuando se desarrollan aplicaciones

simples, fáciles de usar y aprender, que asuman poco costo de entrenamiento y adaptación, y que admitan la diversidad de usuarios y sus diferentes escenarios de interacción. En el módulo de Usabilidad se pretende formar y brindar las herramientas necesarias para evitar que los típicos problemas de interacción que aún subsisten en los nuevos desarrollos disminuyan y, de esta manera, mejorar la percepción y experiencia del usuario frente a los distintos sistemas. En el módulo de Accesibilidad se brindan las normativas y estándares internacionales que rigen las normas de accesibilidad universales. Además, se muestran y utilizan un conjunto de herramientas para validar el cumplimiento de estas normas y en base a esto decidir sobre aspectos de diseño y cuestiones de usabilidad, teniendo en cuenta la inclusión como premisa principal.

### ***Formación al Finalizar la Carrera***

Estas actividades ininterrumpidas que impulsan el reconocimiento y tratamiento de la Accesibilidad en forma institucional, trae aparejado consecuencias muy positivas, no sólo en la relación Facultad de Informática-comunidad sino hacia el interior de la misma, generando conciencia y compromiso entre el estudiantado de Informática.

Esto se ve reflejado en las más de veinte tesinas desde 2002 hasta la fecha sobre accesibilidad y discapacidad. Algunos con reconocimiento institucional como Mención de Interés General por la Cámara de Diputados por el trabajo de tesis “Portal para ciegos y minusválidos visuales” (2002), el Primer Premio de Gobierno Electrónico por la institución SADIO en 2008, por el trabajo “Accesibilidad en portales gubernamentales de la Pcia de Buenos Aires”, o el premio ITU SUMSUNG por “Talk Louder- un comunicador para personas sordas” (2017), entre otros.

Entre otros trabajos muy importantes por su impacto social, se encuentran las tesinas: “Reconocimiento Automático de Texto en Braille” (2006); “Estrategias para la utilización de tecnología en educación especial” (2009); “Accesibilidad en Redes Sociales” (2011); “Engineering Accessible Web Applications: An aspect-Oriented Approach” (2011); “Colaboración y comunicación aumentativa mediada por TIC” (2013); “e-Quino: un videojuego interactivo para complementar la terapia asistida por caballos” (2014); “Calzado háptico: navegabilidad asistida para personas con disminución visual” (2014); “Accesibilidad y crowdsourcing: uso de etiquetas semánticas para mejorar la accesibilidad de aplicaciones web” (2015); “SiMor: analizador intensivo de accesibilidad web basado en reglas” (2015); “Accesocial, una aplicación móvil para determinar lugares accesibles en La Plata” (2016); “TraCS

-trabajo, colaboración y seguimiento- Una herramienta de seguimiento colaborativa de tratamientos de pacientes con discapacidad” (2016); “SENDA Herramienta Informática educativa para Dificultades de Aprendizaje” (2017); “TransitaWeb: un sitio que accesibiliza sitios Web”

(2017); “Navegación Web asistida por comando de voz” (2017); “DeafQuiz Juego educativo para la enseñanza de LSA a niños” (2018); “e-tutor: mesa de ayuda virtual para el Programa de Tutorías” (2019); “Chatbot Accesible para el Sector Académico” (2019); “Kit basado en sensores para personas ciegas” (2020); “Aplicación mobile de acompañamiento terapéutico en el tratamiento de los trastornos del lenguaje, del habla y de la alimentación” (2022); “Accesibilidad en videojuegos: su aporte a la inclusión y al proceso de creación de videojuegos inclusivos en la industria independiente” (2023); “Accesibilidad en documentos para personas con discapacidad visual y ceguera en CILSA” (2023).

Estas tesinas y otras más se encuentran publicadas en el Servicio de Difusión de la Creación Institucional (SEDICI), que es el Repositorio Institucional de la UNLP (SEDICI, 2003).

### ***Formación al Interior de la Comunidad Universitaria***

Desde la Dirección de Accesibilidad, se llevan a cabo en forma continua diferentes actividades con los integrantes de la institución: docentes, auxiliares, agrupaciones estudiantiles, tutores de los cursos y nodocentes. Las actividades consisten en charlas y reuniones donde es posible la difusión y concientización de la importancia de la accesibilidad en todos los aspectos. Se realizan encuentros con los docentes de las cátedras para hablar sobre la problemática puntual de cada una, cómo llevar a cabo la enseñanza de los temas, cómo asistir a los estudiantes que tienen algún tipo de discapacidad y lo que es muy importante cómo generar recursos educativos accesibles. Se comenzó a trabajar en la tarea de accesibilizar los materiales de las materias de primer año por ser los más demandados por las y los estudiantes en general y en particular por los que cuentan con alguna discapacidad, esto comprende la digitalización de libros y el subtítulo de videos de clases y explicaciones. También se realizan reuniones con el personal de las distintas dependencias, como la Dirección de Enseñanza para optimizar los trámites de inscripción y del seguimiento académico de los estudiantes. Se llevan a cabo capacitaciones a tutores y demás personas interesadas en la problemática de accesibilidad, para que puedan responder a las demandas requeridas. El objetivo de estas actividades es poder escuchar y conocer lo que cada persona como integrante de la Facultad en el rol que le toca, ya sea docente, nodocente o alumno “piensa de la discapacidad” (Katz, 2023).

### ***Formación a Toda la Comunidad***

Respecto del curso sobre Accesibilidad Web a cargo de la Dirección de Accesibilidad, es importante mencionar que se viene desarrollando la decimoquinta edición, y está orientado tanto a personas sin conocimientos de informática, como a programadores, que están interesados en aprender sobre esta temática para aplicarla cada uno en su espacio de trabajo. Es un curso totalmente virtual, que generalmente se dicta en la segunda parte del año, gratuito y abierto a

toda la comunidad, de unas 64 horas de carga horaria y con certificado de aprobación. En este curso se brindan los conceptos básicos de Accesibilidad, ejemplos, normativas, casos de estudio, casos reales de personas con discapacidad, mostrados a través de videos y texto. En el curso se promueve a los participantes a realizar aportes sobre experiencias de vida, ponerse en el lugar del otro e interactuar con la web como si fuera una persona con discapacidad, intervenir en debates y participar en la construcción del concepto de accesibilidad en forma integral y multidisciplinar. Por el momento se capacitaron a más de 900 personas en forma gratuita. Personas que trabajan en el área de Informática tanto en instituciones gubernamentales como privadas, de nacionalidad argentina como de otros países tales como Uruguay, Paraguay, Perú, Ecuador, Colombia, Venezuela, México y España. Es importante destacar que en varias de las ediciones del curso, han participado personas con algún tipo de discapacidad visual, lo cual nos permitió sumar sus aportes y experiencias en relación al contenido y a sus vivencias personales. En las encuestas realizadas al finalizar cada uno de los cursos, los participantes expresaron que los conceptos adquiridos, y el contenido brindado fue de gran utilidad para su actividad laboral y su formación profesional. El curso se promociona a través de la página web de la Secretaría de Extensión de la Facultad de Informática ([www.extension.info.unlp.edu.ar](http://www.extension.info.unlp.edu.ar)) y se puede solicitar información al correo [accesibilida@info.unlp.edu.ar](mailto:accesibilida@info.unlp.edu.ar).

### ***Formación en el Nivel Secundario***

Desde el año 2013 se vienen desarrollando en la Facultad de Informática diversos proyectos de Extensión acreditados por la Universidad Nacional de La Plata, que comprenden diferentes actividades como jornadas, talleres, charlas y asesoramiento sobre el tema. Algunos de estos proyectos son: “Trabajando por una Web Accesible”, “Por una Web inclusiva”, “Por una Web inclusiva: abordaje en escuelas secundarias” (Dictamen de Proyectos de Extensión, s.f.) Las intervenciones en las escuelas secundarias con el fin de apoyar, fomentar y formar sobre la inclusión en la Web, fueron siempre desarrolladas teniendo en cuenta distintas perspectivas, disciplinas y roles. El accionar extensionista se fue propagando articulando con otros proyectos y entidades, permitiendo una comunicación y crecimiento conjunto. El grupo de trabajo fue creciendo a lo largo de los años y se fueron incorporando personas con discapacidad, algunos estudiantes ciegos, con sordera y un estudiante con Síndrome de Asperger. Es importante la participación de las instituciones y entidades que trabajan y nuclean personas con discapacidad, que siempre nos apoyan y colaboran en este proceso de llevar y trabajar con las escuelas secundarias el tema de inclusión en las tecnologías. El objetivo es promover e incentivar en los jóvenes de la escuela secundaria la concientización por la diversidad, la empatía, la inclusión y accesibilidad en las tecnologías, en especial en la Web. Camino en conjunto con la comunidad secundaria que se inició en el 2013 y que aspira mantenerse, sostenerse en el tiempo e institucionalizarse de manera tal que constituya un vínculo ininterrumpido.

### ***Formación Externa en Otras Entidades***

Se han llevado a cabo durante varios años, diferentes cursos para organizaciones y universidades. Algunos cursos se basaron en Accesibilidad en recursos digitales educativos, y eran destinados a profesionales del área informática, docentes, bibliotecarios, y todas aquellas personas que tenían la responsabilidad y necesidad de generar recursos educativos que fueran accesibles. En estos cursos se planteó la problemática de la accesibilidad al utilizar el material educativo y se brindaron las bases para la construcción tanto de documentos como de videos que cumplieran las pautas de accesibilidad para que pudieran ser utilizados por todos los estudiantes. Se incluyeron las normativas y diferentes herramientas que permiten accesibilizar los distintos tipos de archivos.

Todos los procesos de formación que se vienen impartiendo tienen como eje el reconocer la relevancia de trabajar la inclusión y accesibilidad, entendiendo que la enseñanza es la herramienta para plantear nuevos problemas que desafían la formación profesional. Los resultados de las actividades son muy alentadores y satisfactorios, donde se demuestra que la accesibilidad web no es un concepto teórico y utópico. Los participantes comienzan a tomar conciencia de las barreras existentes en el acceso a las tecnologías, desarrollando sentimientos de empatía al ver experiencias concretas de gente con algún tipo de discapacidad.

### **Conclusiones**

Recuperamos algunas ideas planteadas:

- Pensar una universidad que incluya, requiere reconocer que hay saberes disciplinares, académicos y no disciplinares que hacen a la cultura institucional que es necesario enseñar.
- La accesibilidad, constituye un elemento fundamental en la calidad de vida de las personas con discapacidad, siendo un medio imprescindible para la integración y participación en la vida pública, y un presupuesto ineludible para el ejercicio de derechos fundamentales, como la libertad de circulación, de comunicación y de expresión.
- Dentro del desarrollo web, la accesibilidad no está ligada exclusivamente al área de desarrollo y programación. Todos los roles están involucrados, desde los contenidistas, diseñadores, comunicadores hasta los evaluadores de calidad de software. Es una temática transversal a diferentes disciplinas por lo que su abordaje debe ser integral.
- Las políticas de acceso y “calidad” para una educación inclusiva deben formar parte de un planteo estratégico para favorecer la disminución de la desigualdad.

Es aquí donde el trabajo sostenido desde diferentes propuestas y con los diferentes sujetos institucionales se hace fundamental no sólo para la relevancia de la comunicación de esta problemática sino también por el compromiso de entender que la educación es un derecho que implica la construcción de subjetividad y sentido en el marco de un espacio pedagógico de diálogo y reflexión permanente.

## Referencias

Arendt, H. (2005), "La condición Humana". Paidós.

Barletta, Cesar Martin. Inclusión y accesibilidad en la Educación Superior: estrategias para promover un entorno educativo inclusivo *Trayectorias Universitarias*, 9 (16), e129, 2023 ISSN 2469-0090 | <https://revistas.unlp.edu.ar/TrayectoriasUniversitarias> Universidad Nacional de La Plata La Plata | Buenos Aires | Argentina

Dictamen de Proyectos de Extensión acreditados en la UNLP, [https://unlp.edu.ar/proyctosext/proyectos\\_de\\_extension\\_acreditados\\_y\\_subsidiados-4708](https://unlp.edu.ar/proyctosext/proyectos_de_extension_acreditados_y_subsidiados-4708)

Estatuto de la Universidad Nacional de La Plata [Honorable Asamblea Universitaria]. 28 de Junio de 2008. Disponible en: [https://unlp.edu.ar/institucional/unlp/gobierno/estatuto\\_unlp-4287-9287/](https://unlp.edu.ar/institucional/unlp/gobierno/estatuto_unlp-4287-9287/)

Freire, Paulo (1970). *Pedagogía del oprimido*. Siglo XXI Editores. ISBN-10: 9682302366. ISBN-13 : 978-9682302367.

Gross Martínez, M. (2016). Accesibilidad al proceso educativo en el entorno universitario. *Revista: Actualidades Investigativas en Educación*, Volumen 16 Número 1, pp. 1-17. Año 2016, ISSN 1409-4703. DOI: <http://dx.doi.org/10.15517/aie.v16i1.21920>

Katz, S. L. (2023). ¿Por qué hay personas con discapacidad en la UNLP?.

*Trayectorias Universitarias*, 9(16), 117. <https://doi.org/10.24215/24690090e117>

Ley 15.115 de 2019 de la Provincia de Buenos Aires. 14 de Enero de 2019. Disponible en: [https://intranet.hcdiputados-ba.gov.ar/includes/ley\\_completa.php?vnrole=15115](https://intranet.hcdiputados-ba.gov.ar/includes/ley_completa.php?vnrole=15115)

Ley 24.521 de 1995. Ley de Educación Superior. 20 de Julio de 1995. Disponible en: <https://www.argentina.gob.ar/normativa/nacional/ley-24521-1995-25394>

Ley 26.653 de 1995. Acceso a la Información Pública. 3 de Noviembre de 2010. Disponible en: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/175000-179999/175694/norma.htm>

Mendez, M., Krause Arriagada, A., García, G. I., Solano Meneses, E. E., Cabeza, J., & Araguas, N. (2023). Red Interuniversitaria Latinoamericana y del Caribe sobre Discapacidad y Derechos Humanos: Tejiendo lazos para la transformación. *Trayectorias Universitarias*, 9(16), 118. <https://doi.org/10.24215/24690090e118>

- SEDICI, Servicio de Difusión de la Creación Intelectual, 2003. Disponible en: <http://sedici.unlp.edu.ar/>
- Tauber, F. (2020). El desafío de gestionar en un escenario de pandemia. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/108475>)
- Tejada Ríos, J. (2021). Accesibilidad de las personas con discapacidad en España y Portugal: perspectiva jurídica. Revista Jurídica de Investigación e Innovación Educativa, (24), 93-114. <https://doi.org/10.24310/REJIE.2021.vi24.12140> 10
- Ungaro Ana, Ivana Harari, Paola Amadeo, Anahí Alman. Pensar la accesibilidad en la educación universitaria: concepciones y perspectivas en diálogo con la experiencia Trayectorias Universitarias, 9 (16),e128, 2023 ISSN 2469-0090 | <https://revistas.unlp.edu.ar/TrayectoriasUniversitarias> Universidad Nacional de La Plata La Plata | Buenos Aires | Argentina
- Ungaro Ana-2023 Taller de Inserción a la Vida Universitaria la experiencia de pensar una estrategia de inclusión estudiantil durante la pandemia. Autor: Esp. Ana María Ungaro Escenarios y recorrido educativos-Compilación de experiencias y buenas prácticas de virtualización en carreras de grado y pregrado de la UNLP. SIED-UNLP (en prensa)
- Web Content Accessibility Guidelines 1.0, W3C Recommendation, 5 de Mayo de 1999. Disponible en <https://www.w3.org/TR/WCAG10/>
- Web Content Accessibility Guidelines 2.0, W3C Recommendation, 11 de Diciembre de 2008. Disponible en <https://www.w3.org/TR/WCAG20/>

**Comarca Digital:  
Impulsando la enseñanza de la Computación en el Nivel Inicial**

Cristián Rojo Pérez, Natalia Andrea Zalazar  
{crojoperez715, nzalazar}@isep-cba.edu.ar

Instituto Superior de Estudios Pedagógicos.  
Ministerio de Educación de la Provincia de Córdoba

### **Resumen**

El presente trabajo se enmarca en un proyecto educativo para la enseñanza de ciencias de la computación, dependiente de una plataforma digital de recursos para la enseñanza de la jurisdicción provincial. Los materiales producidos en este marco conforman una colección de secuencias y recursos didácticos publicados en la segunda mitad del año 2022, orientados a la enseñanza de la computación en el Nivel Inicial del sistema educativo de la provincia de Córdoba. Se puede acceder a ellos por medio de internet y cuentan con licencias libres Creative Commons (CC-BY-NC-SA 4.0).

En este artículo, se presentan los antecedentes, demandas y necesidades que motivaron este proyecto así como las decisiones pedagógicas y didácticas en relación a los contenidos a abordar, la propuesta metodológica y el acompañamiento a docentes del nivel que no han recibido formación específica en la enseñanza de la computación.

El estudio de las tecnologías forma parte de la currícula del Nivel Inicial de la provincia de Córdoba, de manera más específica, las tecnologías digitales se han abarcado mayoritariamente desde un enfoque vinculado a las TIC, dado que se alienta a trabajar utilizando programas computacionales como medios para abordar otros contenidos (de las áreas de matemática, lengua y literatura, ciencias naturales y sociales, etc.). Otra característica de las instituciones públicas del nivel, es que las mismas cuentan con equipamiento tecnológico disponible en sus salas, el ministerio ha provisto un kit que incluye un conjunto de tablets, una notebook y un robot educativo, entre otros dispositivos de sonido y proyección.

Es por ello que, atendiendo a la preocupación por incorporar saberes disciplinares que permitan ir introduciendo conceptos en los que la computadora se posicione como objeto de estudio y aprendizaje, se trabajó colectiva e interdisciplinariamente entre personas informáticas, especialistas en educación y un equipo de producción de materiales en línea en el desarrollo y publicación de las secuencias que serán presentadas en este artículo. Esta producción fue realizada por grupos de trabajo que se inscriben en el Instituto Superior de Estudios Pedagógicos (ISEP)<sup>1</sup>.

---

<sup>1</sup> <https://isep-cba.edu.ar/>

**Palabras clave:** enseñanza de las ciencias de la computación, nivel inicial, secuencias didácticas.

## Introducción

Con intención de profundizar saberes vinculados a las tecnologías digitales, el Ministerio de Educación de la Provincia de Córdoba ha desarrollado para 2022-2023 el Plan Leer, Escribir y Pensar en la Cultura Digital<sup>2</sup>, como manera de implementar la enseñanza de los contenidos propuestos por los Núcleos de Aprendizaje Prioritarios (NAP) de educación digital, programación y robótica, según la Resolución N° 343/18. Los contenidos propios de este plan para el Nivel Inicial integran tanto prácticas de uso de los dispositivos y programas como habilidades de resolución de desafíos mediante la construcción de algoritmos. Más recientemente, en marzo de 2023 se presentó una actualización curricular<sup>3</sup> que propone la incorporación de contenidos de Ciencias de la Computación, comenzando por el nivel inicial y proponiendo una gradualidad en los saberes que alcanza al ciclo básico de educación secundaria.

Por otra parte, una de las preguntas recurrentes cuando nos proponemos enseñar conceptos de Ciencias de la Computación es el acceso a equipamiento adecuado para trabajar de un modo práctico. Muchas veces la falta de acceso o las amplias brechas en la presencia o ausencia de hardware, o incluso su estado, hacen de la búsqueda de igualdad de oportunidades una tarea compleja.

En el caso de la provincia de Córdoba, en consonancia con el programa nacional, se equipó a las instituciones públicas de Nivel Inicial a partir del año 2019 con un kit que contiene:

- 1) Un robot comercial producido con finalidades educativas llamado Blue-Bot o “Robotita”, como se lo nombra popularmente en las salas de la provincia. Este dispositivo permite recibir instrucciones para desplazarse en el espacio, ya sea por medio de los botones de dirección que tiene incorporados en su estructura o por medio de la interfaz Bluetooth que lo comunica a un programa que debe estar instalado en las tablets o teléfonos con Android y que permite su programación en un lenguaje de bloques. Además, está acompañado de un conjunto de tapices con diversas propuestas que permiten simular situaciones de la vida cotidiana por las cuales debe desplazarse el robot.
- 2) Un conjunto de tabletas que traen programas instalados para programar (Scratch Jr y Blue Bot) así como programas para dibujar, crear personajes y un banco de recursos didácticos para navegar.

---

<sup>2</sup> <https://www.igualdadycalidadcba.gov.ar/SIPEC-CBA/Programas/2022/culturaDigital.php>

<sup>3</sup> <https://www.igualdadycalidadcba.gov.ar/SIPEC-CBA/publicaciones/DyPCurriculares/DAC/2023/aportes-de-cultura-digital-en-la-edu-tecno-ac.pdf>

- 3) Una notebook.
- 4) Un proyector con sistema de sonido.
- 5) Una pizarra digital interactiva.
- 6) Un *access point* con contenidos educativos

Además, en algunos casos las instituciones cuentan con infraestructura de conectividad y servicio de internet.

La provisión de equipamiento puede ser aprovechada de manera significativa en la medida que la política pública acompañe con formación docente y con materiales didácticos que orienten la enseñanza en estos temas. Esto último es lo que ofrece la plataforma digital de recursos para la enseñanza del Ministerio de Educación de la provincia llamada recientemente “Hacemos Escuela”<sup>4</sup> que fue creada en la pandemia bajo el nombre “Tu Escuela en Casa”. En ella, se pueden encontrar secuencias didácticas para todos los niveles del sistema educativo, en múltiples espacios curriculares, que han sido escritas por los departamentos disciplinares de ISEP, instituto a cargo de la iniciativa. Hacemos Escuela es un portal educativo que ha sido reconocido por Unicef por tomar una perspectiva no determinista de la tecnología y por hacer foco en la comprensión de su funcionamiento y el análisis sociocultural de los medios digitales. (Unicef, 2022).

Con los recursos mencionados, y con el carácter entusiasta e innovador que caracteriza al colectivo docente del Nivel Inicial, recibiendo -en algunos casos- el acompañamiento de la dirección de Educación Digital de la provincia, muchas salas se apropiaron de los recursos tecnológicos provistos y realizaron experiencias de manipulación de las tabletas y del robot. La revisión de los registros de dichas experiencias, que fueron recopilados por la dirección de Educación Digital, permite observar que en todos los casos, el uso de las tecnologías promovió situaciones educativas significativas. Sin embargo, en estos casos las computadoras son tomadas como recursos didácticos en lugar de objetos de estudio en sí. El uso del robot o la tableta se transforma en un medio para aprender sobre algún otro contenido: las normas viales, los tipos de cuentos, la huerta, entre otros. Es decir, no se pudo detectar en las experiencias relevadas la presencia de contenidos propios de un campo disciplinar que estudia cómo son, cómo funcionan y cómo se programan los dispositivos computacionales.

En este marco, se planteó la necesidad de desarrollar materiales didácticos que tomaran en cuenta la vacancia de recursos para la enseñanza de la informática en el nivel inicial, enfocando los objetivos en el aprendizaje de algunos fundamentos básicos del campo disciplinar.

---

<sup>4</sup> <https://hacemosescuela.cba.gov.ar/>

Si el nivel inicial es el lugar donde por primera vez los niños y las niñas asisten al sistema escolar, y en este comienzan a tomar contacto con las normas sociales y los aprendizajes fundamentales de la vida en comunidad, merece entonces que también sea la puerta de entrada al entendimiento de cómo funcionan las computadoras y a una aproximación crítica sobre sus usos debido a la gran cantidad de procesos sociales mediados por tecnologías computacionales que nos atraviesan diariamente.

Ante esta necesidad, subyace un desafío importante: generar materiales que resulten lo suficientemente precisos en lo disciplinar, pero al mismo tiempo, entendibles y realizables en las salas por docentes que no tuvieron formación sobre computación ni sobre su didáctica.

### **El encuadre del proyecto, desde los lineamientos nacionales a los jurisdiccionales**

Para el desarrollo de la propuesta fue relevante integrar las particularidades del nivel inicial, reflejadas en aportes teóricos, así como en disposiciones jurisdiccionales y nacionales que se incluyen en el propio diseño curricular.

En su tesis de maestría, Torres (2021) rescata aquellos lineamientos que la Ley Nacional de Educación 26.206 plantea para el nivel inicial y que fueron tomados en cuenta para el diseño de los materiales didácticos que este trabajo presenta. En especial, fueron considerados los siguientes objetivos de los que se mencionan en el Artículo 20 de dicha normativa:

c) Desarrollar su capacidad creativa y el placer por el conocimiento en las experiencias de aprendizaje. d) Promover el juego como contenido de alto valor cultural para el desarrollo cognitivo, afectivo, ético, estético, motor y social. e) Desarrollar la capacidad de expresión y comunicación a través de los distintos lenguajes, verbales y no verbales: el movimiento, la música, la expresión plástica y la literatura. g) Propiciar la participación de las familias en el cuidado y la tarea educativa promoviendo la comunicación y el respeto mutuo.

Este mismo autor recupera también las ideas de Soto y Violante (2011) quienes proponen los pilares de la Didáctica de la Educación Inicial, los cuales pueden sintetizarse como: a) la centralidad del juego, b) la multitarea y el trabajo en pequeños grupos como modalidad organizativa privilegiada, c) la enseñanza centrada en la construcción de escenarios, d) el principio de globalización-articulación de contenidos como modo de reunir aportes de los diferentes campos de conocimiento alrededor de ejes organizadores significativos para los niños. e) el desarrollo personal y social y la alfabetización cultural, dimensiones de una Educación Integral. f) la conformación de lazos de sostén, confianza, respeto, complementariedad con el niño y las familias (Torres, 2021).

Además de los aportes mencionados, uno de los documentos de referencia para generar las

secuencias didácticas, fue el Diseño Curricular del Nivel Inicial de la Provincia de Córdoba vigente a la fecha<sup>5</sup> En él encontramos un modo de trabajo donde las disciplinas que se enseñan tienen un amplio margen para ser trabajadas de forma conjunta y coordinada. Esto demandó que las secuencias didácticas asuman el desafío de no quedarse solo en lo disciplinar de la computación, sino también buscar nexos con otros aprendizajes propios del nivel.

Tomando en cuenta lo anterior, y bajo la premisa de que el aprendizaje integral se debe dar en un ambiente participativo que habilite un rol activo en los y las estudiantes, es que las secuencias buscaron ofrecer actividades que conectaran sus experiencias previas con los conocimientos de computación vistos como parte de un sistema social complejo, del que ellos y ellas, el jardín y sus familias son parte. En este sentido el diseño curricular vigente dice:

Es importante que se trabaje la complejidad del ambiente y también su dinamismo, a partir de la curiosidad, estimulando la formulación de interrogantes que contemplen los qué, los cómo, los cuándo, tendiendo a incorporar los por qué y para qué de los fenómenos y hechos que ocurren en el espacio cercano al niño (Ministerio de Educación, 2011, p. 52).

Otro de los puntos de partida, tras conversar con docentes y directoras del nivel, fue la recomendación de generar actividades que tengan un carácter concreto y que utilicen, sobre todo al inicio, el cuerpo y las tecnologías previamente conocidas, para luego desde ahí, pasar a las actividades propias de la computación.

Entre los objetivos declarados en los lineamientos que ofrece el mencionado diseño curricular encontramos para salas de 3 y 4: "Identificar, a través de la exploración sensorial, cómo son los objetos, materiales, herramientas e instrumentos del ambiente cercano." (p. 51). Para sala de 5: "Relacionar las características de los objetos, materiales y herramientas del ambiente cercano con sus usos." (p. 46).

Específicamente en los campos de conocimiento integrados de Cs. Sociales, Cs. Naturales y Tecnología se plantea la "Identificación de la relación de los productos tecnológicos del ambiente cercano con las necesidades que satisfacen." y el "Reconocimiento de los objetos como resultado de distintos procesos de producción." (p. 51).

Siguiendo con esta línea, se buscó que el trabajo colaborativo, de indagación y de prueba y error, guiado por las y los docentes tuviesen un anclaje en las "Prácticas de cooperación y colaboración" (p. 52) prescriptas para el abordaje de Tecnología en las salas de 3, 4 y 5.

Otra de las claves, siguiendo la idea anterior proviene de los objetivos de Ciencias Sociales para las salas de 3, 4 y 5 que proponen la "Valoración del diálogo y la escucha como prácticas

---

<sup>5</sup> <https://www.igualdadycalidadcba.gov.ar/SIPEC-CBA/publicaciones/DPCurriculares-v2.php>

necesarias para el afianzamiento del respeto por las diversas opiniones y por las normas que organizan la vida en sociedad.” (p. 49).

Además, a partir del diagrama de objetivos se tomó en consideración para salas de 3 y 4: “Ampliar la noción de ambiente comenzando el reconocimiento de la diversidad social, natural y tecnológica.” y para sala de 5: “Identificar las principales invenciones que el hombre ha realizado para mejorar su calidad de vida.” (p. 44) así como “Identificar algunos materiales, herramientas, máquinas y artefactos inventados y usados en distintos contextos sociales.” (p. 46)

De manera complementaria, para establecer un posicionamiento sobre el abordaje de la propuesta didáctica, también han sido relevadas y tenidas en cuenta, algunas otras experiencias de actualización de diseños curriculares que incorporan en el nivel inicial saberes relativos a las ciencias de la computación. Se recuperó el caso de la Municipalidad de la ciudad de Córdoba, el caso de España que ha actualizado el curriculum para la educación infantil y que comenzó su aplicación en el año 2022 y el conocido Marco de las ciencias de la computación K-12 (2016).

En el caso de la Ciudad de Córdoba, la Secretaría de Educación de la municipalidad ha desarrollado una propuesta de aprendizajes y contenidos fundamentales para la inclusión de la Educación Digital, Programación y Robótica en la Educación Inicial y Primaria con implementación obligatoria a partir del año 2022. Para el nivel inicial, se propone incorporar un campo de conocimiento específico. Dicha propuesta curricular plantea que:

Se avanza desde el Nivel Inicial, sala de 4, con el propósito de incorporar, de manera temprana, las tecnologías en la trayectoria escolar a los fines de promover la alfabetización digital de manera gradual, crítica y creativa. En toda la propuesta, desde Sala de 4 a 6to grado, se plantea el trabajo con robótica y lenguajes de programación para el desarrollo del pensamiento computacional a través del planteo, análisis y resolución de situaciones problemáticas, con la intención de que los y las estudiantes comprendan, se apropien y construyan los saberes mediante la participación en exploraciones y actividades de resolución de problemas y desafíos (Secretaría de Educación, 2020, p. 1).

Los contenidos, para las salas de 4 y 5 años, están organizados en tres ejes y se observa cierta correspondencia con lo propuesto en los NAPs de Educación Digital, Programación y Robótica. El primero de los ejes, nombrado Exploración de sí mismo y el entorno, se enfoca hacia el uso de los dispositivos digitales para registrar el entorno y para identificar sus usos en la cotidianeidad, también se propone “la resolución de problemas aplicando una secuencia ordenada de acciones” (p. 2). El segundo eje, Imaginación, juego y creatividad, propone trabajar con producciones digitales en distintos formatos abriendo espacio a lo artístico y lo lúdico, también se hace una mención a conocimientos introductorios de robótica a partir de juegos de

construcción. El tercer eje, Aprendizajes, comunicación y convivencia digital apela a la lectura y escritura en formatos digitales incorporándose a otros formatos multimediales la “iniciación en el uso de códigos de programación.” (p. 3).

En otro de los casos, el Ministerio de Educación y Formación Profesional de España, actualizó en el año 2022 la normativa que establece las enseñanzas mínimas para la Educación Infantil. En esta normativa se prevé, por un lado, comenzar en el nivel inicial con la alfabetización digital determinado por los siguientes saberes básicos:

Aplicaciones y herramientas digitales con distintos fines: creación, comunicación, aprendizaje y disfrute. Uso saludable y responsable de las tecnologías digitales.

Lectura e interpretación crítica de imágenes e información recibida a través de medios digitales. Función educativa de los dispositivos y elementos tecnológicos de su entorno (Ministerio de Educación y Formación Profesional, 2022, p. 33).

Por otro lado, también se mencionan acercamientos al pensamiento computacional. Dentro del área de conocimiento descubrimiento y exploración del entorno, se propone la interacción con elementos del entorno más cercano por medio de la observación, la indagación, el análisis, comprobación y razonamiento; añade que cada estudiante “experimentará y desplegará progresivamente destrezas sencillas propias del método científico y del pensamiento computacional y de diseño.” (p. 22) y define las siguientes competencias:

El alumnado deberá encontrar soluciones o alternativas originales y creativas a diferentes cuestiones, retos o situaciones. Y lo hará mediante la aplicación de procesos inicialmente sencillos y manipulativos, que progresivamente ganarán en complejidad y requerirán mayor capacidad de abstracción. Dichos procesos son propios tanto de las destrezas de pensamiento computacional y de diseño como del método científico, y se aplicarán descomponiendo una tarea en otras más simples, formulando y comprobando hipótesis, explorando e investigando, relacionando conocimientos y planteando ideas o soluciones originales (p. 22).

Luego, como criterios de evaluación del alcance de estas competencias se sugiere específicamente “Programar secuencias de acciones o instrucciones para la resolución de tareas analógicas y digitales, desarrollando habilidades básicas de pensamiento computacional.” (p. 24)

En estos dos casos, se puede observar cierto enfoque compartido para la incorporación de los saberes digitales y computacionales dentro de las particularidades del aprendizaje en las edades tempranas.

### **Descripción de la propuesta**

Partiendo del encuadre y antecedentes mencionados anteriormente, se desarrollaron las primeras seis secuencias didácticas que conforman la colección denominada “Explorando la Comarca Digital” y que puede consultarse en el sitio web correspondiente. A continuación, se describen algunas decisiones didácticas que las orientan.

Esta propuesta toma el desafío de ofrecer una atractiva exploración del entorno digital que rodea tanto a niños y niñas como a toda la comunidad educativa. En cada secuencia, y en su articulación, estos materiales trazan un camino posible para trabajar saberes propios del campo de conocimiento de las ciencias de la computación / informática con estudiantes de salas de 3, 4 y 5 años. Sin embargo, no se busca posicionar al hardware computacional en el centro de la escena. En muchos casos se desarrollan una serie de actividades desenchufadas que buscaban acercarse a los conceptos desde situaciones que inviten en primer lugar, a trabajar con lo conocido. Es decir, se optó por darle mayor relevancia a trabajar y aprender sobre conceptos informáticos de base. El carácter abierto de la propuesta incentiva la apropiación por parte de cada docente, de manera tal que le permita transformarla, recontextualizarla e incluso reescribirla en función de las características de su grupo y de las instituciones donde se desempeñe (Anónimo, 2022).

Las bases metodológicas de las secuencias buscan que los y las estudiantes se involucren en el desarrollo de las actividades propuestas, desde una mirada constructorista de los conocimientos, tal como enfatizan Kafai y Resnick citados en Brennan y Resnick (2012), quienes resaltan la importancia de que los jóvenes se comprometan o involucren.

El conjunto de secuencias y los temas que abordan se resumen en esta tabla:

Nombre de la secuencia	Contenidos y habilidades involucradas	Link
Computadora, ¿sos vos?	Hardware y software	<a href="https://hacemosescuela.cba.gov.ar/computadorasos-vos/">https://hacemosescuela.cba.gov.ar/computadorasos-vos/</a>
El celular: una computadora de bolsillo	Hardware y software Redes y telecomunicaciones Redes sociales Editores multimedia Navegador web  Resolución de problemas	<a href="https://hacemosescuela.cba.gov.ar/el-celular-una-computadora-de-bolsillo/">https://hacemosescuela.cba.gov.ar/el-celular-una-computadora-de-bolsillo/</a>

	con dispositivos.	
¡Manos a la tablet!	Hardware y software Instalación de programas Editores multimedia  Tipo y formato de archivo Creación de contenido digital multimedia	<a href="https://hacemosescuela.cba.gov.ar/manos-a-la-tablet/">https://hacemosescuela.cba.gov.ar/manos-a-la-tablet/</a>
Arrancamos con la Robotita	Hardware y software Robótica Programación Algoritmo  Programa  Instrucción o comando Pensamiento algorítmico Testeo o depuración	<a href="https://hacemosescuela.cba.gov.ar/arrancamos-con-robotita/">https://hacemosescuela.cba.gov.ar/arrancamos-con-robotita/</a>
En la red, pero sin enredarme (Parte I)	Seguridad y privacidad Redes y telecomunicaciones  Hardware y software	<a href="https://hacemosescuela.cba.gov.ar/en-la-red-pero-sin-enredarme-parte-i/">https://hacemosescuela.cba.gov.ar/en-la-red-pero-sin-enredarme-parte-i/</a>
En la red, pero sin enredarme (Parte II)	Redes y telecomunicaciones  Hardware y software	<a href="https://hacemosescuela.cba.gov.ar/en-la-red-pero-sin-enredarme-parte-ii/">https://hacemosescuela.cba.gov.ar/en-la-red-pero-sin-enredarme-parte-ii/</a>

### Docentes destinatarias/os

Al momento de escribir esta colección, gran parte de las decisiones y de los desafíos se centraron en el modo de dirigirse a las y los docentes del nivel, considerando que son quienes llevan la mediación definitiva entre esta propuesta y lo que sucede en las salas.

Es así que concluimos que deberíamos, por una parte, dirigirnos con rigurosidad disciplinar en cuanto a los temas relacionados a la informática, pero con espacios dentro del texto que ofrezcan explicaciones conceptuales sobre la disciplina que sean aprovechadas como sustento

para guiar las actividades propuestas. Esta búsqueda tuvo como horizonte generar experiencias significativas donde las/los docentes encontraran las bases disciplinares que dieran sustento a experiencias de aprendizaje sobre conceptos computacionales.

Si bien teníamos conocimiento sobre el hecho de que las y los docentes del nivel no suelen tener una formación disciplinar ni didáctica sobre informática en sus propios trayectos formativos, no contábamos con datos precisos respecto a cuántas/os de ellas/os tenían algún acercamiento previo. Es por esto que, con el fin de orientar a las y los docentes y de hacer evidente el tipo de contenidos que se trabaja en cada secuencia es que se diseñó un dispositivo llamado la “Brújula de la Informática”<sup>6</sup>, disponible en internet. Este recurso tiene un diseño que permite la interactividad, por lo que quien la navega puede ir adentrándose en los detalles a partir de tres grandes agrupamientos o categorías:

- Sistemas informáticos
- Programación
- TIC

Esta organización de temas corresponde a la necesidad de clarificar aquellos aspectos que suelen confundirse por la tradición escolar de abordaje de las tecnologías. Se apostó a distinguir entre TIC y Ciencias de la Computación, contribuyendo a que cada docente defina en qué momento hace hincapié en cada uno de ellos y a evitar las confusiones.

Las tres categorías se dividen internamente en tres subtemas:

- **Conceptos**, contenidos o temas.
- **Habilidades**, prácticas que son propias del desarrollo del tema.
- **Herramientas, programas o aplicaciones** con los que trabajar los temas y desarrollar las habilidades.

Este recurso es entonces, es el resultado de una de las estrategias que se diseñaron para poder “mapear” el campo disciplinar escolar y reconocer puntos de encuentro entre sus distintas aristas.

### **Decisiones pedagógicas para el abordaje de los temas**

En este intento de acercar las primeras nociones de computación a los y las estudiantes del Nivel Inicial, se tomaron decisiones relevantes en relación a qué temas incluir en el marco de

---

<sup>6</sup> <https://view.genial.ly/62b4cfd18f132001a5200b1>

algunos de los acuerdos generados, tales como:

- Trabajar con el hardware existente en las instituciones públicas de Nivel Inicial (robot, tabletas, internet, notebook y proyector).
- Incluir actividades desenchufadas.
- Comenzar desde lo concreto para pasar a lo abstracto. Tratando de hacer evidente este movimiento a las y los docentes.
- Enfoque construccionista, que aliente a la participación computacional y el aprendizaje colaborativo. Según Kafai (2016) recuperado en Martínez, et al., (2022):

[...] el enfoque de la participación computacional diferencia entre pensamiento y participación (Burke, Orian y Kafai, 2015). Pensar computacionalmente un problema o proyecto significa pensar en el aporte que puede hacer el cómputo, el procesamiento de datos o la automatización de tareas al mismo. La participación computacional –en cambio– implica crear aplicaciones, formar parte de comunidades de software, remixar, recombinar, re-mezclar código y usar múltiples interfaces e instrumentos para abordar un problema computacional (Kafai, 2016) (p. 7).

Teniendo en cuenta estas decisiones se eligieron cinco temas generales (la computadora, el celular, la tableta, el robot educativo y la red de internet) que buscaron trabajar con fundamentos de computación. Cada uno de estos temas se relaciona con los dispositivos computacionales disponibles en las salas y con la propuesta de categorización presentes en la brújula de la informática.

### **Formato de las secuencias**

Las secuencias didácticas producidas buscan dar respuesta, por un lado, a la necesidad de que los y las estudiantes comiencen a conocer fundamentos de computación y, por otro, a la necesidad de formación docente para quienes no obtuvieron en sus estudios de base, un acercamiento a la computación y su didáctica. Es por esto que se generó un trabajo dividido en tres capas, las cuales se diferencian en el texto a partir del diseño gráfico y el registro de escritura. El propósito de este formato de escritura es que las y los docentes cuenten con lo necesario para guiar y coordinar las actividades propuestas sin que el conocimiento disciplinar sea una barrera limitante.

- **Primera capa**

Su redacción apunta a hablar de manera directa a los y las estudiantes. Es decir está pensada para ser reproducida de un modo más o menos textual. Esta decisión se tomó pensando en presentar una manera práctica de implementar y llevar al aula las secuencias. Se alienta a que las/los docentes lo tomen como una pauta general a la cual puedan realizar aportes o modificaciones.

- **Segunda capa**

Se trata de las orientaciones docentes que comunican a las y los docentes la pertinencia de los contenidos de computación propuestos y la relación de éstos con las actividades y los aprendizajes esperados.

En esta capa se trató darle sentido a los contenidos y a las actividades planteadas, proponer alternativas y sugerencias didácticas con el fin de ampliar las herramientas de quien por primera vez aborda contenidos de Ciencias de la Computación en su sala. Hubo una búsqueda didáctica para equilibrar simpleza en lo propuesto con rigurosidad conceptual.

- **Tercera capa**

Esta capa se diferencia estructuralmente de las anteriores al tratarse de anexos a los cuales se puede acceder por medio de hipervínculos presentes en el texto principal.

El contenido de esta capa tiene directa relación con los saberes disciplinares de la computación y tratan de ofrecer un piso mínimo de conocimientos al respecto. Es decir, se explican brevemente aquellas bases computacionales que sustentan las actividades propuestas. Se trata de un espacio de apoyo para las y los docentes, curado por nuestro equipo en busca de información relevante y precisa.

En la siguiente imagen se puede apreciar en tipografía negra, la primera capa; en tipografía azul, la segunda capa; y, al final un enlace que apunta a un archivo de tercera capa con orientaciones conceptuales sobre el tema.

## Relación entre robots y computadoras

*Esta actividad del momento 1 inicia en el lugar donde se dispusieron los afiches con el registro de las secuencias de la colección ya implementadas, en particular, los afiches de la primera secuencia, con las imágenes de computadoras y máquinas.*

*En caso de que no se hayan realizado, será preciso hacer una adaptación y trabajar con el grupo clase la idea de que las computadoras son un tipo de máquina particular que puede adoptar distintas formas (PC, notebook, tablet, celular) y realizar muchísimas tareas diversas. La conversación debe orientarse hacia una de las características centrales de los robots: la presencia de una computadora en su interior. Si los niños y niñas no pueden recuperar esa relación, el o la docente colaborará con preguntas y orientaciones para que puedan aproximarse a ella.*

Por lo que pudimos ver, hay robots para hacer un montón de cosas diferentes. ¿Por qué será que pueden hacer tantas cosas sin que una persona los “maneje”?

El otro día estuvimos conversando de unas máquinas muy especiales que también podían usarse para muchas cosas distintas. ¿Se acuerdan de cuáles eran? Repasemos juntos todo lo que trabajamos esa vez y seguro que se van a acordar.

¡Claro! ¡Las computadoras! Todos esos robots que vimos tienen una computadora dentro. Algunas más grandes y otras más pequeñas. Esas computadoras dan las instrucciones y hacen funcionar al robot sin que tengamos que estar todo el tiempo controlándolo. Nos permiten decirle una vez qué hacer, y que el robot pueda recordarlo y hacerlo.



### **Las computadoras: componentes y funciones**

Para saber más, hagan clic [aquí](#)



En resumen, las y los docentes cuentan en cada secuencia didáctica con las consignas que pueden ser usadas de manera más o menos textuales para el desarrollo de las actividades; luego, orientaciones donde van a encontrar el sentido de realizar estas tareas y una última capa donde encontrarán contenidos disciplinares de computación a modo de base conceptual para sostener el proceso de enseñanza y aprendizaje.

### **Proyecciones**

Tras la publicación de la primera colección de Comarca Digital, el gobierno de la provincia de Córdoba, por medio de su Ministerio de Educación publicó, al inicio del año escolar 2023, una actualización curricular para todo su sistema escolar. Desde el Nivel Inicial hasta el Nivel Secundario. El texto que lleva como título “Aportes de la Cultura Digital en la Educación Tecnológica” incluye contenidos de computación dentro de la currícula. En particular para el Nivel Inicial, estos contenidos se tratan dentro del espacio “Ciencias Sociales, Ciencias Naturales y Tecnología”.

Esta definición abre nuevas posibilidades para continuar con una próxima colección. Entre ellas, la escritura de secuencias didácticas que sigan trabajando fundamentos de las ciencias de la computación, más específicamente enfocadas al aprendizaje de conceptos de programación. Antecedentes como el desarrollo de entornos de programación en bloques específicos para las edades del nivel inicial, como Scratch Jr., donde se destaca el trabajo de la científica argentina Marina Umaschi Bers, abren puertas para que las y los estudiantes se acerquen a los fundamentos de la disciplina de un modo práctico. La utilización de juegos, una metodología constructorista y la integración con otros espacios curriculares seguirán marcando el horizonte para la continuidad de La Comarca Digital.

Se espera también, para la continuidad de la propuesta, contar con devoluciones de docentes que estén implementando estas propuestas en sus salas. Al momento de presentar este artículo se ha diseñado un encuesta y se están recibiendo y procesando los datos resultantes de consultar a maestras que se desempeñan en el nivel y están cursando distintas capacitaciones afines.

### **Bibliografía**

Brennan, K., & Resnick, M. (2012). Entrevistas basadas en artefactos para estudiar el desarrollo del Pensamiento Computacional (PC) en el diseño de medios interactivos. Documento presentado en el encuentro anual de la “American Educational Research Association”, AERA 2012, Vancouver, BC, Canada.

<https://eduteka.icesi.edu.co/pdfdir/EvaluarPensamientoComputacional.pdf>

Dabbah J., Garzón M., Gómez M., Martínez M.C., Martínez López P. E. (2023). Propuesta

- curricular para la inclusión de las Ciencias de la Computación en la educación obligatoria de Argentina. Fundación Sadosky, Buenos Aires.  
[https://curriculum.program.ar/wp-content/uploads/2023/05/2023-05-Program.ar\\_Propuesta-Curricular-para-la-inclusion-de-las-Ciencias-de-la-Computacion\\_v03\\_2023.pdf](https://curriculum.program.ar/wp-content/uploads/2023/05/2023-05-Program.ar_Propuesta-Curricular-para-la-inclusion-de-las-Ciencias-de-la-Computacion_v03_2023.pdf)
- K-12 Computer Science Framework Steering Committee (2016). K12 Computer Science Framework. Technical Report. Association for Computing Machinery. New York, NY, USA.
- Martínez, C., E, P., Gómez, M. J., Borchardt, M., & Garzón, M. (2022). Hacia un currículum emancipador de las Ciencias de la Computación. *Revista Latinoamericana de Economía Y Sociedad Digital*, 3.  
<https://doi.org/10.53857/LBUS5649>
- Ministerio de Educación de la provincia de Córdoba. (2023). Aportes de la Cultura Digital en la Educación Tecnológica.  
<https://www.igualdadycalidadcba.gov.ar/SIPEC-CBA/publicaciones/DyPCurriculares/DAC/2023/aportes-de-cultura-digital-en-la-edu-tecno-ac.pdf>
- Ministerio de Educación de la provincia de Córdoba. (2022). Plan Leer, escribir y pensar en la Cultura Digital.  
<https://www.igualdadycalidadcba.gov.ar/SIPEC-CBA/Programas/2022/culturaDigital.php>
- Ministerio de Educación de la provincia de Córdoba. Secretaría de Educación. Subsecretaría de Promoción de Igualdad y Calidad Educativa. Dirección General de Planeamiento e Información Educativa. (2011). Diseño Curricular de la Educación Inicial. 2011-2020.  
[https://www.igualdadycalidadcba.gov.ar/SIPEC-CBA/publicaciones/EducacionInicial/DCJ\\_Inicial-23-02-2018.pdf](https://www.igualdadycalidadcba.gov.ar/SIPEC-CBA/publicaciones/EducacionInicial/DCJ_Inicial-23-02-2018.pdf)
- Ministerio de Educación y Formación Profesional. (2022). BOE núm. 28, Real Decreto 95/2022. España.  
<https://www.boe.es/eli/es/rd/2022/02/01/95/con>
- Municipalidad de Córdoba, Secretaría de Educación. (2020). Propuesta curricular de educación digital, programación y robótica para Educación Inicial y Primaria del Sistema Educativo Municipal. Documento 3.  
<https://documentos.cordoba.gob.ar/MUNCBA/AreasGob/Edu/DOCS/Seguimos%20con%20vos%20aprendiendo%20en%20casa/Documentos%20de%20apoyo/edipro.pdf>
- Torres, M.I. (2021). Aportes para una apropiación crítica de conocimientos y usos de hardware y software de programación y robótica en la educación para la primera infancia de Argentina. Universidad Nacional de Córdoba. Centro de Estudios Avanzados. Maestría en Tecnología, Políticas y Culturas.  
<https://rdu.unc.edu.ar/handle/11086/23970>
- Violante R., Soto C. A. (2011). Didáctica de la Educación Inicial: Los Pilares. Foro para la Educación Inicial Encuentro Regional Sur. Políticas de enseñanza y definiciones curriculares.  
<https://isfdyt24-bue.infed.edu.ar/sitio/wp-content/uploads/2018/09/2011Conferencia-So>

[to-Vi olante-Did.Ed.-Inicial-PILARES-MCyE.pdf](#)

UNICEF (2022). Las plataformas digitales educativas antes y después del contexto de pandemia por COVID-19. Logros, aprendizajes y desafíos. Serie: Generación Única. Buenos Aires.

<https://www.unicef.org/argentina/media/17116/file/Las%20plataformas%20digitales%20educativas%20antes%20y%20despu%C3%A9s%20del%20contexto%20de%20pandemia%20por%20COVID-19..pdf>

## **Enseñanza de la Inteligencia Artificial en la Escuela Secundaria. Un enfoque metodológico en acción**

Sol Bertochi<sup>1</sup>, Jorge Navarro<sup>1</sup>, Jorge P. Rodríguez<sup>2</sup>, Laura A. Cecchi<sup>2</sup>  
solbertochi92@gmail.com, jorge.navarro@est.fi.uncoma.edu.ar,  
{j.rodrig, lcecchi}@fi.uncoma.edu.ar

<sup>1</sup> Consejo Provincial de Educación  
Ministerio de Educación de la Provincia de Neuquén

<sup>2</sup> Grupo de Investigación en Lenguajes e Inteligencia Artificial  
Facultad de Informática  
Universidad Nacional del Comahue

### **Resumen**

Actualmente, existe consenso global sobre la importancia de incluir conceptos de Inteligencia Artificial (IA) en la educación secundaria, pero no hay acuerdo en los contenidos ni la metodología.

Algunos trabajos consideran de interés, que el conocimiento de la IA no sea enseñado solo desde el enfoque disciplinar, sino que se trabaje en forma transdisciplinar, considerando que en la vida cotidiana los problemas son transversales a diferentes dominios y requieren de varias disciplinas para poder resolverse. Particularmente, se valora la intervención de las Ciencias Sociales, para analizar el impacto que está teniendo la IA y para estudiar las implicancias éticas, sociales y legales de la misma.

En este trabajo, se expone un enfoque didáctico transdisciplinario que ya ha sido presentado y compartido en TEyET 2021. El enfoque cuya propuesta metodológica atraviesa diversas áreas disciplinares y confluye en la enseñanza de la IA desde diferentes dimensiones, está destinado a estudiantes de la escuela secundaria.

Asimismo, se describe un trabajo de campo, que ya fue presentado en TEyET 2021, realizado en una escuela secundaria de la región y que aborda el tema bajo el enfoque didáctico propuesto. Se describe la experiencia, detallando los dispositivos tecnológicos utilizados, las actividades realizadas y los artefactos digitales construidos a partir de ellas.

En esta implementación participaron en forma conjunta y colaborativa docentes de los espacios curriculares destinados a la enseñanza de la Informática y las Ciencias Sociales. Desde las Ciencias de la Computación se abordaron los siguientes temas: conceptos fundamentales sobre IA, incluyendo noción de agentes inteligentes y Test de Turing, entre otros; machine learning, particularmente aprendizaje automático supervisado y Programación Lógica, diseñando una base de conocimiento y razonando sobre ella a través del lenguaje de programación Prolog. Desde las Ciencias Sociales, se desarrollaron conceptos de ética en el uso de la IA en forma sincronizada y paralela al abordaje de los temas dictados en la materia Informática. Se trabajó sobre la red social Youtube, como receptora de datos que son almacenados y administrados por ella, analizando la privacidad de esos datos. A partir de esto, se consideraron aspectos relacionados a las predicciones que se podrían hacer sobre esos datos recopilados.

Las actividades llevaron a los estudiantes a cuestionar la responsabilidad social en la recopilación de datos y en la predicción de conductas mediante IA, ayudándoles a construir criterios y a tomar posición ante situaciones de conflicto e intereses derivados de estas tecnologías.

**Palabras claves:** Educación en Ciencias de la Computación, Escuela Secundaria, Inteligencia Artificial, Ciencias Sociales, Enfoque Transdisciplinar.

## Introducción

Los vehículos auto-conducidos o autónomos, los chatbots que mantienen una conversación con humanos, los robots que colaboran en tareas cotidianas y aquellos que viajan al espacio, la posibilidad de predecir fraudes o el comportamiento de un individuo y dar recomendaciones personalizadas y la tendencia hacia las ciudades inteligentes, ya no son elementos de ciencia ficción.

Así, podemos afirmar que la Inteligencia Artificial (de ahora en más IA) ya forma parte de la vida cotidiana de los sujetos, que desde temprana edad han sido socializados en este área, utilizando videojuegos, asistentes virtuales, Smart TV y teléfonos inteligentes, entre otros.

En consecuencia, esta interacción de los humanos con la IA ha impactado en la forma en que los sujetos se comportan, lo que ha llevado al advenimiento de nuevos efectos en la economía y en las cuestiones sociales, morales y éticas.

Este contexto parece ser el génesis de una nueva era en la historia de la humanidad, donde los humanos deben interactuar con la IA. Esto supone buscar nuevas herramientas para que la ciudadanía piense y actúe críticamente.

Las 10 tendencias tecnológicas estratégicas del 2020 de la consultora Gartner<sup>1</sup> están estructuradas sobre la idea de espacios inteligentes centrados en la persona, lo que significa considerar cómo las tecnologías afectan a las personas (clientes, empleados, etc.) y a los lugares donde viven o

---

<sup>1</sup> <https://www.gartner.com/en/publications/top-tech-trends-2020>

trabajan (sus hogares, oficinas, autos, etc.). Este enfoque direcciona la innovación, considerando a la robótica autónoma y a la IA, particularmente al campo del Aprendizaje Automatizado (Machine Learning), no sólo hacia el enfoque tecnológico sino también hacia el social.

La educación de nuestros futuros ciudadanos debería tomar en cuenta estas dimensiones. Así, el trabajo transdisciplinar y colaborativo entre áreas permitirá una formación holística de nuestros estudiantes.

Si bien hay consenso en la importancia de incorporar el campo de IA al diseño curricular del nivel secundario, no hay un acuerdo en qué contenidos deberían ser incluidos ni tampoco en la metodología a seguir. No obstante, algunos trabajos consideran de interés, que el conocimiento de la IA no sea enseñado solo desde el enfoque disciplinar, sino que se trabaje en forma transdisciplinar, considerando que en la vida cotidiana los problemas son transversales a diferentes dominios y requieren de varias disciplinas para poder resolverse (AAAI, 2018).

Particularmente, se valora la intervención de las Ciencias Sociales, para analizar el impacto que está teniendo la IA y estudiar las implicancias éticas, sociales y legales de la misma (UNESCO, 2019). Por otra parte, se desea que la educación busque desarrollar y anticipar habilidades que requerirán los futuros ciudadanos para vivir y trabajar en la era de la IA (UNESCO, 2019).

De manera reciente, los países de América Latina han incorporado la enseñanza de las Ciencias de la Computación en sus sistemas educativos (Borchardt y Roggi, 2017). Particularmente, en Argentina se establecieron los Núcleos de Aprendizaje Prioritarios (NAP) de Educación Digital, Programación y Robótica (C.F.E., 2018) dirigidos a toda la educación obligatoria para promover la formación de ciudadanos activos, capaces de entender y hacer un uso crítico de las tecnologías digitales. Asimismo, y a comienzos de ese mismo año, Argentina lanzó el Plan Aprender Conectados (P.E.N., 2018) , una propuesta pedagógica integral que busca garantizar la alfabetización digital para el aprendizaje de competencias y saberes necesarios para la integración en la cultura y la sociedad del futuro. La UNESCO en el año 2019, ``destacó a la Argentina como ejemplo de vanguardia en la formación de estudiantes orientada al desarrollo de IA, a través del plan de alfabetización digital Aprender Conectados que impulsa el Ministerio de Educación, Cultura, Ciencia y Tecnología" <sup>2</sup>(M.E.A., 2018).

En nuestra región y en concordancia con el Plan Aprender Conectados, el Consejo Provincial de Educación de la Provincia de Neuquén presentó en 2018, un nuevo diseño curricular de Educación Secundaria de la Provincia de Neuquén (C.P.E., 2018). Los NAP están incluidos en el diseño curricular, distribuidos tanto en el Ciclo Básico Común e Interciclo, como en el Ciclo Orientado de todas las Modalidades del nivel. Asimismo, y acompañando la implementación del área tecnológica

---

<sup>2</sup> <https://www.argentina.gob.ar/noticias/unesco-reconocio-el-trabajo-argentino-en-materia-de-inteligencia-artificial-y-educacion>

del nuevo diseño, la provincia está llevando a cabo el proyecto Educación Digital Neuquén<sup>3</sup> que colabora con la integración de la robótica, entre otros temas.

Sin embargo, la IA es un área compleja en la que han contribuido a su desarrollo, con ideas, puntos de vista y técnicas, diferentes disciplinas como la Matemática, la Filosofía, la Lingüística y la Neurociencia (Russell, 2010).

Por otra parte, en esta nueva era es incierto cómo se relacionarán las nuevas tareas que serán creadas por la IA, con los trabajos existentes y los que aún no han sido instaurados (Wilson et al., 2017).

Por estas razones, se suma, como un reto del siglo XXI, buscar la integralidad y transversalidad de la tecnología en los diversos ejes del saber. Así, la IA requiere de su estudio no solo desde un enfoque tecnológico, sino también debe incluir otras dimensiones disciplinares.

El estudio de las Ciencias de la Computación y de la IA en particular, podría adquirir una perspectiva más holística y transdisciplinar, no sólo para analizar la realidad actual y futura, sino principalmente para poder intervenir en ella (Cobo, 2019). Y así, tanto alumnos como docentes podrían, desde esta perspectiva, construir un conocimiento colaborativo y compartido.

Todos los países reconocen que la IA es la nueva frontera tecnológica y de competitividad, por lo que se han comenzado a delinear estrategias políticas. La educación es uno de los ejes, siendo la incorporación de la IA al currículum escolar uno de los puntos en estudio. Así, varios países lo han incorporado a sus agendas y a sus presupuestos (Lee, 2018). Por otra parte, organizaciones como la International Society for Technology in Education (ISTE) (ISTE, 2018) han reconocido esta necesidad y han comenzado a trabajar en ello. Asimismo, la Association for the Advancement of Artificial Intelligence (AAAI), la Computer Science Teachers Association (CSTA) e IA4All<sup>4</sup> están trabajando en conjunto para desarrollar pautas para la enseñanza de la IA a estudiantes de las escuelas primarias y secundarias (AAAI, 2018).

Los conceptos básicos y los fundamentos de IA están entre los temas a desarrollar en la escuela secundaria. Sin embargo, el foco hacia donde deberíamos orientar los conocimientos de IA no están aún estipulados, como tampoco los temas a desarrollar ni la metodología a seguir, siendo actualmente, puntos de interés en estudio (Wong et al., 2019).

A fin de cubrir este vacío, en (Wong et al., 2020) se han categorizado en tres dimensiones el alcance de la IA en la educación primaria y secundaria: conceptos, aplicaciones y seguridad y ética en IA. En este mismo sentido, AI4K12.org propone *Five Big Ideas in AI*<sup>5</sup> donde una de ellas es el *Impacto Social de la IA* marcando las cuestiones sociales positivas y negativas de su uso. Algunos

---

<sup>3</sup> <http://educaciondigital.neuquen.gov.ar/>

<sup>4</sup> <http://ai-4-all.org/>

<sup>5</sup> <https://github.com/touretzkyds/ai4k12/wiki>

trabajos (Sabuncuoglu, 2020; Touretzky et al., 2019; Wong et al., 2020) han propuesto actividades a desarrollar para la enseñanza de IA, aunque pocos han propuesto una conexión con otras disciplinas para su estudio.

Al incorporar la IA a la educación secundaria, debemos desarrollar habilidades para que el estudiante comprenda de qué se trata el área desde el punto de vista tecnológico, anticipando las capacidades técnicas que necesitará para el trabajo en la Era de la IA. Pero también deberemos enfocarnos en despertar su interés en analizar las consecuencias y potencialidades de su uso, a fin de formar sujetos críticos, capaces de comprender y tomar decisiones en el mundo del futuro. En esta dirección, las Ciencias Sociales pueden integrarse con otras ciencias para reforzar las explicaciones desde una lógica social.

La formación holística de nuestros futuros ciudadanos involucra el entendimiento de la IA, de tal modo que sea posible utilizarla en la vida cotidiana conociendo los riesgos y las ventajas que presenta y que puedan integrarla a sus trabajos en diferentes dominios de modo ético y seguro. El estudio de la IA en la escuela secundaria, en el que en forma transversal y complementaria a los contenidos disciplinares técnicos, las Ciencias Sociales contribuyan en el análisis y reflexión, respecto del impacto social de este área de las Ciencias de la Computación, nos permitirá dar un paso en este sentido.

Usar y entender estas tecnologías se han convertido en habilidades necesarias en la formación de los estudiantes a fin de prepararlos para integrar la sociedad del futuro. En este sentido, se plantea como un desafío, el enfrentar los cambios en la educación para empoderar a los futuros ciudadanos, integrando conocimientos de la IA en las propuestas curriculares.

En este trabajo presentamos un enfoque didáctico ya publicado en (Bertochi et al., 2021), destinado a estudiantes del nivel secundario, cuya propuesta metodológica atraviesa diversas áreas disciplinares y confluye en la enseñanza de la IA desde diferentes dimensiones. A fin de lograr una curva de aprendizaje suave, el enfoque se compone de cinco estructuras de conocimiento que indican diferentes intensidades en el estudio de la IA, iniciando con los saberes preelaborados y que en forma incremental alcanza la capacidad de aplicar un concepto en forma concreta.

Asimismo, se presenta un trabajo de campo, publicado en (Bertochi et al., 2021) que fue realizado en la escuela secundaria C.P.E.M. N° 71, de la ciudad de Centenario, Provincia de Neuquén, durante el año 2020 en un contexto de no presencialidad, en el que el enfoque propuesto fue llevado a la práctica. Esta experiencia involucró la colaboración de las áreas de Informática y de Ciencias Sociales de la institución. Se describe la experiencia, siguiendo el enfoque propuesto, detallando los dispositivos tecnológicos utilizados, las actividades realizadas y los artefactos digitales construidos a partir de ellas.

### **Enfoque Didáctico**

Actualmente, se está enfatizando a nivel mundial sobre la importancia de incorporar al diseño

curricular del nivel secundario, conceptos referidos al área de conocimiento IA como agentes inteligentes que modelan el mundo y usan esa representación para razonar sobre él, técnicas de Machine Learning y sistemas que interactúan naturalmente con los humanos.

Si bien la necesidad de aproximar este tipo de conocimiento a la escuela secundaria tiene un amplio consenso, en tanto se ubica como una pieza clave para mejorar las oportunidades de comprender el mundo, la naturaleza de este tipo de conocimiento exige prestar especial atención a las transposiciones didácticas y las estrategias metodológicas que ayuden a que conocer sobre IA resulte un desafío posible para estudiantes de la escuela secundaria.

El cuerpo de conocimiento relacionado con el área IA se caracteriza por ser de tipo abstracto, como por ejemplo estrategias de búsqueda y representación del conocimiento. Por otra parte, suele requerir el dominio de otras áreas de las Ciencias de la Computación.

En términos de la didáctica específica, el desafío es construir pisos bajos y techos altos para la IA, como lo hace la programación basada en bloques para el área Algoritmos y Programación.

Considerando las diferencias que se plantean con las otras áreas y los aspectos curriculares descritos surge la necesidad de proponer un enfoque didáctico definido específicamente para facilitar enseñanzas y aprendizajes en el campo de la IA.

El enfoque didáctico que se propone busca una aproximación al área de forma multidisciplinar, sosteniendo la especificidad disciplinar y al mismo tiempo ofrecer pisos bajos que tiendan a hacer accesible este tipo de conocimiento a una población estudiantil sin formación previa en el área de conocimiento.

Asimismo, el enfoque propuesto busca desarrollar habilidades en espiral; en el centro se ubica en conjunto de conocimiento más próximos al tipo de tareas que la población estudiantil puede resolver al inicio del proceso formativo y a las concepciones elaboradas en forma previa. Cada ciclo, bucle o iteración incorpora progresivamente prácticas y conceptos más complejos, ampliando en forma gradual la zona de desarrollo próximo (Vigotsky, 1979).

Cada ciclo, representa el abordaje del problema de aprender IA desde las perspectivas epistemológicas ofrecidas por cada una de las áreas disciplinares. Así, por ejemplo desde el área Ciencias Sociales se prestará mayor atención a las implicaciones éticas, políticas y sociales, mientras desde el área Computación se trabajará con mayor énfasis en la aproximación a conceptos como representación del conocimiento. El orden en que interviene cada área no es fijado a ninguna prioridad, sino que se secuencian en relación a la organización escolar.

En este modelo de organización de los contenidos, las disciplinas tienen la función de proporcionar los medios e instrumentos para posibilitar la consecución de los objetivos de aprendizaje. En esta dirección, el criterio de estructuración de los contenidos está determinado por las necesidades emergentes en la elaboración de un producto.

Así, las áreas de conocimiento entran en juego en el momento en que el conocimiento disciplinar es necesario para avanzar en la elaboración de un artefacto concreto. Las producciones construidas en este proceso son una expresión de los aprendizajes logrados y ganan complejidad progresivamente al ser interpeladas desde diferentes perspectivas epistemológicas. Desde este punto de vista se considera que la organización de los contenidos tiene forma transdisciplinar.

El enfoque didáctico propuesto supone la construcción sucesiva de estructuras cognitivas de complejidad creciente. La Figura 1 muestra la definición para estas estructuras. Cada una incorpora nuevos elementos que amplían las posibilidades de comprender y construir artefactos que incorporan técnicas y conceptos sobre IA. Esta definición no es absoluta, necesariamente deberá considerar las particularidades del sujeto que aprende y la realidad contextual.

Las áreas que intervienen tienen que ver con la realidad institucional. La cantidad de iteraciones para cubrir una estructura tiene que ver con las particularidades de la población estudiantil. Por otra parte, la cantidad de estructuras a explorar tiene que ver con los sujetos, la docencia, las temporalidades y los objetivos formativos, entre otras.

El enfoque se organiza en estructuras de conocimiento de complejidad creciente, donde la intensidad con que se abordan los conocimientos sobre IA aumenta gradualmente. En el contexto de esta propuesta, se plantea una integración equilibrada de conocimientos, habilidades y actitudes para comprender en forma satisfactoria problemas relacionados con el área de conocimiento.

Se propone componer el enfoque con cinco estructuras de conocimiento, que en conjunto, ofrecen un recorrido amplio y transdisciplinar por el área de conocimiento y al mismo tiempo sostienen una curva de aprendizaje suave. A continuación se presentan:

- Saberes pre-elaborados acerca de IA.
- Conocer sobre IA y percibir la presencia en artefactos de uso cotidiano. Reconocer aspectos sociales y culturales.
- Conocer conceptos fundamentales sobre Inteligencia Artificial. Analizar aspectos éticos, sociales y legales.
- Describir cómo maneja la IA sistemas de software y hardware. Comprender la responsabilidad individual y colectiva.
- Capacidad para utilizar o aplicar un concepto de manera concreta. Comunicar en forma oral y escrita las implicancias de la IA.

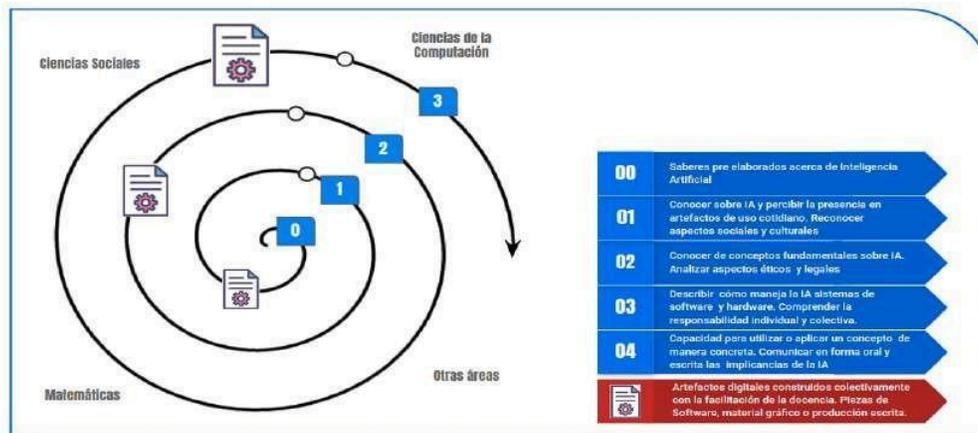


Figura 1: Enfoque didáctico propuesto

En relación a la interacción entre áreas de conocimiento, a simple vista parece razonable adoptar modelos integrados donde las áreas comparten tiempo y espacio. Sin embargo, numerosos estudios muestran que estos modelos resultan de difícil implementación y tienen grandes posibilidades de que algunas áreas tomen lugares preponderantes en detrimento de otras (Fluck et al., 2016).

En este marco, se plantea un abordaje de carácter transdisciplinar que permite tratar la enseñanza de la IA desde diferentes perspectivas y al mismo tiempo sostenga la especificidad de cada área de conocimiento.

En esta dirección, se presta especial atención a la producción colectiva de artefactos digitales como una forma de organizar los procesos de enseñanzas y de aprendizajes, donde los contenidos que se abordan se presentan relacionados en torno al artefacto concreto que se está elaborando. Se trata del desarrollo de piezas de Software, material gráfico o producción escrita, con la facilitación docente desde diferentes áreas de conocimiento.

El desarrollo de artefactos es tratado desde cada perspectiva disciplinar en el tiempo y espacio curricular asignado al área de conocimiento. Así, por ejemplo en el espacio curricular destinado a Ciencias de la Computación se abordará la producción buscando poner en juego conceptos como representación del conocimiento o Machine Learning, mientras la facilitación desde el área Ciencias Sociales prestará mayor atención a explicitar las implicaciones éticas, políticas y sociales. La producción de artefactos tratada desde múltiples perspectivas es una pieza clave de este enfoque; es un modo de organizar los procesos de enseñanzas y aprendizajes que posibilita la confluencia de áreas de conocimiento, evita parcelar el conocimiento en asignaturas aisladas y al mismo tiempo sostiene la especificidad disciplinar.

La Figura 1 muestra como un producto elaborado por estudiantes transita por diferentes espacios curriculares. En ese trayecto incorpora los puntos de vista ofrecidos por las disciplinas e incrementa la complejidad del producto, como también la del cuerpo de conocimiento que se pone en juego durante su desarrollo.

## Trabajo de Campo

El enfoque presentado en este trabajo fue llevado a las aulas de la escuela secundaria

C.P.E.M. N° 71, de la ciudad de Centenario, Provincia de Neuquén, con la intención de ajustarlo progresivamente, a partir de la consideración de comentarios, sugerencias y revisiones realizadas sobre el trabajo de campo.

La experiencia desarrollada consistió en la realización de diferentes actividades, tanto grupales como individuales, todas enmarcadas en un contexto de no presencialidad. En este sentido, se eligió utilizar la Plataforma Educativa Chamilo<sup>6</sup>, que es la adoptada por la institución. Asimismo, diferentes recursos digitales tales como herramientas de videoconferencia, correo electrónico, mensajería instantánea, documentos compartidos y foros de debate fueron utilizados para llevar a cabo la experiencia.

La población estudiantil que participó de esta experiencia se compone por 54 adolescentes que cursan el quinto año de estudios de la educación secundaria. Dicha población se agrupa en dos cursos, no cuenta con conocimientos previos de carácter formal sobre los temas tratados, aunque dispone de un manejo fluido de las tecnologías y mayoritariamente cuenta con la conectividad requerida para participar de actividades académicas.

En relación a la interacción de las áreas de conocimiento, la experiencia se desarrolló en los espacios curriculares asignados a Ciencias Sociales y a Informática. Cada área de conocimiento desarrolló la actividad en los tiempos y espacios propios, con la definición de puntos de encuentro y sincronización destinados a mejorar la articulación.

La propuesta se organizó en cuatro iteraciones, cada una agrega diferentes herramientas nativas de la plataforma educativa, como cuestionarios y lecciones. Asimismo suma recursos externos, como muros colaborativos, vídeos y presentaciones en línea. La utilización de recursos en línea contribuye a que las producciones en desarrollo estén accesibles a toda la docencia otorgando continuidad transdisciplinar: cada asignatura interviene un mismo producto desde la perspectiva disciplinar.

La experiencia se organizó en 4 iteraciones:

- **Iteración 0:** Con intención de recuperar los saberes pre elaborados por la población estudiantil acerca de IA y sus implicaciones sociales, se indaga acerca de sus percepciones en relación a la IA a través de un cuestionario en línea. En el mismo, también se investigó sobre su apreciación respecto del uso de TIC y de su nivel de confianza al depositar en forma voluntaria o involuntaria, en estos artefactos, muchos de sus datos. Los resultados de este cuestionario permitieron ajustar el proyecto, formular y revisar las actividades. Este dispositivo de indagación se

---

<sup>6</sup> <https://chamilo.cpem71.edu.ar>

armó en conjunto entre los docentes del área de Informática y de Ciencias Sociales.

- **Iteración 1:** A fin de que los estudiantes conozcan sobre IA, se realizó una primera actividad que involucró el uso de diferentes recursos didácticos digitales destinados a la presentación de información. De esta forma se ofreció un primer contacto con conceptos fundamentales a partir de presentar el conocimiento de manera sencilla a través de vídeos breves. Por otro lado, como segunda estrategia de aproximación a estos tópicos se utiliza Akinator<sup>7</sup>, una aplicación web que identifica un personaje a partir de preguntas sobre características del mismo. Se convoca a los estudiantes a revisar la experiencia reciente, recuperar conclusiones y conectar con situaciones de la vida real, como las interacciones que se establecen con agentes que realizan recomendaciones personalizadas. Por otra parte, se utilizó Eliza, un chatbot básico que intenta emular a una persona. Esta actividad contribuyó a mejorar la percepción sobre la presencia de IA en artefactos de uso cotidiano, como el Asistente de Google. Para dar continuidad al estudio sobre recursos disponibles en la red, en esta oportunidad enfocando la atención en reconocer aspectos sociales y culturales, se propuso a los estudiantes que piensen atributos que debería tener "la mejor hamburguesa" desde las perspectivas de diferentes actores, como podrían ser el padre, el niño o el fabricante. A partir del análisis de una matriz ética (Payne, 2019), los estudiantes examinan dónde se superponen o entran en conflicto esos valores. Este mecanismo se transpone para estudiar aspectos éticos y sociales de una red social. En este punto del trayecto se busca incorporar al estudio de la IA las perspectivas de análisis aportadas por las Ciencias Sociales.

- **Iteración 2:** Esta iteración tiene el propósito principal de conectar las experiencias concretas compartidas en la Iteración 1 con conceptos fundamentales sobre Inteligencia Artificial. En esta oportunidad se revisa la experiencia de juego con Akinator centrando la atención en intentar comprender cómo la aplicación logra identificar un personaje entre miles de opciones. En el marco del proceso reflexivo se conecta la experiencia reciente con conceptos fundamentales sobre Machine Learning y representación del conocimiento. Se plantea una actividad de carácter retrospectiva sobre los diálogos establecidos con el chatbot Eliza<sup>8</sup>, para avanzar en la construcción de la definición de la IA a partir del Test de Turing (Russell, 2010). En el paso por el área de las Ciencias Sociales, se continuó el trabajo a partir de la matriz ética construida considerando una red social. En este punto se analizaron los aspectos éticos, sociales y legales, respecto de las entradas a la matriz. Particularmente, se distinguieron los puntos que son conflictivos entre los diferentes actores, como privacidad de datos para el usuario y recomendaciones personalizadas para los que publicitan sus productos.

- **Iteración 3:** La primera actividad de esta iteración estuvo a cargo del área de Informática y se enfocó en el estudio de la representación del conocimiento y el razonamiento y de Machine Learning. En este sentido, se trabajó sobre el juego de mesa ¿Quién es quién? identificando entre

---

<sup>7</sup> <https://es.akinator.com/>

<sup>8</sup> <http://deixilabs.com/eliza.html>

los personajes históricos características comunes. Los datos obtenidos se volcaron en un muro colaborativo para que cada alumno pueda agregar las características reconocidas. A partir de estos datos se generó una Base de Conocimiento. Para trabajar sobre la representación del conocimiento y razonamiento, se utilizó el lenguaje de Programación Lógica Prolog, particularmente SWI-Prolog<sup>9</sup>. Los estudiantes accedieron a la Base de Conocimiento generada, realizaron programas declarativos simples, construyeron consultas y las ejecutaron. Respecto de Machine Learning se explicó la técnica de aprendizaje supervisado sobre la misma Base de Conocimiento, introduciendo la noción de atributo y de árboles de decisión. Se remarcó la importancia de la cantidad y calidad de los datos de entrada al algoritmo y se enfatizó la predicción generada a partir de ellos.

## Conclusión

En este trabajo se presentó un enfoque didáctico diseñado específicamente para hacer accesible una amplia gama de tópicos del área IA a estudiantes secundarios sin formación previa en el área de conocimiento.

Los resultados obtenidos a partir de revisar la experiencia educativa permiten confirmar las principales opciones didácticas adoptadas al momento de la formulación del enfoque y aportan información para desarrollar ajustes progresivos. La vuelta a la presencialidad plena y los ajustes graduales introducidos debido a consideraciones institucionales, obstaculizaron la realización de nuevas experiencias de campo. La intención es llevar a cabo estas actividades nuevamente en el año 2024. Asimismo, se planea diseñar instrumentos de evaluación y utilizarlos de manera integral para evaluar el impacto de las nuevas experiencias programadas.

Disponer de este tipo de construcciones teóricas se ubica como pieza clave para el mejoramiento de las prácticas educativas, que tienen intención de desarrollar las habilidades necesarias para participar activamente de la sociedad del presente y del futuro que incluye tareas aún desconocidas y fuertemente influenciadas por la IA.

Abordar estos temas en toda su complejidad requiere de la participación transdisciplinaria de diferentes sistemas de contenidos. En este sentido el aprendizaje y trabajo colaborativo soportados por computadoras ocupan un lugar central.

## Referencias

AAAI. (2018). *Association for the Advancement of Artificial Intelligence—AAAI Launches “AI for K-12” Initiative in Collaboration with the Computer Science Teachers Association (CSTA) and AI4All.*

Argentina. (2018). *Agenda Digital Argentina 2030—Boletín Oficial Decreto 996/2018—Gobierno de la República Argentina.*  
<https://www.caserosada.gob.ar/informacion/actividad-oficial/9-noticias/44081-el-gobierno-presento-la-nueva-agenda-digital-2030>

---

<sup>9</sup> <https://www.swi-prolog.org/>

- Bertochi, S., Navarro, J., Rodriguez, J., & Cecchi, L. A. (2021, La Plata). Enfoque didáctico para la enseñanza transdisciplinar de la inteligencia artificial. *XVI Congreso de Tecnología en Educación & Educación en Tecnología. TE&ET 2021*.
- Cobo, C. (2019). *Acepto las Condiciones: Usos y abusos de las tecnologías digitales*. Fundación Santillana.
- C.P.E. (2018). *Consejo Provincial de Educación de la Provincia de Neuquén—Diseño Curricular Jurisdiccional de los tres primeros años de la Escuela Secundaria Neuquina Resolución N° 1463/18*.  
<https://educaciondigital.neuquen.gov.ar/normativa-y-documentos-legales/#:~:text=Resoluci%C3%B3n%20CPE%201463%2F2018%20%E2%80%93%20APROBAR,Anexo%20%C3%9Anico%20de%20la%20presente>.
- C.F.E (2018). *Consejo Federal de Educación - Educación Digital, Programación y Robótica (NAP) Resolución C.F.E. N° 343/18*.
- Fluck, A., Webb, M., Cox, M., Angeli, C., Malyn-Smith, J., Voogt, J., & Zagami, J. (2016). Arguing for computer science in the school curriculum. *Journal of educational technology & society*, 19(3), 38-46.
- ISTE. (2018). *International Society for Technology in Education. Bold New Program Helps Teachers and Students Explore the Power of AI [Press release*.
- Lee, K.-F. (2018). *AI superpowers: China, Silicon Valley, and the new world order*. Houghton Mifflin.
- M.E.A. (2018). *Ministerio de Educación de la República Argentina—UNESCO reconoció el trabajo argentino en materia de inteligencia artificial y educación*.  
<https://www.argentina.gob.ar/noticias/unesco-reconocio-el-trabajo-argentino-en-materia-de-inteligencia-artificial-y-educacion>
- Payne, B. H. (2019). An ethics of artificial intelligence curriculum for middle school students. *MIT Media Lab Personal Robots Group*. Retrieved Oct, 10, 2019.
- P.E.N. (2018). Poder Ejecutivo Nacional, Presidencia de la Nación, Plan aprender conectados—Decreto 386/2018. *Boletín Nacional*.
- Russell, S. J. (2010). *Artificial intelligence a modern approach*. Pearson Education, Inc.
- Sabuncuoglu, A. (2020). Designing one year curriculum to teach artificial intelligence for middle school. *Proceedings of the 2020 ACM conference on innovation and technology in computer science education*, 96-102.
- Touretzky, D., Gardner-McCune, C., Martin, F., & Seehorn, D. (2019). Envisioning AI for K- 12: What should every child know about AI? *Proceedings of the AAAI conference on artificial intelligence*, 33(01), 9795-9799.
- UNESCO. (2019). *International Conference on Artificial Intelligence and Education. Final Report. Planning Education in the AI Era: Lead the leap*.  
<https://unesdoc.unesco.org/ark:/48223/pf0000370967>
- Vigosky, L. (1979). *El desarrollo de los procesos psicológicos superiores*. Edición crítica. Barcelona.

- Wilson, H. J., Daugherty, P., & Bianzino, N. (2017). The jobs that artificial intelligence will create. *MIT Sloan Management Review*, 58(4), 14.
- Wong, G. K., Ma, X., Dillenbourg, P., & Huan, J. (2020). Broadening artificial intelligence education in K-12: Where to start? *ACM Inroads*, 11(1), 20-29.
- Wong, G. K., Ma, X., & Huen, J. (2019). When schools meet artificial intelligence in Hong Kong. *ACM Inroads*, 10(4), 43-46.

## **Minirobot app: un caso de estudio y puesta en aula**

Laura Felice, Paula Tristán, Fermín de Fiore, Francisco Prieto  
{laufelice, ptristan1, dffermin, priettt}@gmail.com

INTIA. Facultad de Ciencias Exactas  
Universidad Nacional del Centro de la Provincia de Buenos Aires

### **Resumen**

Minirobot app es una aplicación creada como producto de un proyecto con el objetivo de mejorar la usabilidad de los minirobots en la enseñanza de la robótica en el nivel inicial de educación formal. Para llevar a cabo el objetivo principal, se diseñaron tarjetas físicas que representan las instrucciones que el robot puede ejecutar, reemplazando el uso de un editor gráfico por un elemento tangible, con el que los niños pueden crear los programas que luego ejecutará el robot.

En este artículo, se describe el funcionamiento, la arquitectura y un análisis de resultados de la aplicación, como así también la experimentación llevada a cabo en un aula de educación inicial. Se fundamentan las mejoras respecto a las funcionalidades existentes del minirobot, las que lograron optimizar la interacción entre el usuario y el dispositivo, generando nuevas oportunidades de uso y comodidad para los niños.

**Palabras clave:** Robótica Educativa, Minirobot, Educación Inicial, Machine Learning

### **Introducción**

El área de robótica educativa constituye un tema de investigación continua. Está consensuado entre las comunidades de ciencias de la educación y ciencias de la computación, que la programación y la robótica en todos los niveles de la educación formal, fomenta la creatividad, el pensamiento lógico y la precisión en la resolución de problemas.

Minirobot es un pequeño dispositivo que se programa mediante un editor basado en Logo, un lenguaje diseñado para enseñar los conceptos básicos de programación, que permite repasar y reforzar conceptos de geometría al programar la tortuga (el cursor que dibuja en Logo) para que dibuje distintas figuras geométricas. Está impreso en 3D, cuenta con ruedas, un lápiz, luces led y parlantes, instalados sobre un chip Arduino ESP8226 preprogramado que permite hacer uso de cada uno de los recursos mediante instrucciones simples enviadas al robot desde una aplicación web. El dispositivo es parte del proyecto denominado también Minirobot [<http://minirobots.com.ar/>] que busca fomentar el desarrollo de la creatividad en los alumnos, el aprender a aprender, y aprender haciendo desde un punto de vista lúdico.

El proyecto Minirobot app [Prieto, 2023] incluye el desarrollo de una aplicación Android que garantiza el acceso al minirobots mediante la utilización de cualquier dispositivo móvil,

configurándose de manera sencilla y evitando la necesidad de dispositivos adicionales, que podrían desalentar el uso del robot.

Minirobot app se transformó en una aplicación producto de un proyecto abarcador cuyo objetivo es mejorar la usabilidad de los minirobots en la enseñanza de robótica para el nivel inicial. Para llevar a cabo el objetivo principal, se diseñaron tarjetas físicas que representan las instrucciones que el robot puede ejecutar, reemplazando el uso del editor gráfico por un elemento tangible, con el que los niños pueden crear los programas que luego ejecutará el robot. Para lograr este objetivo puntual, se desarrollaron una serie de algoritmos y se aplicaron técnicas de Machine Learning (ML) [Chollet, 2018] para el reconocimiento de las mismas. Adicionalmente, se implementó un mecanismo de traducción de las tarjetas reconocidas a instrucciones entendibles por el robot. Se realizaron pruebas experimentales y la puesta en aula de la aplicación.

Este artículo está organizado de la siguiente manera, en la sección 2 se describe la aplicación, su funcionalidad y la arquitectura de la misma. La sección 3 muestra los resultados de la implementación de la aplicación y en la sección 4 la experimentación realizada en el aula. La sección 5 brinda conclusiones y trabajos futuros.

### **Minirobot App**

El proyecto Minirobots surge de la visión de llevar la programación a todos los hogares, a través de la creación de robots que se controlan mediante algoritmos. Este proyecto de software libre recibió el Fondo Semilla del Ministerio de la Producción, y fue ganador del Premio a la Educación Clarín-Zurich 2018. Los objetivos principales de este proyecto se basan en la búsqueda de la estimulación del desarrollo cognitivo a través de distintos canales de percepción, utilizando recursos lúdicos visuales, auditivos y táctiles.

El prototipo de Minirobots es un pequeño dispositivo que se programa mediante un editor basado en Logo, un lenguaje diseñado para enseñar los conceptos básicos de programación, que permite repasar y reforzar conceptos de geometría al programar la tortuga (el cursor que dibuja en Logo) para que dibuje distintas figuras geométricas. Están impresos en 3D y cuentan con ruedas, un lápiz, luces led y parlantes, instalados sobre un chip Arduino ESP8226 preprogramado que permite hacer uso de cada uno de los recursos mediante instrucciones simples enviadas al robot desde una aplicación web. La Figura 1 ilustra el kit.



Figura 1. Kit de Minirobot

Para poder enmarcar el objetivo de este trabajo es necesario describir la funcionalidad y arquitectura de la aplicación desarrollada. En este artículo se resume tanto como es posible para poder entender la aplicación en su contexto.

### **Funcionalidad de la app**

En [Prieto, 2023] se ha explicado detalladamente las mejoras hechas al robot en cuanto a su hardware. Como consecuencia de estas mejoras, la infraestructura necesaria para conectarse al robot y enviar instrucciones, ha disminuido significativamente. Esto es así, ya que sólo se requiere un dispositivo móvil sin necesidad de una conexión a internet. No obstante, se completó un desafío propuesto al inicio del proyecto de mejora: desarrollar una alternativa que permita generar, editar y enviar programas al robot de manera offline, considerando que los programas serán construidos utilizando fichas físicas. La construcción de las fichas y el propósito de cada una, se encuentra debidamente descrito y justificado en el documento nombrado anteriormente. Para lograr la funcionalidad, se desarrolló la aplicación móvil con dos funcionalidades fundamentales:

- El reconocimiento de las instrucciones en base a una foto de un programa armado con las piezas de rompecabezas.
- El envío del programa al robot para que sea ejecutado.

A continuación, se describe la arquitectura de la aplicación, haciendo hincapié en los diferentes componentes y módulos que la integran para entender no sólo cómo está desarrollada la aplicación sino fundamentar el objetivo de la misma.

### **Arquitectura de la app**

La arquitectura general de la aplicación se divide en tres capas: presentación, dominio e infraestructura (figura 2).

En la capa de presentación, se encuentran los componentes nativos de Android que tienen que ver con la interfaz de usuario, y que definen cómo se muestran los elementos en pantalla. En la capa de dominio, se encuentran casos de uso y entidades, que describen

acciones concretas que la aplicación puede realizar, y clases destinadas a definir los datos necesarios para estas acciones. Así, se genera una interfaz de código que permite cambiar implementaciones de ciertas acciones sin afectar la capa de presentación de manera innecesaria. En la capa de infraestructura, se agrupan los componentes nativos de red, almacenamiento, algoritmos de ML o partes del código complejas que se quieran abstraer de la capa de presentación.

Esta división en capas se basa en el concepto de Clean Architecture [Martin, 2017] y en los principios SOLID, una serie de lineamientos y prácticas formuladas por Robert C. Martin [Martin, 2002] seguidos por una gran cantidad de desarrolladores de Android. Estos principios permiten mejorar la separación de intereses en el código: cada clase tendrá menos motivos para cambiar, menos responsabilidades, será fácil de modificar, de extender y de testear.

Para cada pantalla o funcionalidad de la aplicación se creó un paquete, y dentro de estos paquetes se crearon otros para representar las capas mencionadas anteriormente. Esto permite acceder y visualizar con mayor facilidad el código que lleva a cabo cada funcionalidad.

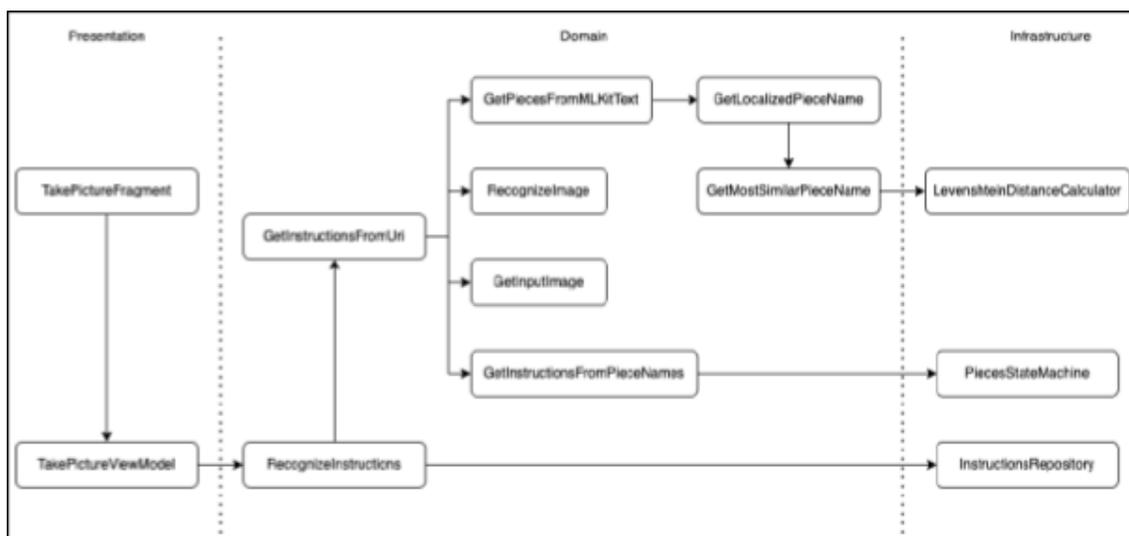


Figura 2. Arquitectura de la app

En la capa de presentación (figura 3), se utiliza la técnica de MVVM (Model, View, ViewModel), que permite estructurar la lógica de presentación de una forma sencilla y escalable. Esta técnica se divide en tres componentes principales: la vista (View), que representa lo que el usuario visualiza en la pantalla, es decir, la interfaz de usuario. Se intenta que este código sea simple, y sólo cambia si el ViewModel lo requiere. El modelo de vista (ViewModel) almacena el estado de la interfaz de usuario, junto con la lógica necesaria para mostrar información en pantalla. Se comunica con la vista para indicar cambios en la información a mostrar. A su vez, el ViewModel actualiza el modelo mediante los casos de uso descritos anteriormente, con los que puede enviar

y recibir datos según las necesidades de cada pantalla. En esta arquitectura, el modelo de datos (Model) corresponde a las capas de dominio e infraestructura. La capa de presentación está estrechamente relacionada con el framework o sistema operativo a utilizar, ya que la visualización en pantalla depende del sistema operativo. Se podría utilizar un mismo lenguaje para definir la capa de dominio de una aplicación Android, iOS o una página web, ya que sólo contendría la lógica de negocio. Sin embargo, esto no sería posible para la capa de presentación, ya que para mostrar la interfaz de usuario se requiere utilizar lenguajes y frameworks específicos de cada plataforma.

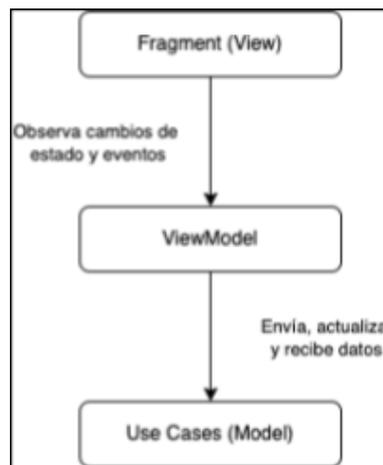


Figura 3. Capa de presentación

En la capa de dominio se encuentra la lógica de negocio, que debe ser, en la medida de lo posible, independiente de las capas de presentación y de infraestructura. Aquí se pueden encontrar casos de uso, que contendrán la lógica necesaria para ejecutar diversas acciones, y serán los encargados de invocar a otros casos de uso, repositorios o elementos de la infraestructura para cumplir su cometido. También se pueden encontrar entidades de dominio, que definirán los datos a utilizar por los casos de uso y luego traducirán a objetos de infraestructura o de presentación, según sea necesario.

En la capa de infraestructura se encuentran las clases que llaman a las bibliotecas que sean necesarias para el almacenamiento de datos, la comunicación con servidores, los algoritmos específicos de procesamiento de datos o de OCR [Rusell, 2020], en el caso de esta implementación. La idea de esta capa es acceder de forma organizada a diversas herramientas que son necesarias para poder llevar a cabo las acciones necesarias por la capa de dominio, pero separando la lógica de modo tal que dicha capa sea agnóstica de la tecnología a utilizarse.

La interfaz de la aplicación, con el detalle de cada una de las pantallas que la componen puede encontrarse en [Prieto, 2023]. La figura 4 resume el flujo general de la aplicación.

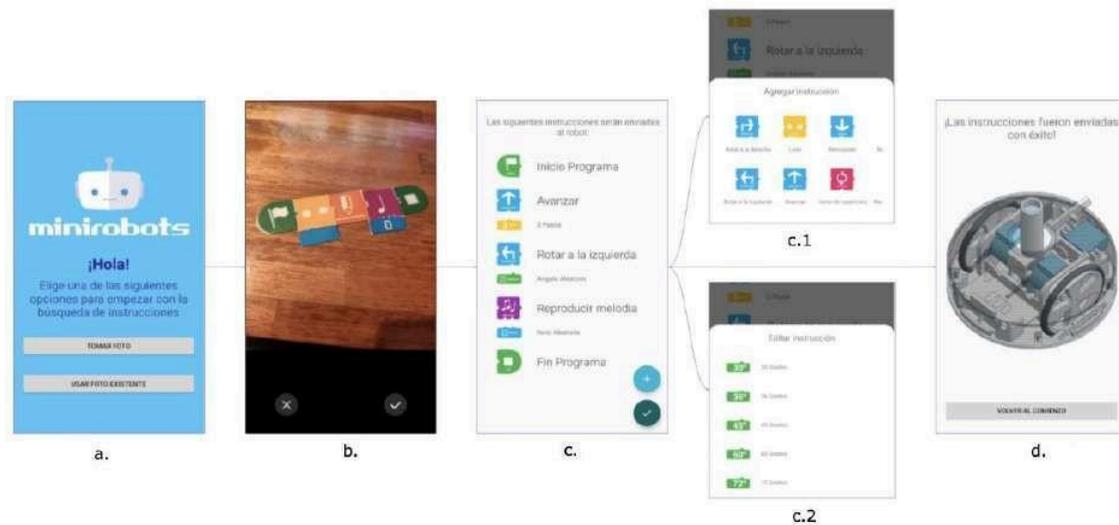


Figura 4. Flujo general de la aplicación. (a) Pantalla de bienvenida. (b) Reconocimiento de instrucciones. (c) Previsualización del programa. (c.1) Agregado de instrucciones. (c.2) Edición de instrucciones. (d) Transmisión del programa al robot.

En la sección siguiente se muestran los resultados del funcionamiento de la aplicación. Estos resultados son los que sirven de feedback para la mejora de la misma, siendo esencial para optimizar tanto la aplicación misma como la experiencia del usuario.

### Resultados del funcionamiento de la app

Para evaluar la efectividad del algoritmo utilizado para el reconocimiento de tarjetas, se diseñó un programa compuesto por siete fichas principales y cuatro fichas modificadoras: Inicio, Derecha (30°) - Abajo - Adelante (6) - Negra (SI) - Izquierda (60°) - Fin, como se muestra en la figura 5.



Figura 5. Programa utilizado para evaluar la eficacia

En esta sección se describe en detalle el proceso y los resultados obtenidos para el programa de la figura. Se tomaron diversas fotografías del programa en diferentes escenarios para obtener resultados más precisos y cercanos a lo que se podría experimentar al usar el robot. Esta tarea fue realizada por un usuario que desconoce el escenario ideal para capturar las imágenes, por lo

tanto, no busca facilitar la tarea de reconocimiento del algoritmo. Durante el procedimiento se tuvieron en cuenta las variaciones que pueden afectar el reconocimiento de las tarjetas, tales como las condiciones de luz, los tipos de fondos, los ángulos distintos, los reflejos, las sombras y las combinaciones de estos factores. Para comenzar, se tomaron diecisiete imágenes del programa, cada una con propiedades diferentes. Se utilizaron dos tipos de fondos distintos, uno liso y oscuro, y otro de madera vetada. Se variaron las luces entre luz cálida y fría, y se aplicaron filtros de luces azules y rojas. En cuanto a los distintos ángulos, se tomaron fotos ubicando el dispositivo móvil de manera paralela a la mesa donde se encontraba el programa y, también, se inclinó el dispositivo con respecto a la mesa para dificultar la identificación. Además, algunas imágenes poseen reflejos de luces en las tarjetas o sombras del dispositivo móvil al momento de sacar la imagen. A simple vista, se observa que estos factores disminuyen la calidad de las imágenes y, por lo tanto, agregan complejidad al proceso de reconocimiento. A continuación, se muestran algunos resultados de las pruebas mencionadas.

### Resultados de las pruebas

En la etapa de prueba, se identificó un alto porcentaje de casos en los que se registraba al menos una falla en la identificación de alguna de las tarjetas de un programa. En vista de estos resultados, se revisó detalladamente el código del algoritmo y se aplicaron cambios en ciclos iterativos, junto con las pruebas, para lograr una mejora en el reconocimiento. Gracias a los resultados obtenidos en el análisis de la eficiencia, se modificó el mecanismo de distancia de Levenshtein [<https://gist.github.com/josejuan/2875084>] para ignorar textos reconocidos con una distancia demasiado corta.

Foto número	Total de tarjetas	Aciertos	Fallos	Tasa de aciertos
1	11	9	2	81,8%
2	11	11	0	100%
3	11	11	0	100%
4	11	10	1	90,91%
5	11	8	3	72,73%
6	11	11	0	100%
7	11	10	1	90,91%

8	11	10	1	90,91%
9	11	10	1	90,91%
10	11	10	1	90,91%
11	11	10	1	90,91%
12	11	11	0	100%
13	11	10	1	90,91%
14	11	11	1	100%
15	11	11	1	100%
16	11	8	3	72,73%
17	11	11	0	100%
<b>Totales</b>	<b>187</b>	<b>170</b>	<b>17</b>	<b>90,91%</b>

Tabla 1. Resultados obtenidos antes de las mejoras aplicadas

En la tabla 1 se muestran los resultados obtenidos antes y después de las mejoras aplicadas. Se analizaron un total de 187 tarjetas de las cuales se reconocieron exitosamente 170 y se obtuvieron fallas en los 17 restantes. De esta forma, el porcentaje de eficiencia de reconocimiento válido de tarjetas fue del 90%. El promedio es de una falla por programa.

Si bien el porcentaje de eficiencia es alto, del total de 17 programas evaluados, se encontraron 11 programas que presentaron al menos una falla. Esto indica que antes del cambio en el algoritmo, el porcentaje de casos en los que se percibió al menos una falla fue de 64%.

Luego de las modificaciones en el algoritmo de reconocimiento, se obtuvieron resultados positivos, logrando reconocer exitosamente un porcentaje mayor de tarjetas. De las 187 utilizadas, 179 fueron reconocidas correctamente, lo que representa un porcentaje de éxito del 95,8%, expresado en tabla 2.

Foto número	Total de tarjetas	Aciertos	Fallos	Tasa de aciertos
1	11	9	2	81,8%

2	11	11	0	100%
3	11	11	0	100%
4	11	10	1	90,9%
5	11	8	3	72,7%
6	11	11	0	100%
7	11	11	0	100%
8	11	11	0	100%
9	11	11	0	100%
10	11	11	0	100%
11	11	11	0	100%
12	11	11	0	100%
13	11	11	0	100%
14	11	11	0	100%
15	11	11	0	100%
16	11	11	0	100%
17	11	11	0	100%
<b>Totales</b>	<b>187</b>	<b>179</b>	<b>8</b>	<b>95,72%</b>

Tabla 2. Resultados obtenidos después de las mejoras aplicadas

Estos resultados indican que las modificaciones aplicadas mejoraron la eficiencia del reconocimiento, permitiendo que de un total de 17 programas analizados, sólo 5 presentan fallas. Al mismo tiempo, el porcentaje de casos en los que se detectó al menos una falla bajó de 64% al 29%. El promedio de fallas por programa se redujo a la mitad.

En conclusión, las mejoras implementadas en el sistema de reconocimiento tuvieron un impacto positivo en su capacidad de detectar correctamente las tarjetas. Se observó un aumento del 5,8% en el porcentaje de éxito, así como una reducción tanto en el número total de fallos como en la cantidad de programas que presentaron al menos una tarjeta

fallida. Todo esto sugiere que las mejoras aplicadas fueron efectivas en aumentar la precisión del sistema.

### **Experimentación**

En esta sección, se muestran los resultados obtenidos de la puesta en aula de la aplicación. Las actividades realizadas para obtener resultados experimentales fueron desarrolladas en distintas etapas en conjunto con personal docente de un jardín de Infantes de la ciudad de Tandil. Los tiempos de feedback necesario para cada etapa fueron más amplios de lo que se había programado en un principio, la tarea de coordinación para el trabajo en conjunto con los docentes depende fuertemente de las actividades y tiempos extra que los equipos disponían.

A pesar de esta dificultad, dado que este trabajo se enmarca en un proyecto de extensión institucional de la Facultad de Ciencias Exactas con el Jardín de Infantes “Dr. R.Zarini” dependiente de la Universidad Nacional del Centro de la Provincia de Buenos Aires, se decidió trabajar para las pruebas iniciales con esta institución agregando valor al proyecto institucional. Dado que los docentes de todas las áreas del establecimiento habían recibido capacitación previa en robótica educativa, el diseño de las secuencias didácticas se desarrolló de manera fluida y no presentó dificultades. Se planificaron un conjunto de actividades teniendo en cuenta experiencias de aprendizaje previas, a fin de que el desarrollo de la actividad fluya de manera progresiva. Estas secuencias didácticas tienen como objetivo principal promover el desarrollo integral de los niños, abordando aspectos cognitivos, emocionales, sociales y físicos. El trabajo realizado pudo ser compartido por todo el personal docente de la institución.

Se trabajó con alumnos de cuatro y cinco años que habían participado en actividades relacionadas a la robótica de forma desconectada, y en algunos casos ya habían experimentado con Bee-Bot [<https://codigo21.educacion.navarra.es/autoaprendizaje/bee-bot-robot-infantil-programable/>].

Los objetivos de las actividades en las que trabajaron anteriormente pueden resumirse, en palabras de la docente, de la siguiente manera:

*“... se privilegia el desarrollo del pensamiento computacional, aunque no sea la intención desde el nivel inicial, ya que se iniciará a los niños y niñas en los contenidos de algoritmo y de programación. Permite pasar de vivenciar el movimiento con el cuerpo, al movimiento las fichas en el tablero de mesa, hasta llegar finalmente a la programación de las instrucciones para que la robotita logre el objetivo...”*

Los materiales y recursos utilizados en las actividades previas a la experimentación fueron tarjetas de direcciones, una cuadrícula dibujada en el suelo, un tablero de mesa, dados, fichas o

tapas de botellas de gaseosa, y el robot Bee-Bot.

De una actividad que duró aproximadamente dos horas en el aula, se documentó paso a paso la puesta en marcha de la experimentación. A continuación, se describen objetivos parciales y actividades realizadas para obtener los resultados experimentales.

### **Objetivos**

Antes de comenzar las actividades en el aula con los alumnos, se estableció el propósito de la experimentación. Este trabajo es el resultado de un trabajo en conjunto con la docente a cargo de las salas antes mencionadas.

Los objetivos principales de docente a alumno, de esta puesta en aula son:

- Desarrollar habilidades para la orientación espacial y la acción del cuerpo en relación a las instrucciones adelante, atrás, derecha, izquierda.
- Comprender el significado de las tarjetas de comandos proporcionadas con el equipamiento.
- Analizar una secuencia ordenada de instrucciones, con el registro de las distintas interpretaciones de los alumnos.

Los objetivos principales para el feedback de este proyecto, en la puesta en aula se pueden resumir en los siguientes:

- Comprobar la efectividad que se ha logrado con los cambios en la programación del robot en cuanto a la no necesidad de una conectividad física.
- Observar y documentar el impacto que tiene la presentación de las tarjetas, su forma y manera de interacción con ellas a fin de crear un programa.
- Que los alumnos comprendan que la actividad que están realizando es la programación del robot y que luego del desarrollo de la secuencia, el robot hará lo que ellos le indiquen.

### **Presentación de las tarjetas**

Los alumnos exploraron y realizaron un intercambio oral acerca del formato de las tarjetas, y de lo que representa cada figura, color y leyenda que aparece en ellas (muchos de los alumnos saben leer y escribir). En una primera instancia se realizó con un grupo reducido de alumnos, un grupo quedó en otra sala e iban incorporándose previa consulta de quien quería participar.

Antes que la docente mostrara el minirobot, se debatió acerca de las diferencias entre éste y

Bee-Bot, especialmente en lo que respecta al uso de las tarjetas. En este momento, se enfocó la atención en las tarjetas, cuya forma de pieza de rompecabezas las vuelve un elemento lúdico conocido, que utilizan en diversas actividades.

### **Puesta en conocimiento de la actividad a desarrollar**

La actividad inicial fue la “*construcción de caminos*” con las tarjetas. Los alumnos comenzaron a construirlos, guiados por la docente atendiendo el intercambio oral en la construcción de los mismos, para llegar a un consenso y concluir en la propuesta guiada por la docente. En este punto, como complemento a la actividad, la docente comparó esta construcción con el uso de los botones del robot Bee-Bot, generando así la inquietud del funcionamiento del nuevo robot, con la consecuente explicación del funcionamiento de la aplicación.

### **Secuencia didáctica: Construcción de caminos: Recorrido con obstáculos**

La secuencia propuesta por la docente estuvo centrada en los *giros*. Se propuso un recorrido con obstáculos ubicados en el piso de la sala, y se inició la interacción entre la docente y los alumnos. En principio, la docente escuchó y colocó las fichas según las propuestas de los alumnos, y puso la aplicación en funcionamiento. Posteriormente, surgieron inquietudes por parte de los alumnos, tales como el cambio de algunas tarjetas por otras según el color, o direcciones diferentes de las flechas, entre otros.

Luego de algunos cambios sugeridos por los alumnos según los distintos requerimientos para la misma secuencia, se volvió a poner la aplicación en funcionamiento.

Se repitió varias veces entre grupos de no más de cuatro alumnos para evitar la dispersión y también, por sugerencia de la docente, que los tiempos de la actividad no se extiendan de manera tal que se pierda el interés por parte de los alumnos.

Cabe destacar que, a pesar de que los teléfonos celulares no eran elementos de uso habitual para los alumnos en el ámbito escolar, en este tipo de actividades, su incorporación y uso fue muy natural y sin ninguna complicación extra.

### **Conclusiones y trabajos futuros**

Minirobots app nació como un proyecto con el objetivo de proporcionar una dinámica innovadora en el aula. Antes de comenzar con el proyecto, se determinaron objetivos enfocados en mejorar la usabilidad de Minirobots y generar una experiencia atractiva para los alumnos. A la propuesta inicial se agregaron tarjetas físicas para añadir un elemento lúdico a la hora de diseñar programas para el robot.

En primer lugar, es importante destacar que la conectividad con el robot y su facilidad de uso experimentaron mejoras significativas. En este sentido, se trabajó exitosamente en una dinámica de conexión que resuelve los impedimentos de la conexión, sin distraer de la didáctica y las actividades que formula el docente, logrando eliminar la necesidad de utilizar una computadora para operar el robot.

Las tarjetas diseñadas cumplieron su cometido como alternativa didáctica atractiva e innovadora para que los alumnos puedan diseñar programas y enviarlos al robot. Luego de varias iteraciones buscando un equilibrio entre la usabilidad de las tarjetas y la precisión de su reconocimiento, se logró un diseño atractivo y funcional que despertó el interés y la participación de los alumnos durante las pruebas realizadas.

Por su parte, el desarrollo de la aplicación dejó varios puntos altamente positivos superando los objetivos iniciales del proyecto, que se resumen a continuación:

- Se logró soportar un abanico amplio de dispositivos móviles sin importar sus prestaciones ni su antigüedad.
- Se consiguió una tasa de aciertos del 95% para el reconocimiento de imágenes, brindando una experiencia de uso fluida y eficiente.
- La posibilidad de reordenar y editar las tarjetas reconocidas en la imagen aumentó la tolerancia a errores, y generó una aplicación más segura y estable.
- Los docentes que probaron la aplicación la entendieron y utilizaron con facilidad, siendo ellos también parte activa de la creación de secuencias didácticas adecuadas para probar la aplicación.

Como trabajos futuros, el principal y a corto plazo, es la presentación de la aplicación en distintos jardines de infantes en la ciudad y en el área de influencia. Será un desafío desarrollar distintas actividades como la elaboración de secuencias didácticas (tarea que se desarrolla en conjunto con docentes). Esto es debido a que la capacitación en docentes de distintas instituciones es totalmente desigual, además de la disponibilidad de introducir el tema en los proyectos institucionales.

Hay varias posibilidades de mejoras futuras para la aplicación Minirobots. En primer lugar, se podría agregar un símbolo a las tarjetas de programación para permitir que los niños que aún no saben leer reconozcan la orientación correcta de las mismas. Este agregado surge luego de la experimentación que se hizo con los niños observando que algunos de ellos confundieron las flechas hacia arriba y hacia abajo, o derecha a izquierda, por ser muy similares, lo que los llevó a utilizarlas indistintamente.

**Referencias**

- F. Chollet (2018) Deep Learning with Python. Manning Publications Co. Levenshtein substring minimal distance. <https://gist.github.com/josejuan/2875084>
- MiniRobots (2023) <http://minirobots.com.ar/>.
- Robert C Martin (2002) Design Principles and Design Patterns. objectmentor.com. Archived from the original on 2021.
- Robert C Martin (2017) Clean Architecture: A Craftsman's Guide to Software Structure and Design. cleancoders.com. Retrieved November 2021.
- Prieto, Francisco y de Fiore Fermín (2023) Minirobots App: una aplicación educativa para aprender a programar jugando (2023). Tesis de grado Ingeniería de Sistemas. Facultad de Ciencias Exactas. UNCPBA.
- S. Rusell y P. Norvig (2020) Artificial Intelligence: A modern approach (Fourth Edition).

## Hacia un Análisis de Contenidos y Lenguajes de Programación en Materias Iniciales de Programación de Carreras de Ciencias de la Computación

José Massa<sup>1</sup>, Carmen Leonardi<sup>1</sup>, Virginia Mauco<sup>1</sup>, Laura Felice<sup>1</sup>, Jorge Doorn<sup>2</sup>  
{jmassa,cleonard, vmauco, lfelice}@exa.unicen.edu.ar, jdoorn@untref.edu.ar

<sup>1</sup> INTIA - Facultad de Ciencias Exactas.  
Universidad Nacional del Centro de la Provincia de Buenos Aires

<sup>2</sup> Universidad Nacional del Oeste

<sup>2</sup> Universidad Nacional de Tres de Febrero

### Resumen

Decidir cuál o cuáles lenguajes utilizar para enseñar los elementos básicos de la programación de computadoras no es una tarea sencilla, ya que se deben tener en cuenta aspectos notoriamente diferentes, tales como la expresividad del lenguaje para representar los conceptos a transmitir, la riqueza conceptual del mismo y las posibilidades laborales, entre muchos otros. Tan solo hacer una lista de los lenguajes a considerar es en sí mismo un problema. Conocer las decisiones adoptadas por universidades con características similares brinda una información valiosa dado que se puede aprovechar la enorme experiencia existente sobre el tema. En este contexto es útil disponer de información actualizada acerca de los contenidos y de los lenguajes que se están utilizando en las materias introductorias de programación. Esta información permite intercambiar y compartir experiencias que ayuden a enriquecer mutuamente las propuestas académicas. En este trabajo se describen los resultados de una revisión de los lenguajes de programación utilizados y de los contenidos de la primera materia de programación de un conjunto de carreras (licenciaturas y/o ingenierías) de Ciencias de la Computación de 4 o más años de duración de Universidades Nacionales de Argentina. Se presenta un análisis de los lenguajes de programación más utilizados y su relación con el conjunto de contenidos mínimos detectados en cuanto al proceso de enseñanza y aprendizaje.

**Palabras clave:** Materias de Programación, Carreras de Ciencias de la Computación, Lenguajes de Programación

### Introducción

Más allá que el objetivo de una materia introductoria de programación no es la mera enseñanza de un lenguaje sino que se enfoca principalmente en la enseñanza de los conceptos básicos de programación haciendo uso de un lenguaje, la elección del mismo no es una tarea sencilla (Dapozo, 2022), (Gómez, 2021). La decisión de la elección del lenguaje a utilizar para enseñar los conceptos introductorios de la programación no es una tarea sencilla debido a que se deben

tener en cuenta aspectos de diferente naturaleza como la expresividad del lenguaje para representar los conceptos a transmitir, su riqueza conceptual y las posibilidades laborales, entre muchos otros (Parker 2006). Tal como como se menciona en (Gómez, 2020) *“Cuando se decide dar un curso inicial de programación, la variedad de enfoques y características hace que, a la hora de elegir el lenguaje o entorno, los docentes nos vemos abrumados de opciones, y muchas veces no contamos con los elementos necesarios para tomar una decisión informada. Esto lleva a que se caiga en una elección azarosa, por criterios basados en modas o por simple desconocimiento de otras opciones, antes que tener en cuenta las implicaciones que tal decisión tendrá en el éxito de nuestro curso”*. Esta elección es válida para cualquier nivel educativo y en particular, en el nivel universitario, donde más allá que los estudiantes tengan una educación previa de programación, posiblemente sea la primera aproximación académica a la misma. En este contexto se han publicado trabajos donde se evalúan experiencias en el tema (Gómez, 2020), (Gómez, 2021), (Murphy, 2016), (Mason, 2018), permitiendo enriquecer y ponderar con mayor propiedad cualquier propuesta académica que se considere. En el presente artículo se describen los resultados de una revisión y análisis de los lenguajes de programación utilizados y de los contenidos de la primera materia de programación de un conjunto de carreras (licenciaturas y/o ingenierías) en Ciencias de la Computación de 4 o más años de duración de Universidades Nacionales. El trabajo se centra en el análisis de los lenguajes de programación sin considerar herramientas y entornos de desarrollo debido a que en cierta manera estos exceden el objetivo de las materias de programación de las carreras consideradas.

En la sección 2 se presenta la metodología utilizada para la recolección de los contenidos mínimos de las materias pertenecientes a las carreras seleccionadas y los lenguajes de programación que se utilizan en esas materias, así como también un procedimiento para establecer un conjunto representativo de contenidos mínimos para ser analizados en el contexto de las características de cada uno de los lenguajes. En la sección 3 se presenta un conjunto de características de cada uno de los lenguajes detectados, mientras que en la sección 4 se realiza un análisis exploratorio en cuanto a las características de los lenguajes y su cercanía con los contenidos mínimos. Finalmente en la sección 5 se presentan las conclusiones y líneas de trabajo futuro.

### **Metodología de recolección y análisis de información**

Para la recolección y análisis de la información, se definió una metodología consistente en los siguientes pasos:

1. **Selección de carreras**: Tomar como punto de partida la página del Consejo

Interuniversitario Nacional (CIN)<sup>1</sup>, ya que desde el mismo, se puede acceder directamente a la página oficial de cada universidad para detectar cuáles ofrecen carreras de Ciencias de la Computación.

2. **Recolección de Contenidos Mínimos:** Para cada una de esas carreras, detectar en el plan de estudios la primera materia introductoria<sup>2</sup> de la programación. Para cada materia, buscar contenidos mínimos y programa analítico.
3. **Registración de la Información:** Representar la información recolectada en formato tabular para favorecer el procesamiento posterior.
4. **Obtención del conjunto de Contenidos Mínimos Representativos:** Obtener, a partir de la información del paso anterior, un conjunto de contenidos mínimos a través de una técnica de agrupamiento automático de la información para disminuir el potencial sesgo que podría haber de hacerlo manualmente.

A continuación se presenta la aplicación de los pasos presentados anteriormente y la información obtenida en cada uno.

**Selección de carreras:** El dominio considerado para la recolección de información fue el de las materias introductorias de programación de carreras de Ciencias de la Computación/Informática de una duración igual o mayor a 4 años, considerando las cinco terminales: Ingeniería en Sistemas de Información/Informática (**II/IS**), Ingeniería en Computación (**IC**), Licenciatura en Sistemas/Sistemas en Información/Análisis de Sistemas (**LS**), Licenciatura en Ciencias de la Computación (**LC**), Licenciatura en Informática (**LI**) según lo establecido en la Resolución Ministerial N° 852/2008<sup>3</sup>. A partir del listado de la página del CIN se tomó cada universidad que tuviese en su página Web alguna/s de las carrera/s de grado de las cinco terminales mencionadas anteriormente. Se analizaron los sitios Web de las 71 universidades públicas de ese listado, detectando que 46 universidades ofrecen alguna/s de ellas, encontrando en total 66 carreras (muchas universidades ofrecen más de una carrera).

**Recolección de Contenidos Mínimos:** El siguiente paso fue identificar en el plan de estudios de cada carrera la materia introductoria de programación. Si la carrera tenía en la página los contenidos mínimos o los programas actualizados de la materia, la información se extrajo de ahí. En los casos en los que no se encontró la información buscada, se enviaron e-mails institucionales (de algunas carreras no se recibió respuesta). Este relevamiento finalizó el 31 de

---

<sup>1</sup> <https://www.cin.edu.ar/instituciones-universitarias/>

<sup>2</sup> Se considera materia introductoria de programación a la que introduce conceptos básicos que permiten a un estudiante pensar, diseñar y construir una solución a un problema en un lenguaje de programación

<sup>3</sup> <https://www.argentina.gob.ar/normativa/nacional/resoluci%C3%B3n-852-2008-142546>

julio del 2023, con lo que no se tuvieron en cuenta actualizaciones posteriores.

**Registración de la Información:** Se construyó una tabla con la información de Universidades, Carreras, Materias, Lenguajes de Programación, Contenidos Mínimos y Datos de Contacto. La información que respecta a los lenguajes se resume en las Tablas 1 y 2. En la Tabla 1 se muestra la cantidad de carreras que usan un determinado lenguaje de programación en una primera materia de programación. En dos carreras se encontró que en esa materia introductoria, se usan dos lenguajes de programación (se duplicó la materia para reflejar el uso de los dos lenguajes). Además, se observa que existe una presencia de lenguajes clásicos en las materias introductorias analizadas.

	IS/II	IC	LS/LI/LAI	LC	LI	Total
<b>Pascal/Turbo Pascal</b>	4	2	2	1	1	<b>10</b>
<b>Python</b>	1		3	1		<b>5</b>
<b>Java</b>	1		1	1		<b>3</b>
<b>C</b>	2	1	3	2		<b>8</b>
<b>C++</b>	2	1				<b>3</b>
<b>QBasic</b>	1					<b>1</b>
<b>QDraw</b>					1	<b>1</b>

Tabla 1 - Cantidad de carreras/lenguaje de programación

En algunas carreras la materia introductoria utiliza solamente pseudocódigo (y en algunos casos algún entorno como por ejemplo PseInt). En estos casos, se buscó la materia correlativa que introducía un lenguaje de programación, obteniendo los resultados que se muestran en la Tabla 2.

	IS/II	IC	LS/LI/LAI	LC	LI	Total
<b>Pascal/Turbo Pascal</b>			3		2	<b>5</b>
<b>C</b>	3	1	2	1	1	<b>8</b>
<b>ADA</b>			1		1	<b>2</b>

Tabla 2 - Cantidad de carreras/lenguaje de programación post pseudo código

**Obtención del conjunto de Contenidos Mínimos Representativos:** De las 66 carreras seleccionadas, se obtuvieron los contenidos mínimos o programas de 44 carreras. Para este primer relevamiento se obtuvo un conjunto de contenidos mínimos comunes. En este primer trabajo exploratorio, se decidió limitar el conjunto a 10 contenidos mínimos representativos del conjunto total, ya que se considera que esta cantidad es razonable para definir los contenidos mínimos de una materia. En el Anexo 1 se presenta el procedimiento de agrupamiento que se realizó para la obtención del conjunto. A modo de ejemplo, en la Figura 1 se muestran las nubes

de palabras correspondientes las pruebas de agrupamiento realizadas. Se limitó el número de palabras de cada nube a 5 por cuestiones de legibilidad.



Figura 1 - Nubes de palabras de pruebas de agrupamiento

A partir del proceso de agrupamiento se establecieron los siguientes contenidos mínimos representativos: Tipos de Variables, Programación Estructurada, Tipos de Datos indexados, Estructuras de Control y Repetición, Pasaje de Parámetros, Manejo dinámico de memoria, Paradigmas de Programación, Entrada y Salida de Datos, Tipos Abstractos de Datos, Resolución de Problemas. De esta lista se excluyeron los ítems Paradigmas de programación y Resolución de Problemas por tratarse de contenidos introductorios de naturaleza descriptiva o transversales al uso de un lenguaje seleccionado.

### **Análisis de Características de los Lenguajes de Programación relevados**

Los lenguajes de programación pueden clasificarse de acuerdo a su similitud en términos de un conjunto de características o aspectos entre los que se pueden mencionar la ligadura de tipos de datos, el mecanismo de resolución de ámbito o alcance, la posibilidad del anidamiento de unidades de ejecución entre muchos otros. En este trabajo, de entre todos los aspectos posibles de los lenguajes de programación, se tomó un conjunto de ellos según su relación con el conjunto de contenidos mínimos comunes obtenido y para cada aspecto se establecieron un conjunto de posibilidades. Se confeccionó una tabla, que se puede ver en el Anexo 2, conteniendo los aspectos relevados y para cada uno de ellos se colocó una breve definición que describe cómo se los consideró, a los efectos de desambiguar las posibles interpretaciones de las características de los lenguajes.

En la Tabla 3 se muestran los valores característicos de cada uno de los lenguajes detectados<sup>4</sup> para cada uno de los aspectos seleccionados para el análisis de los lenguajes.

<sup>4</sup> Al no disponer de la información acerca de las versiones de cada Lenguaje de todas las materias, se decidió considerar una versión del lenguaje correspondiente a una ventana de tiempo que no incluye los últimos 2 años ni tampoco versiones anteriores al año 2000.

Característica	Valor	Pascal	C	C++	Java	Python	ADA	QBasic <sup>5</sup>
Ligadura de tipos	Estática	x	x	x	x		x	x
	Dinámica					x		
Resolución de ámbito	Profundo	x	x	x	x	x	x	x
	Superficial							
Manejo de punteros	Explícito	x	x	x			x	x
	Implícito				x	x		
Manejo de memoria	Usuario	x	x	x			x	
	Lenguaje				x	x		x
Anidamiento de unidades de ejecución	Con anidamiento	x				x	x	x
	Sin Anidamiento		x	x	x			
Tamaño de Tipos de Datos indexados (arreglos)	Fijos y Flexibles	x	x	x			x	x
	Flexibles				x	x		
Tipos de las Unidades de ejecución	Solamente funciones y procedimientos	x	x					x
	Funciones, Procedimientos y métodos			x		x	x	
	Solamente métodos				x			
Tamaño de Tipos de Datos indexados (arreglos)	Fijos y Flexibles	x	x	x			x	x
	Flexibles				x	x		
Tipos de las Unidades de ejecución	Solamente funciones y procedimientos	x	x					x
	Funciones, Procedimientos y métodos			x		x	x	
	Solamente métodos				x			
Verificación de límites de datos indexados (arreglos)	Si	x			x	x	x	x
	No		x	x				
Control de Semántica de Pasaje de Parámetros	Usuario	x		x			x	
	Lenguaje		x		x	x		x
Estructuras de control	Convencionales	x	x	x			x	x
	Convencionales e iteradores				x	x		
	Solamente iteradores							
Recursión	Si	x	x	x	x	x	x	x
	No							

<sup>5</sup> Se tuvo en cuenta que QBasic hace referencia a una familia de implementaciones del lenguaje Basic, por lo cual se colocaron las características de los QBasic más utilizados.

Entrada y Salida	Bajo/medio nivel	x	x	x			x	
	Alto nivel				x	x		x
Encapsulamiento	Automático							
	Manual			x	x	x	x	
Herencia	No	x	x					x
	Si			x	x	x	x	
Tipos de Datos definidos por el usuario	Convencionales	x	x					x
	Clases y convencionales			x			x	
	Clases				x	x		
Características Funcionales	Si			x	x	x	x	
	No	x	x					x

Tabla 3 - Características de los lenguajes

### **Análisis de compatibilidad de contenidos mínimos y características de los lenguajes**

En general, puede decirse que no hay una adecuación inmediata de las propiedades de los lenguajes considerados a los contenidos de cualquier materia (Kunkle, 2014) (Koulouri, 2016). Toda situación de esta naturaleza lleva a tomar una de las tres posibles decisiones siguientes: i) privilegiar los contenidos mínimos realizando todos los esfuerzos necesarios para desarrollarlos apropiadamente en el lenguaje elegido, ii) privilegiar la apropiada enseñanza de las características del lenguaje realizando las concesiones necesarias en el cubrimiento de los contenidos mínimos o iii) conciliar ambas visiones realizando concesiones de uno y otro lado. Sin que deba considerarse una toma de posiciones de ningún tipo, en el presente trabajo se evaluó el uso de los lenguajes detectados siguiendo el primer criterio en virtud de tratarse de carreras acreditables que deben cumplir con las pautas establecidas por la CONEAU (Resolución Ministerial N° 852/2008).

Para cada uno de los contenidos mínimos seleccionados, se analizó una propuesta de enseñanza utilizando dos lenguajes diferentes, uno cuyas características eran cercanas al contenido mínimo y otro más distante. A continuación se muestran algunos ejemplos que ilustran esta visión de la relación entre un contenido mínimo y el esfuerzo requerido para su enseñanza:

#### **Escenario 1: Enseñanza de programación estructurada**

En este escenario hipotético, desde el punto de vista de las tareas relacionadas a la ejercitación, las consignas típicamente utilizadas son aquellas en las que el estudiante debe poder plasmar su solución en un conjunto de unidades de ejecución que sigan una cierta estructura de programa. Un ejemplo de este tipo de consigna podría ser: “Escribir un programa, el cual dado un arreglo de enteros, permita devolver el valor mínimo o máximo del arreglo”. En la Tabla 4 se muestran dos posibles soluciones con dos lenguajes que poseen diferentes características en cuanto a las unidades de ejecución estructurada que ofrecen.

**Solución 1:** Esquema de solución posible en un Lenguaje que ofrece unidades de ejecución estructuradas en funciones y procedimientos. En este esquema de solución, se pueden identificar claramente las unidades en las que se estructuran las operaciones a realizar sobre el arreglo de valores enteros.

**Solución 2:** Esquema de solución posible en un lenguaje que únicamente ofrece unidades de ejecución bajo la forma de métodos. En esta solución, el lenguaje fuerza a que el programador deba definir una clase para el programa, dentro de la cual se define el método main, junto con los métodos que implementan la solución al problema. Además, es necesario introducir el modificador **static** para que los métodos puedan ser invocados sin necesidad de crear previamente un objeto y el modificador **public** para que los métodos puedan ser visibles al exterior de la clase. Adicionalmente, en este lenguaje los arreglos son variables de naturaleza dinámica por lo que es necesario crearlo con el uso de **new**.

La necesidad de representar en un lenguaje orientado a objetos una visión conceptual ajena al mismo obliga a introducir un conjunto de artificios que vistos desde el punto de vista del lenguaje, no tienen relación con el contenido y vistos desde el punto de vista del aprendizaje exigen la aceptación de un conjunto de *recetas* con alguna eventual explicación que excede al contenido y pudiendo introducir una confusión en la enseñanza que se pretende llevar a cabo.

Solución 1	Solución 2
<pre>#define MAX_TAMANO 100 int minimo(int arreglo[], int tamano) {     min = arreglo[0];     for ... {         ...     }     return min; }  int maximo(int arreglo[], int tamano) {     ... }  int main() {     ...     return 0; }</pre>	<pre>public class MinMax {     public static void main(String[] args) {         MAX_TAMANO = 100;         int[] arreglo = new int[MAX_TAMANO];         ...     }      public static int Minimo(int[] arreglo){         min = arreglo[0];         for ... {             ...         }         return min;     }      public static int Maximo(int[] arreglo){         ...     } }</pre>

Tabla 4 - Posibles soluciones al Escenario 1 (Solución 1 en C y Solución 2 en Java)

### **Escenario 2:** Enseñanza de semántica de pasaje de parámetros

En este otro escenario hipotético, también en el contexto de las tareas relacionadas a la

ejercitación, se encuentran consignas clásicas en las que el estudiante deba escribir un programa donde en un caso una función use el valor de una variable pasada como parámetro y en otro caso modifique dicho valor. En la Tabla 5 se muestran dos posibles soluciones utilizando dos lenguajes con diferentes características en cuanto al pasaje de parámetros.

**Solución 1:** Esquema de solución posible en un lenguaje que ofrece mecanismos sintácticos para que el programador pueda indicar la semántica de pasaje de parámetros deseada. En esta solución, el lenguaje provee mecanismos sintácticos para que el programador pueda, en este caso, escribir la función “f” y el procedimiento “g” que poseen las semánticas adecuadas para la solución al problema.

**Solución 2:** Esquema de solución posible en un lenguaje que no ofrece mecanismos sintácticos para que el programador pueda indicar la semántica de pasaje de parámetros deseada. En este caso, el lenguaje determina automáticamente y de forma implícita la semántica de pasaje de parámetros. Es así que los objetos serán pasados siempre por referencia y las variables de tipos básicos siempre por copia valor. De modo que la única forma en la que se pueda cambiar la semántica de pasaje de parámetros es cambiando la naturaleza del parámetro real. Como se ve en el ejemplo, para modificar un valor de tipo entero, se asigna este valor al atributo x de la instancia c1 de una clase C y se pasa a la función g esta instancia como parámetro.

Esta forma de abordaje hace que la solución sea también más compleja. Claramente para poder ejercitar el pasaje de parámetros es nuevamente necesario introducir un conjunto de indicaciones sin explicación o brindando una explicación mínima acerca del funcionamiento del lenguaje. Cualquiera de estas estrategias implica apartarse en mayor o menor medida de los contenidos mínimos. Otra alternativa podría ser dar el contenido de manera conceptual sin ahondar en la práctica para no tener que tratar con las características del lenguaje.

Solución 1	Solución 2
<pre> program LeerModificar; var   x: Integer;  function f(v: Integer): Integer; begin   ... end;  procedure g(var v: Integer); begin   ... end;  begin </pre>	<pre> class C:   x = 1    def f(p1):     print (p1)    def g(p2):     p2.x = 3  z = 1 f(z)          # se envía z con valor 1 c1 = C() c1.x = z g(c1)        # se envía c1 con valor 1 en x print (c1.x)# imprime 3 </pre>

f(x); g(x); end.	
------------------------	--

Tabla 5 - Posibles soluciones al Escenario 2 (Solución 1 en Pascal y Solución 2 en Python)

Los ejemplos anteriores muestran el “esfuerzo” para enseñar y aprender un determinado contenido en un lenguaje de programación dadas las características que posee dicho lenguaje. Se entiende por esfuerzo aquellos aspectos extras al contenido objetivo que hay que agregar, pudiendo también optar por dejar prácticas y conceptos para ser profundizados en otras materias del área a lo largo de la carrera. La Tabla 6 intenta reflejar este esfuerzo para los contenidos mínimos considerados, considerando tres posibles niveles de esfuerzo didáctico.

Contenido de la materia	Característica del lenguaje	Esfuerzo Didáctico		
		Directo	Intermedio	Extra
Tipos de Variables	Ligadura de Tipo	Estático		Dinámico
Programación Estructurada	Alcance	Profundo		Superficial
	Tipos de Unidades de ejecución	Funciones y procedimientos		Solamente métodos
		Procedimientos, Funciones y métodos		
	Anidamiento	Con anidamiento	Sin anidamiento	
Tipos de Datos indexados	Tamaño de Tipos Indexados	Fijos y Flexibles		Solamente Flexibles
	Verificación de límites de datos indexados	Si		No
Estructuras de Control	Estructuras de iteración	Convencionales		Solamente iteradores
		Convencionales e Iteradores		
Pasaje de Parámetros	Control de semántica de pasaje de parámetros	Usuario		Automática del lenguaje
Manejo dinámico de memoria	Manejo de punteros	Explícito		Implícito
	Administración de memoria	Usuario		Lenguaje
Entrada y Salida de Datos	Archivos	Bajo nivel		Alto nivel
Tipos Abstractos de Datos	Encapsulamiento	Automático	Manual	No

Tabla 6 - Relación entre contenido y “esfuerzo” para enseñar y aprender en un lenguaje determinado

## Conclusiones y Trabajos Futuros

En este artículo se presentó un relevamiento de los contenidos mínimos y los lenguajes de programación utilizados en una materia introductoria de programación en carreras de las 5 terminales. También, se presentó un análisis de los lenguajes de programación más utilizados y

su relación con el conjunto de contenidos mínimos detectados en cuanto al proceso de enseñanza y aprendizaje. Esta información permite tener una visión general de cómo se está desarrollando la introducción a la programación en las carreras de CC, convirtiéndose en un insumo para intercambiar y analizar experiencias. Se detectó que la mayoría de las carreras utilizan los mismos lenguajes de programación (de las 44 carreras analizadas, se encontró un conjunto de 7 lenguajes de programación).

Asimismo, se analizaron estos lenguajes en cuanto a sus características y el “esfuerzo” que se debe realizar para el proceso de enseñanza de los contenidos mínimos detectados.

Es importante destacar, que en la selección de un lenguaje seguramente también se consideran otros aspectos, por ejemplo, motivación para los estudiantes, facilidad de uso, relevancia para la industria, entre otros. No se trata de decir si un lenguaje es mejor que otro sino que al elegir un lenguaje se deberán considerar los pro y los contra en cuanto a las características del mismo en relación a los contenidos a enseñar. Esto implica realizar una elección consciente teniendo en claro los alcances del lenguaje, los esfuerzos necesarios en el abordaje de los contenidos mínimos y la enseñanza de aquellos conceptos que no se darán en profundidad y deberán ser abordados en otras materias. En este sentido es importante remarcar que las materias no son islas sino que son eslabones de un trayecto formativo que implica varias materias que abordarán la temática, es así que, las carreras tienen varias materias de programación, donde sería ideal evaluar el área de manera integral.

Al haber elegido a lo largo de todo el artículo, privilegiar los contenidos mínimos por sobre las características del lenguaje, hace necesario estudiar con mayor profundidad aquellas situaciones complejas que ocurren en algunos casos, para ponderar apropiadamente si las ventajas de usar un determinado lenguaje efectivamente compensan esas dificultades.

Dentro de los trabajos futuros se pretende recolectar información de una mayor cantidad de carreras y mejorar diferentes etapas en cuanto al proceso de adquisición de información y su análisis. Se propone ampliar este estudio incluyendo carreras de Universidades privadas y además se puede realizar un nuevo análisis sobre las carreras de corta duración. En cuanto al agrupamiento de los contenidos mínimos, la utilización de un modelo o diccionario de lenguaje técnico/académico produciría mejores resultados. Asimismo, sería deseable realizar alguna encuesta cualitativa a nivel docente y estudiantes para analizar las experiencias concretas incorporando de manera más eficaz el factor humano.

Para finalizar, queremos agradecer especialmente a todos los docentes y personal administrativo de diferentes instituciones que respondieron los mails brindándonos la información solicitada.

## **Referencias**

- Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8), 1295.
- Dapozo, G. N., Greiner, C. L., Petris, R. H., Irrazábal, E. A., Espíndola, M. C., & Medina, Y. (2022). Evaluación de habilidades de pensamiento computacional al inicio de una asignatura de programación en una carrera de informática.
- Gómez, M. J. (2020). Aspectos de adquisición de lenguaje en la enseñanza de programación. *Universidad Nacional de Córdoba-Facultad de Matemática, Astronomía, Física y Computación. Universidad Nacional de Córdoba*. <https://rdu.unc.edu.ar/handle/11086/16051>.
- Gómez, M. J., Martínez López Pablo E. "Fidel", "¿ Scratch, Python, o qué?", Jornadas Argentinas de Didáctica de las Ciencias de la Computación JADiCC 2021.
- Koulouri, T., Lauria, S., & Macredie, R. D. (2014). Teaching introductory programming: A quantitative evaluation of different approaches. *ACM Transactions on Computing Education (TOCE)*, 14(4), 1-28.
- Kunkle, W. M., & Allen, R. B. (2016). The impact of different teaching approaches and languages on student learning of introductory programming concepts. *ACM Transactions on Computing Education (TOCE)*, 16(1), 1-26.
- Mason, R., Crick, T., Davenport, J. H., & Murphy, E. (2018, February). Language choice in introductory programming courses at Australasian and UK universities. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 852-857).
- Murphy, E., Crick, T., & Davenport, J. H. (2016). An analysis of introductory programming courses at UK universities. arXiv preprint arXiv:1609.06622.
- Parker, K. R., Ottaway, T. A., & Chao, J. T. (2006). Criteria for the selection of a programming language for introductory courses. *International Journal of Knowledge and Learning*, 2(1-2), 119-139.
- Petrovic, S. (2006, October). A comparison between the silhouette index and the davies-bouldin index in labelling ids clusters. In *Proceedings of the 11th Nordic workshop of secure IT systems* (Vol. 2006, pp. 53-64). Citeseer.
- Vasiliev, Y. (2020). *Natural language processing with Python and spaCy: A practical introduction*. No Starch Press.
- Zhao, G., Liu, Y., Zhang, W., & Wang, Y. (2018, January). TFIDF based feature words extraction and topic modeling for short text. In *Proceedings of the 2018 2nd international conference on management engineering, software engineering and service sciences* (pp. 188-191).

## **Anexo 1: Proceso de Agrupamiento para obtener el conjunto representativo de contenidos mínimos.**

1. **Pre-procesamiento de Datos:** Se transcribieron manualmente los contenidos mínimos de cada materia que estaban contenidos en diferentes formatos, a un documento de formato uniforme. Se llevó a cabo una limpieza del texto eliminando caracteres inválidos, detectando cada contenido mínimo y colocándolo en oraciones separadas por el carácter “coma”, se eliminaron oraciones con errores léxicos y sintácticos y oraciones conteniendo frases largas (más de 20 palabras).
2. **Procesamiento de la información:** Se utilizó un modelo de Procesamiento de Lenguaje Natural para generar a partir de cada oración que refiere a un contenido mínimo, una lista de tokens reconocidos en el idioma Castellano. El modelo utilizado fue Spacy “es\_core\_news\_sm” (Vasiliev, 2020).
3. **Vectorización de contenidos mínimos:** Se convirtió cada contenido mínimo del conjunto de datos en un vector de valores numéricos a partir de las características pre-calculadas de las palabras que componen el contenido mínimo. Se utilizó la biblioteca TfidfVectorizer (Zhao, 2018).
4. **Clustering:** Se aplicó el método k-means++ (Ahmed, 2020) para agrupar los contenidos mínimos (expresados en vectores característicos obtenidos en el paso 3), en 10 grupos o clusters según su similitud.
5. **Evaluación cualitativa y cuantitativa:** Si bien la evaluación cuantitativa de los resultados del clustering arrojó resultados de 0.6 para el índice de Davies-Bouldin (Petrovic, 2006). Al analizar cualitativamente los resultados, se puede observar que la clusterización podría mejorarse a futuro por medio de un preprocesamiento de texto más exhaustivo y empleando modelos de lenguaje natural específico para contenido técnico informático de índole académica.
6. **Visualización:** Se realizó una reducción de dimensionalidad a 2 dimensiones para graficar los clusters encontrados y se generaron gráficos de nubes de palabras para describir cada cluster.

En un principio, el proceso de clusterización no arrojó clusters adecuados para el análisis, ya que por ejemplo en algunos casos el contenido mínimo “Tipos de Datos de Variables” se dividió en 2 o más clusters y en otros casos dos contenidos mínimos diferentes los agrupó en un único cluster debido a que son similares respecto de una o más palabras en común. Una estrategia posible de solución consiste en aumentar la cantidad de clusters, lo cual produce clusters más homogéneos pero la cantidad de contenidos mínimos finales es excesiva. Finalmente, a los efectos de este trabajo se decidió conformar una lista de contenidos mínimos comunes donde

cada uno es una frase representativa de un conjunto de contenidos mínimos que aparecen en al menos una prueba de clusterización.

## Anexo 2: Características de los Lenguajes de Programación

Característica	Valor	Descripción
Ligadura de tipos	Estática	Los tipos de datos de las variables se conocen antes del tiempo de ejecución
	Dinámica	Los tipos de datos de las variables se conocen durante el tiempo de ejecución
Resolución de ámbito	Profundo	El ámbito de los elementos del programa se determina según la organización sintáctica del programa
	Superficial	El ámbito de los elementos del programa se determina según el flujo de ejecución
Manejo de punteros	Explícito	El lenguaje provee operadores explícitos para la manipulación de variables de tipo puntero
	Implícito	El lenguaje realiza la manipulación de variables de tipo puntero de manera implícita y oculta al programador
Manejo de memoria	Usuario	El lenguaje ofrece al programador herramientas para que el programador administre la memoria de algunos tipos de variables
	Lenguaje	El lenguaje administra automáticamente la memoria de todas las variables del programa
Anidamiento de unidades de ejecución	Con anidamiento	Si el lenguaje admite la posibilidad de anidar sintácticamente unidades de ejecución con nombre (funciones o procedimientos) una dentro de otra.
	Sin Anidamiento	Si el lenguaje no admite la posibilidad de anidar sintácticamente unidades de ejecución con nombre (funciones o procedimientos) una dentro de otra.
Tamaño de Tipos de Datos indexados (arreglos)	Fijos y Flexibles	El lenguaje permite definir variables de tipo agregado con tamaño fijo o flexible a elección del programador
	Flexibles	El lenguaje permite definir variables de tipo agregado únicamente de tamaño variable o flexible.
Tipo de las Unidades de ejecución	Funciones y procedimientos	El lenguaje provee únicamente unidades de ejecución del estilo de funciones o procedimientos.
	Funciones, Procedimientos y métodos	El lenguaje provee unidades de ejecución del estilo de funciones o procedimientos y además posee objetos lo cual le permite proveer métodos.
	Solamente métodos	El lenguaje proporciona un mecanismo de orientación a objetos que únicamente permite definir métodos.
Verificación de límites de datos indexados (arreglos)	Si	El lenguaje posee un mecanismo de verificación cuando se intenta acceder a una ubicación que se encuentra fuera de los límites de la variable. Este mecanismo de verificación puede actuar en tiempo de compilación o en tiempo de ejecución.
	No	El lenguaje carece de un mecanismo de verificación cuando se intenta acceder a una ubicación que se encuentra fuera de los límites de la variable.
Control de Semántica de Pasaje de Parámetros	Usuario	El lenguaje ofrece un mecanismo sintáctico para que el programador pueda seleccionar la semántica con la que desea realizar el pasaje de parámetros entre las semánticas posibles que ofrece el lenguaje.
	Automática de Lenguaje	El lenguaje selecciona automáticamente la semántica de pasaje de parámetros según el tipo de datos que se trate y no ofrece

		mecanismos para que el programador pueda seleccionar la semántica deseada.
Estructuras de control	Convencionales	El lenguaje ofrece únicamente estructuras de control que en el caso de las iteraciones, la variable que controla la iteración es explícita y en algunos casos el programador es el responsable de su actualización.
	Convencionales e iteradores	El lenguaje ofrece estructuras de iteración convencionales y además, soporta el uso de iteradores, los cuales permiten realizar bucles y recorridos de estructuras agregadas de forma automática, siendo el lenguaje el que administra las variables que controlan la iteración.
	Solamente iteradores	El lenguaje solamente soporta iteradores y no ofrece al programador la posibilidad de usar estructuras de iteración convencionales.
Recursión	Si	El lenguaje ofrece la posibilidad de que una unidad de ejecución se llame a sí misma recursivamente.
	No	El lenguaje no permite la posibilidad de que una unidad de ejecución se llame a sí misma recursivamente.
Entrada y Salida	Bajo/medio nivel	El lenguaje ofrece mecanismos para que el programador pueda enviar y recibir información por la entrada y salida del sistema controlando la forma en la que éstas tareas son realizadas (dispositivos, unidad de intercambio de información, formato, etc.)
	Alto nivel	El lenguaje solamente ofrece mecanismos de alto nivel de abstracción para el manejo de la entrada y salida ocultando al programador los aspectos que hacen al control de estas tareas.
Encapsulamiento	Automático	El lenguaje provee un mecanismo de encapsulamiento u ocultamiento de la información por defecto que fuerza a que los componentes de determinados tipos de datos no sean visibles al exterior.
	Manual	El lenguaje permite que el programador controle el encapsulamiento de la información mediante mecanismos sintácticos.
	No	El lenguaje no ofrece ningún tipo de mecanismo de encapsulamiento, haciendo que todas los componentes de los tipos de datos sean visibles al exterior de los mismos.
Herencia	Si	El lenguaje ofrece soporte para que el programador pueda definir un tipo de datos que hereda de otro tipo de datos.
	No	El lenguaje no ofrece soporte alguno para que el programador pueda definir un tipo de datos que hereda de otro tipo de datos.
Tipos de Datos definidos por el usuario	Convencionales	El lenguaje permite que el programador pueda definir nuevos tipos de datos, los cuales permitirán la declaración de variables de estos tipos
	Clases y convencionales	El lenguaje admite que el programador defina tipos convencionales y además tipos que poseen la estructura de una clase, es decir con la posibilidad de definir atributos y métodos, además de otras características que ofrezca el lenguaje para las clases.
	Clases	El lenguaje solamente admite la definición de tipos definidos por el usuario bajo la forma de clases.
Características Funcionales	Si	El lenguaje ofrece construcciones que permiten agregar características funcionales a los programas, como por ejemplo la posibilidad de definir unidades de primera clase, expresiones lambda, closures, etc.
	No	El lenguaje no ofrece construcciones propias que permiten agregar características funcionales al programa.



## **ArguiBot IDE: Un entorno de desarrollo basado en la programación en bloque para la enseñanza de la robótica usando la plataforma Arduino**

Nicolás Streri, Marcela Daniele, Marcelo Uva  
{*nstreri, marcela, uva*}@dc.exa.unrc.edu.ar

Facultad de Ciencias Exactas, Físico-Químicas y Naturales  
Universidad Nacional de Río Cuarto Río Cuarto

### **Resumen**

En los últimos años, la enseñanza de la programación y la robótica ha adquirido gran relevancia en todos los ámbitos educativos. En este sentido se han propuesto diversos entornos de programación basados en bloques diseñados en su mayoría para la plataforma Arduino. Estas herramientas se distinguen entre sí por el contexto académico para el cual están diseñadas, por los requerimientos de hardware necesarios y por el enfoque que utilizan para introducir los conceptos de programación.

En este trabajo se presenta ArguiBot IDE, un entorno de desarrollo de código abierto diseñado para la enseñanza de la robótica a estudiantes de nivel secundario. Se basa en programación en bloque y está enfocado en proporcionar una forma sencilla de introducir conceptos de robótica y programación usando la plataforma Arduino.

ArguiBot está diseñado utilizando un conjunto reducido de bloques que simplifican la creación de algoritmos. No requiere de kits de robótica específicos, lo que reduce costos económicos para su implementación en diferentes entornos educativos. Además, ArguiBot busca mejorar la experiencia de enseñanza y aprendizaje proporcionando un conjunto de funcionalidades para el seguimiento del progreso de los estudiantes y la entrega de trabajos.

**Palabras Clave:** Robótica Educativa, Entorno de Desarrollo, Programación Basada en Bloques, Arduino.

### **Introducción**

Actualmente, escuelas, hogares e instituciones, junto al resto de la sociedad en su conjunto, interactúan diariamente con la Tecnología. Esto no sólo implica el uso de nuevos dispositivos inventados por el ser humano, sino también una nueva forma en que la sociedad se relaciona con la naturaleza, y como los aparatos se relacionan entre sí y con las personas (inventores, fabricantes y usuarios). La Tecnología es el lenguaje actual y futuro con el que los objetos artificiales se comunican y colaboran para la resolución de problemas y situaciones cotidianas.

Ante esta situación, ha comenzado a tener especial relevancia la enseñanza de las Ciencias de

la Computación en todos los niveles educativos (Borchardt & Roggi, 2017) (Llambi et al., 2023). Con el objetivo de permitir a los estudiantes desarrollar un pensamiento lógico-computacional que los habilite a resolver situaciones problemáticas del día a día mediante la utilización o construcción de nuevos recursos tecnológicos.

El estudio y desarrollo de diversas estrategias para incorporar las ciencias de la computación en todos los niveles educativos es de un interés generalizado y amplio. Particularmente en el contexto de Argentina, se han dedicado varios años al proceso de integrar la programación y el Pensamiento Computacional en el ámbito educativo. En la resolución N° 263/15<sup>1</sup>, el Consejo Federal de Educación (2015) establece que “la enseñanza y el aprendizaje de la Programación es de importancia estratégica en el Sistema Educativo Nacional”. Además, la resolución N° 343/18 aprobó los Núcleos de Aprendizaje Prioritarios (NAP) en Educación Digital, Programación y Robótica que sentaron la base para la integración de la programación y robótica en las prácticas de enseñanza y aprendizaje en la Argentina (Núcleos de Aprendizajes Prioritarios de Educación Digital, Programación y Robótica, 2018). Dentro de este marco, diversos proyectos han emergido con el propósito de facilitar la enseñanza de la programación en las aulas, como la iniciativa Program.AR y el programa Conectar Igualdad.

La enseñanza de Ciencias de la Computación implica introducir a los estudiantes en un mundo de conceptos abstractos. Ante este desafío educativo, han surgido diversas alternativas pedagógicas para hacer que el aprendizaje sea más accesible y atractivo para los estudiantes.

Entre las principales alternativas se encuentran el desarrollo de videojuegos con herramientas como Scratch<sup>2</sup>, la construcción de algoritmos que resuelven escenarios o situaciones problemáticas como, por ejemplo, la aplicación Pilas Bloques<sup>3</sup>, o alternativas más tangibles como la Robótica Educativa. Estas opciones han sido cuidadosamente diseñadas para adaptarse a distintos niveles educativos y edades, lo que las convierte en herramientas versátiles y adecuadas para estudiantes de diversas etapas de aprendizaje. Ofrecen enfoques pedagógicos variados y contenidos ajustados a las capacidades de cada grupo.

En cuanto a la robótica educativa, esta tiene un alto poder de motivación en los estudiantes y permite cubrir múltiples áreas de conocimiento, entre las cuales se destacan física, mecánica, electrónica, matemática, programación, entre otras. Todas ellas muy importantes para el desarrollo académico y profesional de estudiantes, que reciben una formación integral para afrontar los desafíos presentes y de trabajos del futuro. Además, la robótica se convierte en un valioso recurso pedagógico al proporcionar una forma tangible y concreta de comprender y

<sup>1</sup> [http://www.bnm.me.gov.ar/giga1/normas/RCFE\\_263-15.pdf](http://www.bnm.me.gov.ar/giga1/normas/RCFE_263-15.pdf) (Último acceso: 25/09/2023)

<sup>2</sup> <https://scratch.mit.edu/>

<sup>3</sup> <https://pilasbloques.program.ar/>

dimensionar los conceptos abstractos involucrados en diversas áreas científicas (Horn et al., 2009).

La popularización de la robótica como recurso pedagógico, dio lugar al surgimiento de múltiples herramientas o entornos de desarrollos que facilitan su enseñanza. Desde una perspectiva amplia, es posible distinguir entre las herramientas diseñadas con un propósito específico y aquellas de propósito general. La mayoría de estas herramientas son de uso específico, es decir, vienen acompañadas de un manual, un robot con hardware personalizado y el entorno de programación suele tener bloques o código destinado a operar dicho hardware exclusivo. En contraste, los entornos de propósito general brindan la flexibilidad de programar aplicaciones para placas Arduino y similares.

En cuanto a las herramientas diseñadas para kits de robótica específicos, como micro: bit<sup>4</sup>, Mis Ladrillos Robots<sup>5</sup> y mBlock<sup>6</sup> a menudo presentan una limitación significativa al estar estrechamente vinculadas a un hardware específico. Incluyen piezas y componentes necesarios para realizar un tipo particular de proyecto, lo que restringe la posibilidad de crear soluciones personalizadas y encontrar nuevas aplicaciones para el kit. Suelen tener un costo elevado y en algunas regiones (como Argentina) la disponibilidad de estos kits es limitada o nula para su compra.

En cuanto a las herramientas de propósito general, brindan versatilidad, flexibilidad en la programación y posibilidades de proyectos prácticamente ilimitadas. Entre las comúnmente mencionadas se encuentran ArduBlock<sup>7</sup>, S4A<sup>8</sup>, Visualino<sup>9</sup>, BlocklyDuino<sup>10</sup> y Facilino<sup>11</sup>. Todas incluyen una amplia variedad de bloques que permiten operar con la mayoría de los sensores y actuadores disponibles en el mercado. Sin embargo, en sus interfaces gráficas presentan una gran cantidad de bloques y, en ocasiones, algunos de ellos tienen conceptos técnicos o frases poco intuitivas. Lo cual puede abrumar y frustrar a los usuarios, y son situaciones que deben evitarse según las recomendaciones de los desarrolladores de Blockly (Pasternak et al., 2017). Biblioteca en la que se basan la mayoría de las herramientas previamente mencionadas para renderizar el editor de código visual (bloques).

A pesar del avance en la implementación de la enseñanza de la programación y la robótica, persisten desafíos importantes. Se observa que, algunas de las herramientas existentes en la

---

<sup>4</sup> <https://microbit.org/>

<sup>5</sup> <https://misladrillos.com/ml/>

<sup>6</sup> <https://www.mblock.cc/en/>

<sup>7</sup> <http://ardublock.ru/3/>

<sup>8</sup> <http://s4a.cat/>

<sup>9</sup> <http://www.visualino.net/>

<sup>10</sup> <https://github.com/BlocklyDuino/>

<sup>11</sup> <https://roboticafacil.es/facilino/blockly/Facilino.html>

actualidad no resultan totalmente adecuadas ni facilitan el trabajo áulico y colaborativo entre docentes y estudiantes. Si bien, algunas herramientas permiten a los estudiantes crear cuentas y acceder a plataformas en línea, a menudo resulta complicado para los docentes tener un acceso fluido a las soluciones y proyectos de los estudiantes, así como brindar una retroalimentación efectiva.

En este artículo, se presenta ArguiBot IDE un entorno de desarrollo para la enseñanza de los conceptos básicos de la robótica en los primeros años de la educación secundaria. El entorno está basado en programación por bloques, y ofrece una combinación de las características de las herramientas de propósito general y las de propósito específico revisadas con anterioridad.

ArguiBot IDE provee un conjunto reducido de bloques, estrechamente relacionados al núcleo de Arduino, que permiten trabajar con una amplia variedad de módulos *plug and play* disponibles en el mercado. Elimina la abundancia de bloques que puede resultar confusa y deja al docente la construcción de su propio kit de enseñanza en base a componentes genéricos y al presupuesto disponible. ArguiBot IDE no está acompañado ni depende de un kit de robótica de diseño personalizado. Además, este entorno de desarrollo, provee una organización del trabajo áulico en “Grupos de Trabajo”, facilita el acceso del docente a las soluciones de los estudiantes mediante un sistema de entrega automática, y permite el seguimiento del progreso mediante un sistema de calificaciones.

En las siguientes secciones se describe en detalle la herramienta, las funcionalidades y características que posee, y se mencionan las actividades abordadas utilizando el entorno junto a los resultados obtenidos. Finalmente, se presentan las conclusiones y el trabajo futuro.

### **ArguiBot IDE**

ArguiBot IDE, o simplemente ArguiBot, es un entorno de desarrollo pensado para introducir los conceptos iniciales de la robótica en los primeros años de la educación secundaria. Su nombre tiene origen en la combinación de las palabras: **A**rgentina, **A**rduino y **R**obótica.

ArguiBot tiene como objetivo facilitar el uso de la robótica como soporte para la enseñanza de conceptos relacionados a los Sistemas Digitales, los Sistemas de Control y el diseño de algoritmos. Destaca, principalmente, la representación de los tipos de datos (número, textos, imágenes como mapas de bits, colores), transmisión de la información, protocolos de comunicación, interacción con el usuario, estructuras de control, variables y constantes, modularización, sensores, actuadores, controladores y hardware libre. Al mismo tiempo, permite fomentar la imaginación, la creatividad y el trabajo en equipo al ser útil como herramienta para dar soluciones, a través de dispositivos robotizados, a problemáticas de la sociedad actual.

Mediante su diseño intuitivo y amigable, ArguiBot IDE busca que tanto docentes como estudiantes puedan comenzar a utilizarla sin la necesidad de una curva de aprendizaje pronunciada. El conjunto de bloques propuesto en ArguiBot está especialmente diseñado para evitar abrumar a los estudiantes con una multitud de opciones, priorizando el aprendizaje práctico y el desarrollo de proyectos funcionales a través de elementos básicos de programación.

El entorno se diseñó considerando el trabajo en equipo, el seguimiento del progreso de los estudiantes por parte del docente, y el costo inicial de equipar el aula para robótica. Además, a través del uso de componentes genéricos y ampliamente disponibles, ArguiBot busca reducir significativamente los costos y facilitar la adquisición de los componentes necesarios para la implementación de los proyectos.

A continuación se explican las principales características de la herramienta propuesta en este trabajo (Punto A), la interfaz de usuario con la que interactúan los estudiantes (Punto B) y algunos detalles técnicos sobre su implementación (Punto C).

## A. Características Principales

### Secuencia de Proyectos

En la aplicación, un ejercicio o actividad que el docente propone a los estudiantes reciben el nombre de “Proyecto”. Cada uno se caracteriza por:

- **Título:** Un nombre que identifica la actividad.
- **Consigna:** Un texto que detalla la situación problemática a resolver.
- **Código Base:** Un conjunto de bloques o secciones de algoritmo ya propuestos por el docente como ayuda/guía para resolver la actividad.

ArguiBot permite organizar los proyectos en una secuencia autoguiada, en donde los estudiantes pueden ir seleccionando el proyecto a resolver (Ver Figura 1).

Además, se cuenta con la función “Restablecer Proyecto” que permite eliminar todas las modificaciones realizadas por el alumno y dejar solamente el código base propuesto por el docente.



Figura 1: Ejemplo de una Secuencia de Proyectos.

### Revisión del Docente

Arguibot IDE, permite a los educadores realizar un seguimiento detallado de las resoluciones realizadas por los estudiantes. Luego de realizado un proyecto, el docente puede calificarlo con una escala (Correcto / Revisar) y dejar un mensaje con detalles de los puntos a revisar.

Por otra parte, los estudiantes tienen la posibilidad de observar las correcciones recibidas y reenviar los proyectos para ser revisados nuevamente. En la Figura 2 se observan los iconos utilizados para indicar a los estudiantes las correcciones de sus ejercicios y un ejemplo de corrección.



Figura 2: Recortes de pantalla. Iconos de corrección (Superior).  
Mensaje detallando la corrección recibida (Inferior).

## Trabajo Áulico

Al trabajar con recursos tecnológicos en el aula, surgen algunos desafíos. Durante la captura de requerimientos se identificaron principalmente las siguientes situaciones a resolver: ante la escasez de computadoras se debe recurrir al trabajo grupal; es necesario recordar y compartir credenciales personales o recursos físicos (como *pendrive*) para almacenar y acceder a los archivos grupales; pérdida o no disponibilidad de los archivos o soluciones realizadas en jornadas de trabajo previas; las soluciones se deben enviar al docente por email o copiar los archivos a su computadora personal de forma manual.

En el diseño de ArguiBot se resuelven estos desafíos mediante la funcionalidad “Grupo de Trabajo”. Un “grupo” imita el comportamiento de las cuentas de usuario, con la característica principal de que sólo cuentan con un identificador que puede compartirse entre todos los integrantes del grupo de trabajo.

Antes de comenzar a trabajar, el docente puede generar los identificadores (Por ejemplo: “grupoAzul”) y los comparte con los integrantes de los grupos, sin que los estudiantes tengan que registrarse previamente ni compartir datos personales de inicio de sesión.

Al ingresar con un identificador de grupo, las soluciones a los distintos proyectos se sincronizan desde y hacia un servidor. De esta manera, se elimina la necesidad de contar con un *pendrive*, ya que se puede acceder desde cualquier computadora en donde esté instalada la aplicación. Y las soluciones de los estudiantes están siempre disponibles de forma automática para que el docente pueda revisarlas y darles una devolución.

## Adaptación a la Placa de Desarrollo

Las distintas placas de desarrollo Arduino están basadas en microcontroladores que tienen diferentes características (Arduino.cc, n.d.). Por ejemplo, la cantidad de pines digitales y analógicos.

ArguiBot permite seleccionar la placa de desarrollo con la que se está trabajando, y utiliza esta

información para adaptar los bloques a las características específicas de la placa. En la Figura 3 se observa un ejemplo de esta función.

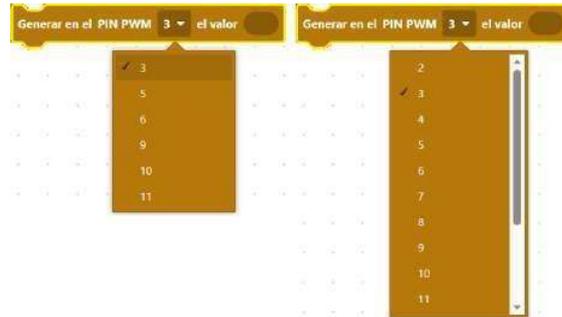


Figura 3: A la izquierda pines disponibles en Arduino UNO, a la derecha pines disponibles en Arduino Mega.

Además, cada proyecto se asocia a un modelo de placa de desarrollo. En caso de que el estudiante cuente con otra placa seleccionada, al momento de abrir el proyecto se informará de tal situación (Figura 4). Permitiendo evitar errores de compilación asociados al uso de características propias de la placa.

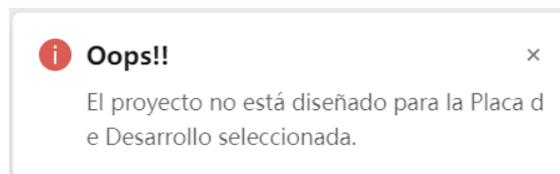


Figura 4: Mensaje de alerta al seleccionar una placa no compatible con el proyecto.

### Mínima Dependencia de Hardware

A diferencia de la mayoría de las herramientas de igual tipo, ArguiBot no está acompañada por un kit específico de robótica. Se diseñó pensando en los costos de preparar un aula de trabajo. Por tal motivo, se dejó de lado los kits de robótica que suelen incluir manuales, placas y modelos de robot con diseño personalizado, ya que suelen tener un costo más elevado.

En particular, se decidió que ArguiBot pueda ser utilizada con cualquier placa de desarrollo que soporte el núcleo Arduino AVR<sup>12</sup>. Respecto a los sensores y actuadores, se orientó la compatibilidad a módulos genéricos, los cuales son construidos por múltiples fabricantes, están basados en el mismo esquema eléctrico, y se pueden adquirir fácilmente a un precio moderado.

Se provee un conjunto de bloques que permiten trabajar con muchos de los módulos de sensores y actuadores disponibles en el mercado que no requieren de bibliotecas específicas para su uso. Por lo general, trabajan con simples señales digitales o analógicas.

Además, se añadieron bloques que permiten trabajar con la *Shield Multifunción* (Figura 5). Una

<sup>12</sup> <https://github.com/arduino/ArduinoCore-avr>

placa de expansión genérica, de muy bajo costo y con disponibilidad en varios países que incluye sensores y actuadores interesantes para proyectos introductorios.



Figura 5: A la izquierda, ejemplos de módulos de sensores y actuadores compatibles.

A la izquierda, Shield Multifunción.

### Actualizaciones Automáticas

ArguiBot es una herramienta en desarrollo, por tal motivo, se decidió incorporar la posibilidad de recibir actualizaciones de forma automática. Se pueden recibir dos tipos de actualizaciones. Por un lado, del binario de la aplicación cuando se producen cambios en la interfaz de usuario, bloques disponibles o las APIs nativas utilizadas. Por otro lado, de las bibliotecas utilizadas por el compilador, como por ejemplo, las relacionadas a los Servos, receptores infrarrojos, etc.

### B. Interfaz de Usuario

La interfaz de usuario de ArguiBot IDE ha sido cuidadosamente diseñada para proporcionar una experiencia fluida y agradable tanto para docentes como para estudiantes. Se enfoca en la eficiencia y la facilidad de uso, permitiendo a los usuarios concentrarse en la creatividad y la resolución de problemas en lugar de lidiar con una complejidad innecesaria. La interfaz se divide en dos áreas principales: la barra de menú y el área de codificación.

Respecto a la barra de menú, para no sobrecargar la visual, se decidió añadir botones para las tareas de uso frecuente como ver la consigna, compilar y subir, rehacer/deshacer cambios, guardar los cambios y gestión de integrantes del grupo de trabajo. Además, sólo si el proyecto tiene una devolución del docente, se muestra el respectivo botón para visualizarla. La selección de puerto y placa utilizada, ver el código generado, cambiar de proyecto, cambiar a modo oscuro, entre otras opciones se ocultaron bajo un menú desplegable.

En cuanto al área de codificación, a la izquierda se encuentra la caja de herramientas en donde los estudiantes pueden localizar los distintos bloques disponibles para utilizar. Los mismos se encuentran organizados en categorías identificadas por colores, y en algunas ocasiones en subcategorías. Luego, hacia la derecha se encuentra el lienzo de trabajo en donde se va construyendo el algoritmo mediante la combinación de bloques usando un gesto de arrastrar y

En la Figura 6, se puede apreciar la interfaz de usuario de ArguiBot. En la misma se puede observar, la caja de herramientas desplegada en la categoría “Toma de Decisión” y un algoritmo de ejemplo que al presionar un botón de la *Shield Multifunción* enciende y apaga un LED.

Además, ArguiBot cuenta con un sistema de tema claro y tema oscuro que permite a los estudiantes adaptar la interfaz a sus preferencias y necesidades visuales. Dicha funcionalidad se incorporó con el fin de garantizar una experiencia óptima ante largas sesiones de programación y ante distintas condiciones de iluminación.

Asimismo, la implementación de esta funcionalidad de tema claro y oscuro no solo busca mejorar la experiencia de uso, sino que también representa un primer paso hacia la inclusión de características de accesibilidad en la herramienta. Este enfoque busca garantizar que todos los usuarios, independientemente de sus capacidades físicas, puedan utilizar la herramienta de manera efectiva y cómoda.

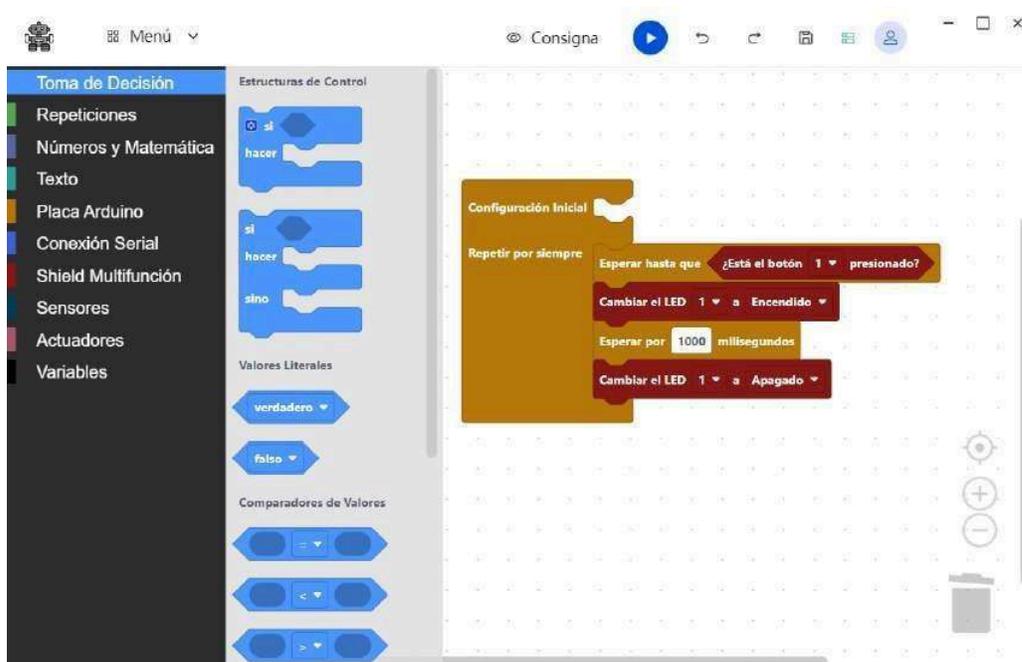


Figura 6: Captura de pantalla de la interfaz de usuario de ArguiBot.

### C. Arquitectura e Implementación

ArguiBot IDE está diseñada como una aplicación híbrida<sup>13</sup> de escritorio multiplataforma<sup>14</sup> que no requiere permisos de administrador para ser instalada y ejecutada. Se encuentra implementada utilizando Tauri<sup>15</sup>, un *toolkit* de construcción de aplicaciones híbridas, que permite acceder

<sup>13</sup> Entendiendo como aplicación híbrida a aquella que combina elementos de aplicaciones nativas y aplicaciones web.

<sup>14</sup> Sistemas operativos de PC: Linux, Windows y MacOS.

<sup>15</sup> <https://tauri.app/>

mediante una aplicación web a las APIs nativas del sistema operativo, como la ejecución de subprocesos y la manipulación del sistema de archivos. Además, trabaja en conjunto a un servidor de sincronización que permite almacenar las soluciones de los estudiantes.

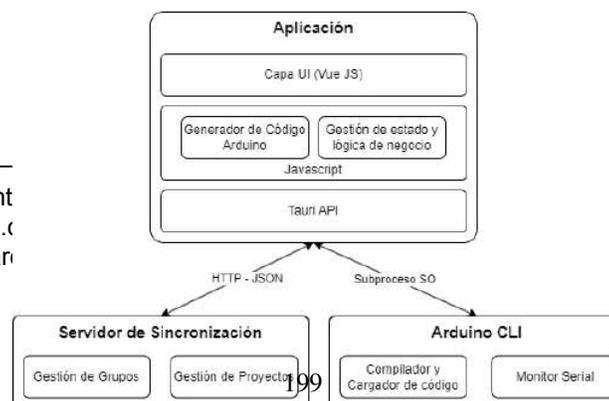
Respecto a la interfaz de usuario, la misma se ejecuta dentro de un navegador web embebido. Está construida utilizando Vue JS con componentes visuales de la biblioteca TDesign<sup>16</sup>. La interfaz incluye una barra de menú y un área de codificación. Respecto a esta última, la misma está implementada utilizando la biblioteca Blockly<sup>17</sup> con bloques personalizados relacionados con la plataforma Arduino.

En cuanto al funcionamiento general de ArguiBot, la construcción del algoritmo mediante bloques, el acceso a los proyectos, la sincronización y guardado en el servidor de los avances de los estudiantes, la revisión de las correcciones realizadas por el docente y la generación del código Arduino suceden a nivel web. Mientras que el proceso de compilación y carga del programa a la placa de desarrollo se realizan a nivel nativo utilizando las API provistas por Tauri y la herramienta Arduino CLI<sup>18</sup>.

En relación a la conectividad, al estar todos los recursos instalados de forma local en la computadora, solo se requiere una conexión a Internet para descargar el proyecto actualmente abierto, subir las modificaciones del usuario y obtener las correcciones del docente desde el servidor de sincronización. Toda esta comunicación se realiza sobre HTTP utilizando JSON, por lo que el consumo de red se mide en unos pocos kilobytes permitiendo ser usada con datos móviles o una conexión a Internet poco estable. Además, está la posibilidad de instalar el servidor de sincronización en una computadora local (por ejemplo, la del docente) y trabajar por LAN.

En cuanto al servidor de sincronización, el mismo cuenta con las funcionalidades de backend mínimas requeridas por ArguiBot. Se decidió implementarlo utilizando PHP como lenguaje de programación, principalmente por la facilidad y bajos costos de alojamiento. Los datos de trabajo se almacenan mediante un conjunto de archivos JSON organizados en carpetas. Por limitaciones de recursos, se decidió priorizar el desarrollo de la aplicación, dejando la incorporación de características de estabilidad y seguridad en el servidor como línea de trabajo futuro.

A modo de resumen, en la Figura 7, se presenta un esquema general de la arquitectura de la herramienta propuesta.



<sup>16</sup> <https://github.com/Tencent>

<sup>17</sup> <https://developers.google.com/blockly/>

<sup>18</sup> <https://arduino.github.io/arduino-cli/>

Figura 7: Esquema de la arquitectura de ArguiBot.

Por último, en cuanto a su distribución, ArguiBot cuenta con un instalador en donde se encuentran empaquetados todos los recursos de la interfaz de usuario y el compilador de Arduino.

### Una Experiencia de Uso

Una versión preliminar de la herramienta, fue utilizada en la segunda etapa del 2022 en el dictado del espacio curricular “Educación Tecnológica” con estudiantes secundarios de 3<sup>er</sup> año.

El trabajo áulico se organizó en grupos de cuatro integrantes, en donde cada grupo contó con una *notebook* y un kit de robótica (Placa Arduino Uno, Shield Multifunción y sensores).

Las primeras actividades tuvieron como objetivo introducir el entorno de desarrollo y recordar las nociones de programación adquiridas en la primera etapa del año. Consistieron en tareas simples como el encendido de un led, mostrar números en pantalla de 7-segmentos, emitir sonidos, detectar pulsaciones de botones, entre otras. Los estudiantes lograron aplicar conceptos de composición secuencial, condicional y operadores lógicos y matemáticos. Estas actividades motivaron a aquellos estudiantes más curiosos y creativos a diseñar programas más avanzados.

Esta primera experiencia con estudiantes permitió detectar algunos fallos en el software e inconvenientes relacionados al trabajo áulico. Entre los que se pueden mencionar la construcción de programas sintácticamente no válidos a partir de la combinación de ciertos bloques, bloqueos en la comunicación con el compilador, confusiones por parte de los estudiantes a la hora de ejecutar el servidor local<sup>19</sup> y la necesidad de disponer de un *pendrive* por grupo en cada clase para almacenar los avances. Todas estas observaciones, permitieron definir y delimitar el objetivo de la herramienta, reemplazar la manera de comunicarse con el compilador, diseñar la gestión de los archivos de los estudiantes eliminando la necesidad de un *pendrive* y la construcción de un *software* más estable y robusto.

Como resultado final de la experiencia, es importante destacar que todos los grupos de estudiantes pudieron desarrollar los proyectos finales propuestos: Barrera de Paso Nivel, detector de incendios y monóxido de carbono, y sistema de aparcamiento por ultrasonido.

Además, en instancia de evaluación se pudo observar que un 80% de los estudiantes logró adquirir los conceptos abordados, tales como, el diseño de soluciones algorítmicas, estructuras de control y abstracción, y la manipulación de nuevos sensores no utilizados durante el dictado. Siendo un resultado alentador para continuar empleando la herramienta para impartir conceptos

---

<sup>19</sup> La versión utilizada en ese momento requería de un servidor local que actuaba como intermediario entre el navegador web y el compilador, actualmente es reemplazado por las API de Tauri.

de robótica e incorporar a la misma como base en la enseñanza de otros temas del espacio curricular como es la representación y transmisión de datos.

## **Conclusiones**

En los últimos años, conceptos como pensamiento computacional, algoritmos, programación, robótica educativa fueron tomando relevancia en el ámbito de la educación secundaria. En este trabajo se ha presentado ArguiBot IDE, un entorno de desarrollo *open source* basado en programación por bloques para la enseñanza de la robótica a estudiantes de nivel secundario.

La herramienta facilita al docente la organización del trabajo en una secuencia de proyectos, permitiendo que los estudiantes puedan aprender gradualmente y resolver desafíos prácticos que aumentan en complejidad.

Los “Grupos de Trabajo” y la funcionalidad de seguimiento del progreso del estudiante son características destacadas de ArguiBot IDE. El entorno permite una interacción fluida entre docente y estudiantes, lo que facilita la revisión y la retroalimentación en el proceso de enseñanza.

Además, la elección de componentes genéricos en lugar de kits específicos de robótica ayuda a reducir los costos para las instituciones educativas y proporciona mayor flexibilidad en la construcción de proyectos.

Se desarrolló una primera experiencia de uso con una versión preliminar de la herramienta. Dicha experiencia permitió detectar algunos fallos y complicaciones relacionadas al trabajo áulico. Además, la herramienta facilitó el dictado de conceptos relacionados a los sistemas de control, sensores y actuadores, brindando a los estudiantes un ambiente de trabajo práctico y de experimentación.

ArguiBot IDE representa un valioso recurso para la enseñanza de la robótica y programación, enfocado en brindar a los estudiantes una experiencia práctica, creativa y colaborativa para enfrentar los desafíos del mundo tecnológico actual. A través de esta herramienta, se busca fomentar la imaginación, la creatividad y la construcción de habilidades de pensamiento computacional. Al mismo tiempo, ArguiBot brinda un recurso tecnológico que permite hacer tangible muchos de los conceptos tratados en los espacios curriculares relacionados a la tecnología.

## **Trabajo Futuro**

Actualmente, se cuenta con una versión estable de ArguiBot. La cual está siendo utilizada para el dictado 2023 del espacio curricular “Educación Tecnológica”. Se espera poder seguir obteniendo

*feedback* de esta segunda experiencia de uso.

En la actualidad, el desafío es evaluar el impacto de estas iniciativas en la escuela (Bocconi et al., 2016), en particular, se prevé evaluar esta experiencia intentando medir la construcción de habilidades de pensamiento lógico-computacional de los estudiantes y su apropiación y aplicación en su desempeño y desarrollo.

Respecto a la herramienta, se proyecta continuar con su desarrollo; aún existen algunas características pendientes de ser incorporadas. A continuación, se detallan las principales:

- Extender el conjunto de bloques disponibles. Faltan incorporar bloques que permitan trabajar con LED RGB, emisor y receptor de radiofrecuencia, entre otros. Siempre conservando la filosofía de mantener la herramienta como un entorno simple para introducir las primeras nociones de robótica.
- Finalizar la integración del Monitor Serial, para eliminar la dependencia de herramientas externas al usar comunicación serial.
- Continuar con la incorporación de características de accesibilidad. En particular, se pretende añadir una alternativa al gesto arrastrar y soltar mediante una navegación basada en teclado.
- Se pretende refactorizar el servidor de sincronización integrando un motor de bases de datos que permita eliminar la dependencia del sistema de archivos.
- Incorporar una pantalla de configuración que permita al docente, personalizar los bloques disponibles para resolver un proyecto. Permitiendo deshabilitar aquellos bloques que no sean relevantes para la situación problemática actual.
- Implementar un Editor de Dashboard, que permita al usuario incorporar componentes (gráficos lineales, medidores o indicadores, pantallas virtuales) para visualizar en la PC los datos capturados mediante los sensores conectados a la placa de desarrollo.

### **Licencia y Uso de ArguiBot**

En un esfuerzo por promover la colaboración y la mejora continua, ArguiBot IDE se distribuye como un proyecto de código abierto bajo la licencia MIT.

La licencia MIT, conocida por su simplicidad y flexibilidad, brinda a los usuarios la libertad de utilizar, modificar y distribuir el software tanto para fines académicos como comerciales. Este enfoque fomenta la colaboración y el intercambio de conocimientos, dos pilares fundamentales que ayudan a extender y mantener el desarrollo de la herramienta en el tiempo.

Los interesados pueden descargar el entorno de desarrollo y contribuir activamente a su

evolución a través del siguiente repositorio: <https://github.com/nicostreri/ArguiBot>

## Referencias

- Arduino.cc. (n.d.). *Arduino Hardware | Arduino*. Retrieved August 21, 2023, from <https://www.arduino.cc/en/hardware>
- Bocconi, S., Chiocciariello, A., Dettori, G., & Ferrari, A. (2016). Developing computational thinking in compulsory education-Implications for policy and practice. *Developing Computational Thinking in Compulsory Education-Implications for Policy and Practice*.
- Borchardt, M., & Roggi, I. (2017). *Ciencias de la computación en los sistemas educativos de América Latina*.
- Núcleos de Aprendizajes Prioritarios de Educación Digital, Programación y Robótica, 343/18 (2018). <https://www.educ.ar/recursos/150123/nap-de-educacion-digital-programacion-y-robotica>
- Horn, M. S., Solovey, E. T., Crouser, R. J., & Jacob, R. J. K. (2009). Comparing the use of tangible and graphical programming languages for informal science education. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 975–984. <https://doi.org/10.1145/1518701.1518851>
- Llambi, C., Borchardt, M., Klinkovich, V., & Locca, N. (2023). *Aprendizajes y desafíos para la enseñanza de las Ciencias de la Computación en las escuelas*.
- Pasternak, E., Fenichel, R., & Marshall, A. N. (2017). Tips for creating a block language with blockly. *2017 IEEE Blocks and Beyond Workshop (B&B)*, 21–24. <https://doi.org/10.1109/BLOCKS.2017.8120404>

**Los juegos de adivinación:  
Análisis didáctico de una propuesta para la enseñanza**

Flavia Buffarini<sup>1</sup>, Fabiana Rosso<sup>2</sup>, Francisco Bavera<sup>1</sup>

fbuffarini@exa.unrc.edu.ar, fabianarosso67@gmail.com, pancho@dc.exa.unrc.edu.ar

<sup>1</sup> Facultad de Ciencias Exactas, Físico-Químicas y Naturales  
Universidad Nacional de Río Cuarto Río Cuarto

<sup>2</sup> Instituto Superior de Formación Docente Menéndez Pidal

### **Resumen**

Con el compromiso de un sostenido y desafiante estudio de *tipos de problemas* que favorezcan el desarrollo del pensamiento computacional. En este trabajo, en primer lugar, se realiza un desarrollo teórico que fundamenta el abordaje de la clasificación en post de la construcción de dicho pensamiento. En segundo lugar, se resignifica la noción de juego que justifica su inclusión en situaciones para la enseñanza. Una vez explicitado el marco, se estudia un tipo de problemas que habilita el desarrollo de capacidades vinculadas a la clasificación en la escuela. En esta oportunidad se quiere poner a consideración una propuesta que involucra problemas de *juegos de adivinación*. En términos generales, los *juegos de adivinación* consisten en que el jugador o el equipo de jugadores adivine lo que se pretende de la manera más eficiente posible a través de preguntas dicotómicas. En este trabajo se abordan algunos de estos problemas y se analizan condiciones para que se constituyan en un tipo de problemas que aporten a la construcción del pensamiento computacional en la enseñanza. Como así también, se analizan las variables didácticas para formular nuevas instancias de estos juegos. El estudio de estos problemas, en el marco de la Didáctica de la Matemática, permite dar cuenta que bajo ciertas condiciones didácticas, los mismos responden a las características que adquiere un problema para desencadenar, de manera implícita, un proceso de clasificación. Así, este tipo de juegos contribuyen al conjunto de actividades desenchufadas que favorecen la introducción y/o fortalecimiento de la alternativa condicional y la construcción de modelos de predicción -árboles de decisión-. En este sentido se considera que los juegos de adivinación resultan potentes para ser incluidos en un proyecto de enseñanza de las ciencias de la computación tanto en la escuela como en la formación docente.

**Palabras Claves:** Pensamiento Computacional, Clasificación, Juegos de adivinación, Didáctica de la Matemática, Didáctica de las Ciencias de la Computación.

### **Introducción**

Estudios y tendencias a nivel mundial dan cuenta de la importancia de introducir y desarrollar el

pensamiento computacional (PC) (Wing, 2008; Brennan y Resnick, 2012) en el sistema educativo obligatorio. Entonces, es relevante el rol del docente en el contexto de los desafíos tendientes a promover la construcción del PC, capacidades y competencias útiles para la búsqueda de soluciones a diversos tipos de problemas que aporten a la formación de ciudadanos críticos, participativos y autónomos.

Por otra parte, numerosas investigaciones concluyen que la mayoría de los docentes que en la actualidad enseñan en los primeros años de educación básica (en especial en países emergentes), no transitaron una formación que incluyera de manera manifiesta el desarrollo de habilidades asociadas al PC como la abstracción, el reconocimiento de patrones, la exhaustividad y/o la sistematización de datos, entre otras. Esto genera deficiencias significativas cuando propuestas educativas en torno al PC son implementadas en la escuela. En este sentido se destaca la necesidad de desarrollar investigaciones didácticas que deriven en propuestas que puedan adaptarse a cada *aula* según se requiera. Aún más, es imprescindible que estén disponibles los análisis y justificaciones de las propuestas para que los docentes que las implementen puedan realizar la transposición de conocimiento de manera sustentada. Propuestas con un fuerte componente didáctico explícito permiten explorar con mayor profundidad las dimensiones, la importancia y beneficios de las actividades tendientes a desarrollar habilidades del PC y contribuyen a la formación de los docentes que las implementen.

En este trabajo se pretende mostrar el rol de cierta clase de juegos *-juegos de adivinación o juegos de descubrir la figura-* en la construcción de procesos de *clasificación* y por ende en la construcción del PC. A partir de este estudio se genera una primera propuesta. Se detalla un *análisis didáctico-matemático* que examina las características de estos juegos para que favorezcan el desarrollo de la habilidad de clasificar.

## **Marco Teórico**

### **La Construcción del Pensamiento Computacional y la Clasificación**

Si bien no hay un estricto consenso en cuanto a la definición del PC existen habilidades comunes en que coinciden la mayoría de los autores (Catlin y Woollard, 2014; Wing, 2006, 2008, 2011; Brennan y Resnick, 2012; Bers, 2018; ISTE y CSTA 2011; Csizmadia, et al., 2015) y que se consideran en este trabajo. Estas habilidades son: *abstracción, pensamiento algorítmico, reconocimiento de patrones, descomposición, generalización, análisis lógico y evaluación.*

Aunque la *clasificación* no aparece explícitamente entre las habilidades mencionadas, entendemos que la misma, en tanto se refiere a la capacidad de organizar elementos, datos o información en grupos, categorías o secuencias lógicas según ciertos criterios o propiedades específicas, no se limita a una sola de las habilidades del PC mencionadas. Más bien, está

inherentemente relacionada con la mayoría de ellas y en ese sentido consideramos muy valiosos los problemas que hacen explícito su estudio para la construcción del PC.

A continuación se describen las habilidades necesarias a desarrollar en la construcción del PC. Luego, se presenta la relación de estas habilidades con los procesos involucrados en la *clasificación*.

- La **abstracción** implica simplificar y/o eliminar detalles innecesarios, lo que facilita la comprensión de los problemas. La habilidad en la abstracción radica en decidir qué detalles ocultar para hacer que el problema sea más manejable, sin perder información esencial. Un aspecto crucial de este proceso es elegir una representación adecuada del sistema, ya que diferentes representaciones simplifican diferentes aspectos del problema.
- El **Pensamiento Algorítmico** es un enfoque para resolver problemas mediante la definición clara de los pasos necesarios.
- El **Reconocimiento de Patrones** consiste en identificar regularidades, estructuras o características en un conjunto de datos u objetos con el fin de tomar decisiones, hacer predicciones o extraer información valiosa a partir de estos patrones.
- La **Descomposición** es una técnica que implica analizar *artefactos* dividiéndolos en sus partes y componentes individuales. Cada parte se comprende, resuelve, desarrolla y evalúa por separado. Esto simplifica la resolución de problemas complejos y facilita el diseño de sistemas más grandes.
- La **Generalización** se relaciona con identificar patrones, similitudes y conexiones, y utilizar estas características para aplicar soluciones a una variedad de contextos. Permite resolver rápidamente nuevos problemas basándose en experiencias anteriores y adaptar algoritmos y/o resoluciones de problemas a clases enteras de problemas similares.
- El **Análisis Lógico** se refiere a la aplicación y la interpretación correcta de la lógica booleana.
- La **Evaluación** implica sistematizar y juzgar soluciones mediante diferentes criterios, que incluyen la depuración, la validación y la verificación. Esta habilidad involucra el uso del análisis lógico para predecir y confirmar los resultados obtenidos.

Para organizar datos o elementos de manera económica que sea a la vez significativa y útil, el proceso de *clasificación* implica la combinación de abstracción, descomposición, reconocimiento de características comunes -patrones-, pensamiento algorítmico, generalización, análisis lógico y evaluación.

Por lo tanto, la clasificación puede considerarse una habilidad interdisciplinaria que abarca múltiples aspectos del PC. Inicialmente, antes de clasificar de manera objetiva, es necesario descomponer el problema o conjunto de datos en conjunto de elementos más pequeños y manejables, así, de manera implícita se está clasificando. Cabe destacar que el **reconocimiento de patrones** es fundamental en la habilidad de clasificar. La misma implica identificar patrones, esto es características comunes entre elementos o datos y agruparlos en función de esas características. Asimismo la clasificación puede llevar a la **generalización**, a partir de la identificación de patrones comunes en un grupo de elementos podrían generalizarse otros conjuntos similares. Esto puede ser útil para extrapolar información y tomar decisiones basadas en la clasificación. Por otro lado, la clasificación requiere un **análisis lógico** para determinar cómo se deben organizar los elementos en función de los criterios de clasificación. Esto implica razonamiento lógico para tomar decisiones coherentes y consistentes en el proceso de clasificación. También para determinar cómo realizar la clasificación de manera eficiente se requiere de habilidades de **pensamiento algorítmico**, ya que a menudo involucra procesos sistemáticos para organizar elementos en categorías o grupos. Una vez que se ha realizado la clasificación, es necesario **evaluar** la calidad y la eficacia de la misma. Esto puede implicar verificar si se han seguido los criterios de clasificación correctamente y si los grupos resultantes son apropiados.

En otras palabras, la habilidad de clasificar implica: por un lado Identificar y/o construir criterios de clasificación para organizar un conjunto de datos y por otro, dada una clasificación poder analizar si tal o cual elemento pertenece o no a cierta categoría de esa clasificación.

En este sentido, es necesario determinar los criterios o características que se utilizarán para agrupar elementos, a partir de **reconocer patrones** y similitudes entre elementos, esto es esencial para agrupar elementos de manera efectiva. Además, si están establecidos los criterios, se deben **organizar los elementos** en grupos o secuencias siguiendo esos criterios, es decir hacer un test para cada elemento. Los criterios establecidos pueden variar según el contexto y el problema.

Cabe mencionar que la clasificación se utiliza en la **resolución de problemas algorítmicos** como la búsqueda de información en bases de datos, la organización de datos en estructuras de datos eficientes o la identificación de elementos en un conjunto de datos que cumplen con ciertos criterios. En informática, se utilizan **algoritmos de clasificación** específicos para organizar datos de manera eficiente y **tomar decisiones** basadas en la clasificación o a extraer información útil a partir de datos organizados de cierta manera.

Lo antes dicho implica que la clasificación es una parte esencial del PC y se aplica en diversas áreas de la informática y la resolución de problemas algorítmicos.

Dada la importancia del desarrollo de la capacidad de clasificar en la construcción del PC es que en este trabajo se estudia un tipo de problemas que habilita su tratamiento en la escuela: Los *juegos de adivinación* o *juegos de descubrir una figura*. Los mismos contribuyen a desarrollar habilidades tales como la clasificación, la formulación de preguntas lógicas, la evaluación de información y la reducción sistemática de opciones basadas en datos limitados. También promueven la resolución de problemas y el razonamiento deductivo, lo que los hace adecuados para el desarrollo de habilidades de PC y lógico en un entorno lúdico.

La *gamificación* en la enseñanza habitualmente se define como una estrategia que utiliza elementos y mecánicas de los juegos en contextos educativos con el fin de aumentar la motivación, el compromiso y el aprendizaje de los estudiantes. Esta técnica se basa en la idea de que los juegos son intrínsecamente atractivos y desafiantes, lo que puede hacer que el proceso de aprendizaje sea atrayente y efectivo. Es importante destacar que la gamificación no se trata solo de convertir las actividades educativas en juegos, sino de utilizar estratégicamente elementos de juego para mejorar el proceso de aprendizaje.

La *didáctica* de una determinada ciencia se preocupa por cómo se enseña y se aprende esta disciplina. Uno de los objetivos de la didáctica de las ciencias de la computación es desarrollar estrategias y enfoques efectivos para desarrollar el PC de manera que sea accesible y significativo para los estudiantes.

Considerando que la didáctica de la ciencias de la computación es incipiente y se está desarrollando, es que se toman los aportes de la didáctica de la matemática en tanto nos da un marco para entender y estudiar la construcción de habilidades lógicas, algorítmicas y de resolución de problemas. Desde este lugar, en este trabajo se realiza el análisis didáctico de un tipo de problemas *-juegos de adivinación-* teniendo en cuenta que las capacidades-habilidades que se pretende desarrollar con este juego son objeto de estudio en ambas disciplinas.

A continuación se resignifica la noción de gamificación en la enseñanza y la importancia de su consideración para la construcción de conocimiento en la escuela.

### **¿Por qué pensar en jugar en las clases?**

Desde la Didáctica de la Matemática, específicamente desde la Teoría de Situaciones Didácticas (TSD) se resignifica la noción de *juego*. A continuación, primeramente, se explicita el posicionamiento de la Didáctica de la Matemática respecto a cómo se construye conocimiento. Guy Brousseau (1986) propone un modelo desde el cual pensar la enseñanza como un proceso de producción de conocimientos matemáticos. De esta manera, concibe la clase como un ámbito de producción, la cual supone tanto establecer nuevas relaciones, como transformar y reorganizar otras. Brousseau sostiene que el alumno produce conocimientos a partir de la

interacción con un *medio* resistente

El alumno aprende adaptándose a un medio que es factor de contradicciones, dificultades, de desequilibrios, un poco como lo ha hecho la sociedad humana. Este saber, fruto de la adaptación del alumno, se manifiesta por respuestas nuevas que son la prueba del aprendizaje. (Brousseau, como citado en Sadovsky, 2005, p.4).

El *medio* comprende tanto la problemática matemática inicial que el sujeto enfrenta, como un conjunto de relaciones que se van modificando a medida que el sujeto produce conocimientos en el transcurso de la situación, transformando en consecuencia la realidad con la que interactúa (Sadovsky, 2005, p.3).

La necesidad teórica de un *medio*, de acuerdo a Brousseau (1986), viene dada por el hecho de que en el futuro el estudiante deberá hacer frente a situaciones desprovistas de intenciones didácticas. La relación didáctica se habrá extinguido y el sujeto sólo habrá adquirido verdaderamente este conocimiento cuando sea capaz de ponerlo en práctica él mismo fuera de todo contexto de enseñanza. En este modelo, el proceso de producción de conocimientos se describe a partir de dos interacciones: La interacción del alumno y el medio (A-M) y la interacción del docente con el alumno, a propósito de la interacción A-M.

La interacción A-M se describe a partir del concepto teórico de *situación adidáctica*, esta situación es una interacción, un “ida y vuelta” entre el sujeto y el medio. El sujeto enfrenta el problema, toma decisiones optando por una alternativa matemática y rechazando otras, la pone en práctica y es el *medio*, a través de retroacciones, quién le permitirá al sujeto analizar los resultados/consecuencias de sus acciones para reafirmarlas o corregirlas. De esta manera el sujeto produce conocimiento, ya sea que sus decisiones se ajusten o no al problema. Esta producción modifica al *medio*: ya no sólo están en él la problemática inicial y los conocimientos iniciales, sino también el conjunto de relaciones que se van produciendo en la interacción con el mismo.

En este marco se considera al *juego* como el instrumento de modelización de una situación de enseñanza. No se debe pensar sólo en la idea lúdica de aquel, en algo que sirve simplemente para producir placer a quien lo juega, sino que, por el contrario, es esta herramienta la que permitirá que en la clase de matemática *la mejor estrategia posible* sea el conocimiento que se pretende que se aprenda. En palabras de Brousseau (1986, p.28): “El juego debe ser tal que el conocimiento aparezca bajo la forma elegida, como la solución, o como el instrumento para establecer la estrategia óptima (...) El juego específico de un saber debe justificar su empleo o su aparición.”.

Es ésta, justamente, la finalidad de la modelización respecto al conocimiento en cuestión;

mientras que, en relación a la actividad de enseñanza, Brousseau (1986) sostiene que “el juego debe permitir representar todas las situaciones observadas en las clases (...) aún las menos ‘satisfactorias’ (...) Debe poder engendrar todas las variantes (...) Estas serán obtenidas por la elección de los valores de ciertas variables<sup>1</sup> características de este juego” (p.28).

En este marco los *sentidos* acordados a la noción de *juego*, es decir, los elementos característicos de un *juego*, se sintetizan a continuación: a) El jugador y las decisiones y acciones durante el transcurso del juego, no están reglamentadas más que por el placer de jugar. b) La organización de esta actividad bajo un sistema de reglas. c) Los instrumentos del juego, lo que sirve para jugar. d) La manera como se juega. e) El *conjunto de posiciones-estados* entre las cuales el jugador puede elegir en un momento dado del juego.

Brousseau pone en relación estos cinco elementos y produce un significado institucional sobre la modelización de la noción juego desde donde se procede, en este trabajo, a analizar y rediseñar, a la luz de este significado, *el juego de adivinación*

### **Juegos de adivinación y el Pensamiento Computacional**

Los juegos en los que se elige una figura y debe encontrarse haciendo preguntas dicotómicas son ejemplos de juegos de adivinanza o juegos de deducción. Estos juegos a menudo involucran el uso de un proceso de PC similar al de los árboles de decisión, ya que los jugadores deben formular preguntas lógicas y usar la información disponible para reducir las opciones y llegar a una solución. En particular, nos ocupamos de aquellos juegos en los que se pretende adivinar una figura que llamaremos *figura secreta* y que se encuentra representada en una plantilla junto a otras de la misma clase. En estos juegos las preguntas que formula el jugador sólo pueden ser contestadas por *Sí* o por *No*. Estas preguntas dicotómicas van dividiendo al conjunto de figuras en dos partes disjuntas, en una de las cuales se encuentra la *figura a adivinar*. Así, las sucesivas preguntas darán lugar a un proceso de clasificación jerárquica que culminará con la identificación de la *figura secreta*.

Tomado como marco la noción de juego antes definida, se señalan a continuación los elementos característicos del juego de adivinación:

- **Instrumentos del juego:** Una plantilla con figuras.
- **Elección de una figura:** El jugador elige una figura o un objeto entre un conjunto de

---

<sup>1</sup> A este respecto Panizza (2003) sostiene que: (...) las situaciones didácticas son objetos teóricos cuya finalidad es estudiar el conjunto de condiciones y relaciones propias de un conocimiento bien determinado. Algunas de esas condiciones pueden variar a voluntad del docente, y constituyen una variable didáctica cuando según los valores que toman se modifican las estrategias de resolución y en consecuencia el conocimiento necesario para resolver la situación. (p.10)

opciones (por ejemplo, un animal, una carta de juego, un objeto geométrico, etc.). Este es el "*objeto objetivo*" (o *figura secreta*) que los otros jugadores intentarán adivinar.

- **Preguntas:** Los otros jugadores (o un jugador en solitario) hacen preguntas para obtener información sobre el *objeto objetivo*. Estas preguntas suelen ser de tipo *sí/no*. Por ejemplo, podrían preguntar: "*¿Es un animal?*" o "*¿Es de color rojo?*"
- **Respuestas y reducción de opciones:** El jugador que conoce el *objeto objetivo* responde a las preguntas de manera honesta y reveladora. Las respuestas ayudan a los otros jugadores a reducir las opciones posibles. Por ejemplo, si la respuesta a la primera pregunta es "sí", los otros jugadores saben que el *objeto objetivo* es un animal y pueden descartar las opciones que no son animales.
- **Continuación del proceso:** Los otros jugadores continúan haciendo preguntas y recibiendo respuestas hasta que puedan identificar con certeza el *objeto objetivo* o hasta que hayan agotado sus preguntas permitidas. Este proceso es similar a un árbol de decisión simplificado o un diagrama de flujo, donde cada pregunta representa una bifurcación en el camino y las respuestas conducen a nuevas preguntas o a la identificación final del *objeto objetivo*.

Como analizaremos más adelante, estos juegos de adivinación ponen en funcionamiento el proceso de clasificación. En algunos casos, pueden requerir la creación de un diagrama mental o gráfico para seguir las diferentes divisiones del conjunto de figuras, habilidad ésta, relacionada con el pensamiento algorítmico. Estas representaciones gráficas pueden tomar la forma de árboles de decisión o diagramas de flujo.

Definir las reglas o pautas que deben cumplir las figuras que intervienen en un juego de adivinación de figuras (plantilla de figura), es un aspecto a tener en cuenta de la planificación didáctica y puede considerarse una variable didáctica importante, especialmente si el objetivo es la construcción de habilidades del PC y/o la representación gráfica (como por ejemplo, diagramas de flujo o árboles de decisión). Estas reglas o pautas son fundamentales para la estructura y la jugabilidad del juego y pueden influir en la forma en que los estudiantes participan y aprenden.

### **La propuesta: El juego de adivinación**

En este apartado se presenta la propuesta del juego organizada en diferentes etapas y el análisis didáctico-matemático-computacional del mismo.

### **Organización de la clase y explicación del juego**

Se divide la clase en dos o más grupos. El docente elige para cada equipo una figura de una plantilla que está a la vista y disponible para todos los jugadores y no la da a conocer.

Llamaremos *figura secreta* a cada figura elegida. El juego consiste, para cada equipo, en adivinar cuál es la *figura secreta* haciendo preguntas que se respondan por Sí o por No.

Los equipos juegan por turnos haciendo solamente una pregunta cada vez. Con cada pregunta tendrán la posibilidad de descartar una, ninguna o más figuras a partir de la respuesta recibida. En la jugada que se arriesga no se puede hacer ninguna pregunta. Si arriesga y no acierta pierde el juego. Gana el equipo que hace menos preguntas para descubrir la *figura secreta* elegida por el docente para ese equipo.

**Primera etapa:** Se forman 3 o 4 equipos. Toda la clase juega al juego de adivinación. El profesor elige una figura de la plantilla para cada equipo. Cada equipo comienza a jugar haciendo preguntas que se puedan responder por Sí o por No. El docente decide quién gana cada partida. (Esta etapa puede incluir dos o más partidas). El objetivo de esta etapa es que los estudiantes comprendan de qué se trata el juego.

**Segunda etapa:** Cada grupo se divide en dos subgrupos y juegan al juego de adivinar la figura. Uno de los integrantes del grupo es quien elige la figura secreta, responde las preguntas y decide quién gana cada partida. Jugar varias partidas, registrar las preguntas de cada jugada.

**Tercera etapa:** Momento de análisis y formulación. Luego de *jugar*, analizar las preguntas realizadas por los equipos, discutir sobre la conveniencia de unas u otras. a) Decidir en cada juego, si todos los equipos saben jugar, quién ganará el juego, es decir, explicitar las condiciones para ganar. b) En cada partida, ¿hay un único equipo ganador? ¿Por qué?

**Cuarta etapa:** Momento de discusión, validación y acuerdos. En esta etapa, coordinados por el docente se abordarán las siguientes cuestiones, entre otras:

- ¿Qué estrategias usaron para quedarse con menos figuras cada vez?
- ¿De qué manera se puede llevar el control de las figuras que se descartan o de las que quedan?
- Con esta plantilla, ¿Qué grupo tiene más posibilidades de ganar el juego? ¿Por qué?
- ¿Cuándo conviene arriesgar?
- Si un equipo adivinó la figura y ahora es momento de que juegue mi equipo, ¿qué conviene hacer?

## **Análisis didáctico-matemático-computacional del juego**

### **Fortaleza del juego. Intenciones didácticas del juego**

Desde la perspectiva del estudiante, la finalidad del juego se reduce a adivinar cuál es la figura secreta para ganar el juego, en tanto, el docente tiene una intencionalidad didáctica, esto es que los estudiantes pongan en funcionamiento estrategias de *clasificación*. Para ello, impone la restricción de las preguntas que pueden formular los jugadores, solo pueden ser preguntas dicotómicas que se respondan por Sí o No.

Por un lado, teniendo en cuenta el conjunto de figuras de la plantilla, los jugadores–estudiantes deben identificar características y/o propiedades de las figuras, analizarlas y utilizarlas para clasificar atendiendo a algún criterio vinculado, posiblemente de manera implícita, a los cuantificadores (tiene todos, tiene algún, tiene uno, tiene al menos, tiene a lo sumo, no tiene... etc). Tanto el criterio empleado para armar las *clases*, como los cuantificadores que subyacen, serán determinantes para economizar en la cantidad de preguntas y ganar el juego. En este sentido, es preciso *decidir* entre seleccionar características comunes o diferentes de las figuras de la colección, lo que redundará en preguntas más inclusivas o exclusivas de acuerdo al estado en que se encuentre el juego. Todas estas decisiones favorecen el desarrollo de capacidades lógicas que son base del PC. Además, explicitar la estrategia de clasificación, a partir de formular una pregunta dicotómica que determine si cierta figura pertenece o no a una clase, pone en funcionamiento una *alternativa condicional* lo que permite que los jugadores-estudiantes desarrollen habilidades de pensamiento crítico y por ende computacional.

Cabe mencionar que, con el propósito de disponer de un encuadre teórico, tomaremos en este trabajo resultados de las investigaciones de Castorina y Palau (1982). Al respecto se considera una definición de *clasificación* que no hace exclusiva referencia a objetos matemáticos:

Clasificación es un sistema<sup>2</sup> cuyos elementos son o bien clases<sup>3</sup> disjuntas o bien clases que guardan entre sí una relación de inclusión jerárquica. Una clase está incluida en otra cuando todo elemento que pertenece a una clase pertenece también a otra. Las clases son disjuntas cuando no tienen elementos comunes entre sí y están relacionadas mediante una relación de inclusión jerárquica si dada una clase se puede construir otra de orden superior que la incluya". (Castorina y Palau, como citado por Rosso, 2005, Cap. III)

---

<sup>2</sup> "Llamaremos *sistema* a un *conjunto de clases* relacionadas entre sí, de tal manera que cada una de las clases del sistema adquiera sentido en virtud de las relaciones que sostiene con las otras clases (Castorina y Palau, 1982)

<sup>3</sup> *Clase o conjunto* es una colección de objetos que tienen una propiedad en común. Los componentes de un conjunto reciben el nombre de elementos y la relación entre elementos y conjunto se denomina pertenencia. (Castorina y Palau, 1982)

Durante el desarrollo del juego de adivinación, cada jugada obliga a realizar una nueva clasificación a partir de observar nuevas características, estas acciones determinan una clasificación de tipo jerárquica que consideramos exige desplegar la construcción de habilidades de orden superior (Churches, 2009). Este tipo de clasificación compuesta, en general, es más compleja de ser abordada en el aula, sin embargo es una herramienta fundamental utilizada en las distintas áreas de las ciencias, en particular en las ciencias de la computación y por tanto es pertinente su abordaje en el aula desde edades tempranas. Otro aspecto del juego que vale la pena mencionar, es que no sólo se tiene que tener en cuenta una gama variada de información no disjunta sino que la misma aparece de manera simultánea en diferentes figuras, a la vez debe considerarse las preguntas ya realizadas, analizar la pertinencia y la conveniencia de realizar una u otra, etc.

### El juego de adivinación modelizado con una plantilla particular

A fin de compartir el planteo de la propuesta con este tipo de problemas se estudia un ejemplo de juego de adivinación con una plantilla constituida por imágenes de doce emojis. Esta decisión se fundamenta en que a partir del estudio de este ejemplo particular se pueda explicar de qué se tratan estos problemas para luego avanzar a cuestiones más generales de los mismos. Uno de estos aspectos está relacionado con el rol preponderante de la plantilla que, como veremos, no se limita a la cantidad de figuras sino que requiere de un estudio mucho más profundo.

La plantilla con las doce figuras fue diseñada especialmente. Cada una está determinada por un conjunto de características que las diferencian de las otras, sin embargo cabe mencionar que cada figura comparte alguna de sus características con otras. En la Figura 1 se muestra la plantilla de emojis.



Figura 1: *Plantilla de emojis.*

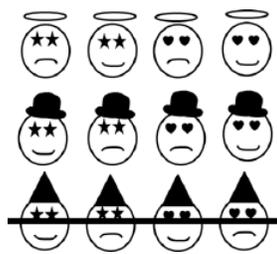


Figura 2: *Figuras que no tienen bonete.*

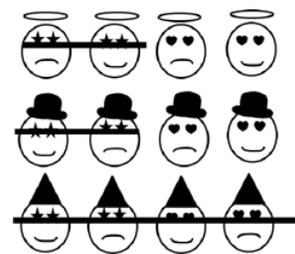


Figura 3: *Figuras que no tienen ojos con estrellas*

A partir de la plantilla de los emojis de la Figura 1, se realiza un análisis con *supuestas* jugadas de un solo grupo, el mismo puede aplicarse a todos los grupos involucrados en una misma jugada.

**Análisis de posibles decisiones de un grupo en una partida del juego de adivinación.**

Vamos a mostrar dos posibles situaciones, bien diferentes, que se pueden presentar al jugar. Esta diferencia está sujeta a una pregunta que determina una partición en dos grupos de distinta cantidad de figuras.

La pregunta: *¿Tiene bonete?* Genera una partición en dos conjuntos: uno contiene ocho figuras y el otro cuatro. Claramente hay dos posibles opciones de respuestas.

La primera opción considera la respuesta *NO* a la primera pregunta 1°: *¿Tiene bonete?* Esta respuesta descarta cuatro figuras y el conjunto donde se encuentra la figura secreta queda restringido a los ocho emojis que no tienen bonete (Figura 2).

La siguiente pregunta debe hacerse en función de las figuras que quedan. En esta plantilla, la segunda pregunta, independientemente si se refiere al tipo de ojos, al gesto de la boca o al elemento sobre la cabeza, determina dos subconjuntos con igual cantidad de figuras. Por ejemplo, 2°: *¿Tiene ojos con estrellas?*, tanto si la respuesta es *SI*, como si es *NO*, se descartan cuatro figuras más y la figura secreta se encuentra entre las cuatro restantes. Por ejemplo si la respuesta es *NO*, la plantilla se vería como en la Figura 3.

A continuación se debería hacer una 3° pregunta, por el gesto de la cara o por lo que tienen sobre su cabeza, la respuesta le permitiría descartar dos figuras más y quedarse solamente con dos. Una nueva pregunta, la 4°, dejaría en evidencia la *figura secreta*. Con lo cual en la 5° jugada están en condiciones de arriesgar y ganar el juego.

Como mencionamos, hay una segunda opción para la primera jugada. Esta nueva opción posibilita encontrar la figura secreta con una menor cantidad de preguntas y esto se da cuando se responde *afirmativamente* a la pregunta: *¿Tiene bonete?* Esta respuesta descarta ocho figuras y el conjunto donde se encuentra la figura secreta queda restringido a los cuatro emojis que tienen bonete, acelerando el proceso de descarte y la cantidad total de preguntas resulta ser una menos respecto de la primera opción.

Este análisis se hizo planteando como primera pregunta *¿Tiene bonete?*, pero sería el mismo para las preguntas *¿Tiene sombrero?* o *¿Tiene aureola de angelito?*

Otras opciones de juegos serían las derivadas de preguntas iniciales sobre el gesto de la cara o por la forma de los ojos. Para estos casos, las respuestas particionan el conjunto de figuras por la mitad, con seis figuras cada una. En principio pareciera que las posibilidades serían diferentes al analizado recientemente aunque para encontrar la figura secreta, en algún momento del juego se debe pasar por la selección de lo que tienen en la cabeza y en ese caso las posibles particiones son diferentes, así la figura secreta está dentro de un tercio o dentro de dos tercios

del subconjunto sobre el que se pregunte. De esta manera podemos concluir que cuando la plantilla es como la de este juego, la cantidad de preguntas que permite identificar la figura secreta se reduce a cuatro o cinco preguntas independientemente de la pregunta inicial.

En definitiva, con una plantilla como la analizada, tanto la figura elegida como la primera pregunta que se realice determinan quién ganará el juego, es decir adivinar la figura con la menor cantidad de preguntas. De este modo, según la modelización de juego determinada por Brousseau (1986), queda en evidencia que *la mejor estrategia posible* es el conocimiento que se pretende que se aprenda, en este juego, las características del proceso de clasificar.

**¿Qué características tiene esta plantilla?** Como se observó, la forma en que fue construida la plantilla determina la cantidad de preguntas requeridas para adivinar la figura secreta. En esta plantilla, cada figura tiene, simultáneamente, tres características, dos de esas características son compartidas entre la mitad de las figuras y la restante la comparten un tercio de las figuras. Así siempre que las plantillas se construyan de esta manera tendrán los mismos resultados, es decir, en cuatro o cinco preguntas -si se sabe jugar- se adivinará la figura. La cantidad de preguntas (que sean cuatro o que sean cinco) quedará determinada cuando se realiza la pregunta respecto a lo que tienen en la cabeza, esto es ¿tiene bonete? o, ¿tiene aureola? o, ¿tiene sombrero?. Teniendo en cuenta que el conjunto de figuras que cumple una de esas características es igual a un tercio del mismo, al hacer una pregunta referida a esa condición el conjunto se divide inequitativamente en un tercio y dos tercios respectivamente. Así, serán cuatro preguntas si la respuesta ubica a la figura secreta en el conjunto del tercio de figuras o, serán cinco preguntas si la figura se encuentra en los dos tercios.

El esquema de la Figura 4 pone de manifiesto una manera de identificar un emoji particular. En primer lugar considera el criterio vinculado al objeto que tienen en la cabeza, luego analizar el tipo de ojos y finalmente el gesto de la boca. En este sentido, la representación permite por un lado visualizar el criterio utilizado en cada nivel para realizar la clasificación. Cada figura de la plantilla de los emojis, está determinada a partir de incluir todas las características que aparecen en cada nivel del camino, desde el emoji hasta el gesto de su cara. Por otro lado, haciendo el camino inverso, es posible, partiendo de cada emoji identificar las características que lo determinan. Este análisis pone en evidencia dos aspectos del proceso de clasificar antes mencionado. En otras palabras, la habilidad de clasificar implica, por un lado, identificar y/o construir criterios de clasificación para organizar un conjunto de datos y, por otro, dada una clasificación poder analizar si tal o cual elemento pertenece o no a cierta categoría de esa clasificación.

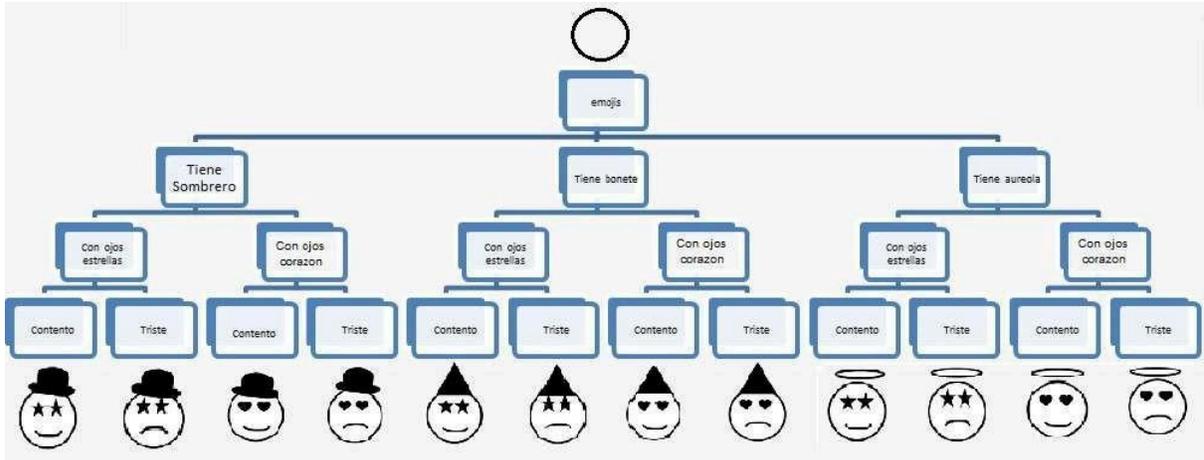


Figura 4: Esquema para identificar los emojis.

Claramente la cantidad de figuras en la plantilla, las características de las figuras forman parte de las posibles variantes del juego que nos proponemos seguir investigando y se constituyen en variables didácticas del problema que permitirá avanzar en las relaciones-conocimientos que se pretendan ir construyendo.

Siguiendo en la misma dirección de la propuesta presentada, es posible proponer otras situaciones problemáticas a posteriori del juego de adivinación, teniendo en cuenta que los jugadores-estudiantes ya se han apropiado de qué se trata el juego, podrán abordarlas de manera autónoma en nuevas etapas del juego.

En este sentido, desde la Didáctica de la Matemática se postula que toda actividad requiere de un trabajo posterior donde se reflexione sobre los conocimientos que se pusieron en funcionamiento y los que emergieron. En relación con eso, proponemos las siguientes tareas, que si bien están relacionadas con el juego, permite plantearlo desde otro lugar.

**Quinta etapa:** Ahora la Figura 5 es la plantilla de adivinación con la que están jugando.

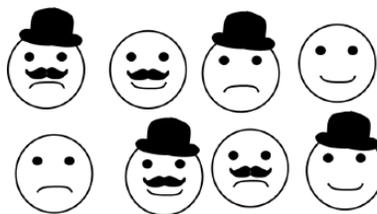


Figura 5: Plantilla de la quinta etapa.

Se plantea la siguiente consigna:

*Durante el juego estas fueron las preguntas y respuestas de un equipo para adivinar la figura secreta:*

*1º: ¿Tiene sombrero? No 2º: ¿Tiene bigotes? Si 3º: ¿Está triste? No*

*¿Es posible averiguar cuál era la figura secreta? En caso afirmativo indica cuál es la figura elegida, caso contrario explica por qué no es posible.*

La vivencia de la actividad anterior permitirá desplegar alguna estrategia de descarte para determinar si con la información que tiene es posible caracterizar la figura. Aún más, este problema puede hacer explícito el proceso de selección en una representación como la de la Figura 6.

Este tipo de tarea es potente para poner en funcionamiento el proceso de clasificar en un a posteriori de jugar. Para este trabajo el modelo que se presenta es una versión simplificada atendiendo a la plantilla que se toma.

**Sexta etapa:** *Suponiendo que quien elige la figura secreta selecciona esta figura Mirando la figura responde las preguntas SI o NO.*



*¿No tiene sombrero? ¿Tiene bigotes? ¿Está contento?*

Las respuestas a estas preguntas se pueden representar en un diagrama de flujo similar al esquema que se muestra en la Figura 6. La representación del proceso predictivo completo de esta etapa se puede observar en la Figura 7 - representado como un árbol de decisión simple -.

Tal como lo señalamos oportunamente el proceso de clasificar elementos implica dos aspectos centrales, por un lado formular los criterios para organizar los elementos en clases, y por otro, conociendo las clases determinar para cada objeto a qué clase pertenece. Si bien ambos aspectos son abordados a lo largo del juego según el rol que tengan los participantes -elegir la figura o adivinarla-, estas dos últimas situaciones pretenden recuperar expresamente esos aspectos, en particular si el trabajo lo abordan de manera individual.

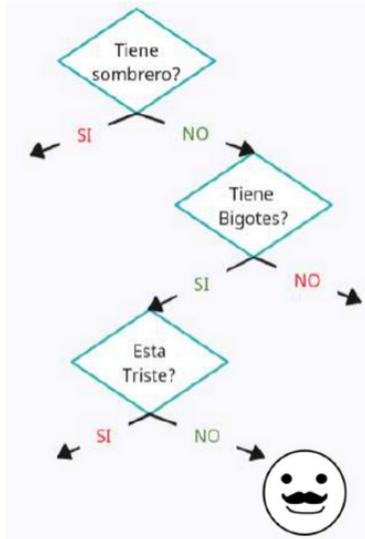


Figura 6: Proceso realizado en la Quinta etapa.

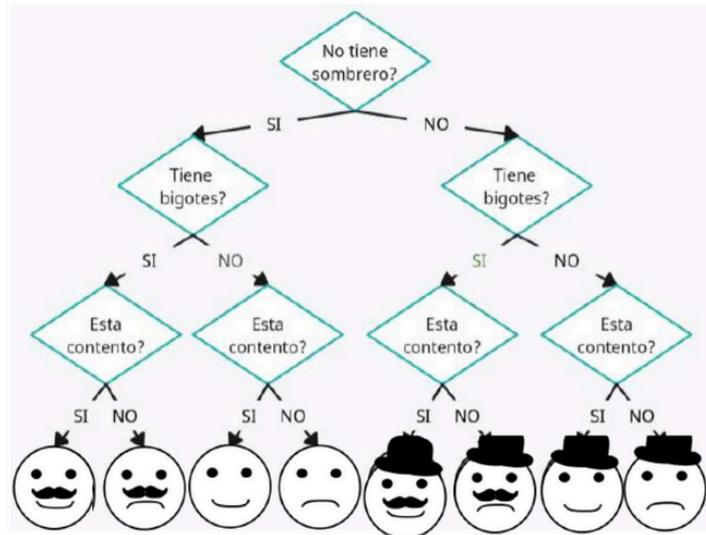


Figura 7: Proceso predictivo completo de la Sexta etapa

## Conclusiones y Trabajos Futuros

En primer lugar, en este trabajo se realiza un desarrollo teórico que fundamenta el abordaje de la clasificación en post de la construcción del PC. En segundo lugar, se resignifica la noción de juego que justifica su inclusión en situaciones para la enseñanza. Una vez explicitado el marco, se estudia un tipo de problemas que habilita el desarrollo de capacidades vinculadas a la clasificación en la escuela. El tipo de problemas involucra situaciones mediadas por el *juego de adivinación* o *juego de descubrir una figura*.

Es de destacar que este estudio pone en evidencia la potencia del juego para ser implementado en el aula. Además hace visible cómo debe ser abordado en la formación docente dado que incluye el análisis didáctico-matemático-computacional y un tratamiento respecto de cómo está confeccionada la plantilla que permite anticipar *la estrategia ganadora*.

Es de nuestro interés continuar con el estudio de la clasificación y ampliar la gama de problemas que la involucren. Claramente la cantidad de figuras en la plantilla y las características de las figuras forman parte de las posibles variantes del juego que nos proponemos seguir investigando y se constituyen en variables didácticas del problema. Se aspira analizar modificaciones del juego con plantillas equiprobables o no equiprobables. Así también, estudiar las plantillas de los juegos que existen en el mercado como *¿Quién es Quién?* Por último, se está evaluando la posibilidad de un trabajo interdisciplinario, por ejemplo, donde las figuras a clasificar sean figuras geométricas, conjuntos numéricos, especies del reino animal o vegetal, entre otros.

Actualmente se implementa la propuesta didáctica con los estudiantes-docentes que se

encuentran cursando la Diplomatura Superior en Didáctica de las Ciencias de la Computación para la Docencia, que se dicta en INSTITUCIÓN. Esta puesta en práctica, entre docentes de escuela primaria y escuela media permitirá hacer un análisis a posteriori que formará parte del análisis a priori de futuras investigaciones.

## Referencias

- Bers, M. U. (2018). *Codings as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*. New York: Routledge. <https://doi.org/10.4324/9781315398945>
- Churches, A. (2009). Taxonomía de Bloom para la era digital. Eduteka. Consultado 10/01/19. <http://eduteka.icesi.edu.co/articulos/TaxonomiaBloomDigital>.
- Brennan, K., y Resnick, M. (2012). *New Frameworks for Studying and Assessing the Development of Computational Thinking*. Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vol. 1, Vancouver, 13-17 April 2012.
- Brousseau, G. (1986). *Fundamentos y Métodos de la Didáctica de la Matemática*. Facultad de Matemática, Astronomía y Física. Universidad Nacional de Córdoba.
- Castorina, J. y Palau, G. (1982). *Introducción a la lógica operatoria de Piaget : Alcances y significado para la psicología genética*. Paidós. (Psicología del siglo XX).
- Catlin, D., y Woollard, J. (2014). *Educational robots and computational thinking*. En and others (Ed.), 4th TRtWR & RIE 2014. International workshop: Teaching robotics & teaching with robotics.
- Csizmadia, A., Dorling, M., Ng, T. y Selby, C. (2015). *Computational thinking-a guide for teachers*. Swindon. Computing at School. 18pp.
- ISTE, y CSTA. (2011). *Pensamiento computacional: caja de herramientas para líderes*. Computer Science Teachers Association (CSTA) y la International Society for Technology in Education (ISTE). [http://eduteka.icesi.edu.co/pdfdir/PensamientoComputacional\\_Definicion.pdf](http://eduteka.icesi.edu.co/pdfdir/PensamientoComputacional_Definicion.pdf)
- Panizza, M. (2003). *Enseñar matemática en el Nivel Inicial y el primer ciclo de la EGB*. Cap II Conceptos básicos de la teoría de situaciones didácticas. Editorial Paidós.
- Piaget, J. e Inhelder (1983). *Génesis de las estructuras lógicas elementales*. Editorial Guadalupe. Argentina.
- Rosso, F. (2005). *La clasificación de cuadriláteros. Razones de ser de su enseñanza*. Tesis de Maestría en Didáctica de la Matemática. UNRC.
- Sadovsky, P. (2005). *La Teoría de Situaciones Didácticas: un marco para pensar y actuar la enseñanza de la Matemática, Reflexiones teóricas para la educación matemática*. Buenos Aires, Libros El Zorzal.
- Wing, J. (2006). *Computational thinking*. CACM Viewpoint. <https://doi.org/10.1145/1118178.1118215>

Wing, J. (2008). *Computational thinking and thinking about computing*. Philosophical Transactions Of The Royal Society, 366, 3717–3725.  
<https://doi.org/10.1098/rsta.2008.0118>

Wing, J. (2011). *Computational thinking: What and Why*. Descargado de <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>

## Resultados Preliminares del Curso la Programación y su Didáctica

Francisco Bavera<sup>1,2</sup>, Teresa Quintero<sup>1</sup>, Marcela Daniele<sup>1</sup>  
{pancho, tqintero, marcela}@dc.exa.unrc.edu.ar

<sup>1</sup> Facultad de Ciencias Exactas, Físico-Químicas y Naturales  
Universidad Nacional de Río Cuarto Río Cuarto

<sup>2</sup> Universidad Nacional de Villa Mercedes

### Resumen

La enseñanza de las Ciencias de la Computación permite promover la construcción del Pensamiento Computacional (PC), capacidades y competencias útiles para la búsqueda de soluciones a diversos tipos de problemas, aportando además a la formación de ciudadanos críticos, participativos y autónomos en el uso de las tecnologías. Estudios y tendencias a nivel mundial dan cuenta de la importancia de introducir y desarrollar el PC en el sistema educativo obligatorio. Por otra parte, numerosas investigaciones, concluyen en que la mayoría de los docentes que en la actualidad enseñan en los primeros años de educación básica (en especial en países emergentes), presentan marcadas falencias en el desarrollo de habilidades asociadas al PC, como abstracción, generalización, definición de patrones, y resolución de problemas. Esto genera deficiencias significativas cuando propuestas educativas en torno al PC son implementadas en aulas de primaria, destacando la necesidad de comenzar por la formación de docentes. En este trabajo, se presenta un análisis sobre el impacto de un curso de didáctica de la programación, destinado a docentes de primaria y secundaria, en la resolución de problemas y la planificación de actividades que involucran habilidades del PC.

**Palabras Claves:** Pensamiento Computacional, Formación Docente Continua, Didáctica de las Ciencias de la Computación, Evaluación

### Introducción

La construcción del Pensamiento Computacional (PC) se manifiesta como un importante desafío para los formadores actuales de todos los niveles educativos. Además se sabe que el desarrollo de ciertas habilidades, tanto cognitivas como actitudinales, son claves en la primera infancia. En la actualidad el uso de las tecnologías constituye un importante elemento que ayuda a plasmar la solución de diferentes tipos problemas en los más diversos ámbitos y contextos. Construir Pensamiento Computacional permite aprender y comprender el desarrollo y diseño de las tecnologías, incluyendo estrategias de descomposición para la resolución de problemas, abstracción, generalización de las soluciones encontradas, la aplicación de procesos que definen

pasos racionales y sistémicos hacia la búsqueda de soluciones, logrando la definición y el diseño de algoritmos que permiten encontrar las soluciones o resultados deseados (Aho, 2012; Denning, 2017; Wing, 2006). La construcción del pensamiento computacional posibilita el desarrollo de ciertas habilidades de orden superior que aportan y enriquecen la manera de comprender, resolver problemas y automatizar dichas soluciones a partir del conocimiento de principios y conceptos fundamentales de las Ciencias de la Computación, como lógica, abstracción, representación de datos y diseño de algoritmos.

Caracterizar la construcción del Pensamiento Computacional, en el marco de la formación docente continua y el impacto sobre sus propias prácticas, es uno de los aspectos en los que están direccionados los esfuerzos de los autores del presente trabajo. Entre los aspectos que se abordan se encuentra el análisis de las prácticas de enseñanza de docentes de nivel primario y secundario que participan en procesos de formación en didáctica de las Ciencias de la Computación y el desarrollo del PC. Se vislumbra la necesidad de profundizar y consolidar investigaciones que aborden al estudio del impacto de la construcción del PC en los formadores. Para esto es necesario evaluar, por un lado, las actividades de formación propuestas, y por otro lado, las habilidades construidas por los educadores que transitan una formación continua en didáctica de las Ciencias de la Computación. En función de la complejidad de esta tarea se hace ineludible el uso de diferentes enfoques, métodos y técnicas que permitan obtener evidencia e indicadores para analizar y evaluar el impacto de estas formaciones.

Los autores de este trabajo forman parte de un equipo multidisciplinar que coordina y dicta distintas instancias de formación docente continua *en Didáctica de las Ciencias de la Computación*. En este marco se llevan a cabo diferentes estudios con el objeto de acercarse a la complejidad inherente del proceso de construcción del PC en estos espacios de capacitación.

En investigaciones previas se abordó el estudio de distintas producciones de los cursantes de distintas instancias de capacitación dictadas. Se analizaron estas producciones usando una metodología cualitativa los datos fueron triangulados entre sí por al menos dos investigadores considerando las categorías de la Taxonomía de Bloom adecuada por Churches (Churches, 1996). Estos análisis permitieron determinar que todas las producciones y las propuestas de prácticas áulicas, reflejaron ciertas habilidades cognitivas relacionadas al mundo digital, mostrando diferentes niveles de desarrollo y profundidad (Daniele, 2019).

Posteriormente se realizó un análisis de las habilidades de PC, tales como, formular problemas donde se usen computadoras y otras herramientas para solucionarlos, organizar datos de manera lógica y analizarlos, representar datos mediante abstracciones, las que son mayormente

abordadas en los módulos específicos de enseñanza de la programación. Este análisis brindó importantes evidencias de que la formación recibida por estos educadores impacta positivamente en la transformación de sus prácticas áulicas (Bavera, 2019).

Luego se analizó las habilidades de Pensamiento Computacional en docentes de primaria que cursaron una *Especialización Docente de Nivel Superior en Didáctica de las Ciencias de la Computación*. Esto permitió detectar que un alto porcentaje de los docentes pudieron resolver problemas de complejidad media que implican el uso de habilidades del PC, tales como, abstracción, reconocimiento de patrones, modelos y simulación, algoritmos y descomposición (Bavera, 2020).

La evaluación del pensamiento computacional está en desarrollo por lo que los educadores encuentran considerables limitaciones al momento de evaluar estas habilidades que son consideradas centrales para la formación de los estudiantes del mundo actual (Lockwood, 2018). En este contexto se encuentra que los Problemas de Bebras (Iniciativa Bebras), que están especialmente diseñados para evaluar habilidades de pensamiento computacional, resultan un instrumento adecuado para medir estas habilidades antes y después de una capacitación. Bebras es una competencia internacional que tiene como objetivo promover el pensamiento computacional entre estudiantes de todas las edades con actividades de tipo “desconectadas”. Se han realizado estudios que analizan la efectividad de los problemas de Bebras y también comparaciones con otras pruebas de pensamiento computacional (Lockwood, 2018; Gouws, 2013; Dagiene, 2016; Hubwieser, 2014; Hubwieser, 2014; Vaicek, 2014). No obstante no se encuentran registros de resultados obtenidos a partir de la utilización de Problemas Bebras en la evaluación de la formación docente continua en PC por lo cual resulta interesante indagar sobre los alcances de esta alternativa de evaluación.

En este trabajo se muestran los resultados obtenidos del análisis de una experiencia, basada en actividades seleccionadas de problemas de Bebras, donde se evalúan las habilidades de PC de los docentes de educación primaria y secundaria que participaron del curso *La Programación y su Didáctica I* aportando otra forma de evaluación de las mismas.

Este trabajo está organizado de la siguiente forma: la sección 2 expone la experiencia de evaluación realizada a partir de problemas Bebras, se describe la población de individuos que se evalúan y el test realizado. En la sección 3 se presenta un análisis de los resultados obtenidos, y por último se exponen algunas conclusiones y los posibles trabajos futuros que dan continuidad a esta investigación.

## **La Experiencia**

## La Formación

En esta experiencia se realiza una evaluación de las habilidades relacionadas con el PC que presentan los docentes nivel primario y secundario que realizaron el curso *La Programación y su Didáctica I*<sup>1</sup>. Este curso se dictó el primer semestre de 2021 de manera virtual. Con instancias sincrónicas y actividades autoasistidas asincrónicas y una carga horaria total de 100 hs.

La Programación y su Didáctica I es un curso ofrecido por la iniciativa Program.ar, una iniciativa Fundación Sadosky, dependiente del Ministerio de Ciencia y Tecnología, que tiene como objetivo promover la inclusión de las Ciencias de la Computación a la escuela argentina. Este curso está dirigido a docentes de educación primaria y secundaria que desean incorporar la enseñanza de la programación en sus clases. Se enfoca en la didáctica de la programación, abordando temas como la planificación y diseño de actividades, la selección de herramientas y recursos pedagógicos, la evaluación de los aprendizajes y la inclusión de la programación en distintas áreas del conocimiento. Entre los contenidos que se abordan en este curso se encuentran:

- Conceptos básicos de programación y lógica computacional.
- Herramientas y recursos para la enseñanza de la programación.
- Diseño de actividades y proyectos de programación.
- Evaluación de los aprendizajes en programación.
- Integración de la programación en distintas áreas del conocimiento.
- Inclusión y accesibilidad en la enseñanza de la programación.

Al finalizar el curso se espera que los participantes hayan adquirido los conocimientos y habilidades necesarias para diseñar y llevar a cabo actividades y proyectos de programación en el ámbito educativo y estén preparados para incorporar la programación en su práctica docente de manera efectiva y significativa. Cabe mencionar que el curso es una propuesta innovadora dado que el objetivo es transmitir conceptos básicos y generales sobre programación y donde los conceptos técnicos están al mismo nivel de importancia que la didáctica. El enfoque didáctico se basa en el aprendizaje por indagación (Dostál, 2015), una metodología de enseñanza-aprendizaje donde los estudiantes deben encontrar soluciones a un problema a partir de un proceso de exploración y reflexión sobre las actividades llevadas a cabo para construir la

---

<sup>1</sup> Se puede consultar una descripción del curso en [https://docs.google.com/document/d/1C00ZEtxLnj1apaX0bSpQwlqKGStqY\\_Elx71e0UAbDdg/edit?usp=sharing](https://docs.google.com/document/d/1C00ZEtxLnj1apaX0bSpQwlqKGStqY_Elx71e0UAbDdg/edit?usp=sharing)

solución.

Las herramientas conceptuales consideradas son tres y conforman, junto con la didáctica propuesta, el punto central de lo que se pretende transmitir con este curso tal como expresa Martínez Lopez (2017):

- La noción de *estrategia de solución*, y la de su explicitación para aplicar la metodología de división en subtarear.
- La noción de *legibilidad*. Los programas son fundamentalmente un medio de comunicación entre personas, además de servir como vehículo para hacer funcionar máquinas, capturado en la importancia de que los programas sean legibles (o sea, claramente entendibles por otros programadores) y esto a su vez expresado mediante la metodología de elección de nombres adecuados para cada una de las partes de un programa que se escribe.
- La noción de *algorítmica* básica, expresada, por ejemplo, en la noción de recorrido.

Las estrategias de solución, la correcta comunicación de soluciones a problemas, y mecanismos elementales de solución de problemas sencillos habilitan y fomentan la formación de capacidades de pensamiento computacional (Aho, 2012; Denning, 2017; Anderson, 2001). Por ejemplo, al llenar la mochila antes de ir al colegio; al recorrer los lugares previamente visitados buscando un objeto perdido; al confeccionar una lista de actividades a realizar para organizar una fiesta; entre muchos otros posibles ejemplos de la vida diaria, son actividades que involucran habilidades del pensamiento computacional que requieren tomar decisiones, resolver problemas, buscar estrategias, aplicar pensamiento lógico y algorítmico.

### **Contexto y Participantes**

Participaron del curso 60 docentes. De estos 60 docentes, el 90% asistieron a más del 80% de las actividades sincrónicas y realizaron el 100% de las actividades solicitadas. El 10% restante asistieron entre el 55% y el 75% de los encuentros sincrónicos y realizaron entre el 70% y el 100% de las actividades solicitadas. Cabe destacar que el 95% realizaron todas las actividades en el aula virtual y el 5% restante realizaron más de  $\frac{2}{3}$  de las mismas.

Son docentes de nivel primario (45%) y secundario (55%). En cuanto al espacio curricular en que se desempeñan, el 20% corresponde a Matemática, 20% Tecnología, sólo el 3% al área de Informática, el 35% son maestras de grado y el resto participa en otros espacios curriculares (Biología, Inglés, Lengua, entre otros). La *media* de edad es 37 años, en un rango que va desde los 21 a los 54 años. Tenemos una *mediana* y una *moda* de 38 y 49 años, respectivamente. El

65% son docentes titulares y los restantes son interinos y suplentes, que se desempeñan en diferentes grados de la escuela primaria y cursos de la escuela secundaria. Los docentes pertenecen a 47 escuelas, donde el 80% son instituciones públicas de gestión estatal y el restante 20% instituciones públicas de gestión privada. Un 51% de estas escuelas están localizadas en la ciudad de Río Cuarto, un 25% en localidades de su región de influencia (en un radio de 30 a 150 km de distancia de Río Cuarto) y las restantes pertenecen a localidades de la provincia de Córdoba más distantes de Río Cuarto.

### **El Desafío Bebras**

Como se mencionó anteriormente, Bebras (Bebras) es una competencia internacional que tiene como objetivo promover el pensamiento computacional, entre estudiantes de todas las edades, con actividades de tipo “desconectadas”. La iniciativa Bebras es parte de una propuesta internacional, abierta y basada en la colaboración entre docentes. La competencia se lleva a cabo anualmente en más de 50 países y consta de una serie de problemas y desafíos en los que los participantes deben aplicar su habilidad en pensamiento lógico, resolución de problemas y programación para encontrar soluciones. Los participantes generalmente son supervisados por maestros y el desafío se realiza en las escuelas usando computadoras o dispositivos móviles. Los problemas o desafíos no requieren conocimientos previos de programación o de Ciencias de la Computación, pero todos abordan estas temáticas, por ejemplo patrones, codificación, criptografía, árboles, entre otros.

Los problemas planteados en el Desafío Bebras son multidisciplinarios, lo que significa que se basan en situaciones del mundo real y pueden requerir conocimientos de matemáticas, ciencias naturales, tecnología y otros campos de estudio. De esta manera, los participantes tienen la oportunidad de desarrollar habilidades en distintas áreas del conocimiento mientras aprenden a aplicarlas en la resolución de problemas informáticos.

El Desafío Bebras es una oportunidad para que los estudiantes desarrollen habilidades en informática y pensamiento computacional, lo que puede ser beneficioso para su futuro académico y profesional en un mundo cada vez más tecnológico. Además, la competencia también puede ayudar a fomentar la colaboración y el trabajo en equipo entre estudiantes y docentes, y a generar interés en la informática en la comunidad educativa en general.

Los resultados muestran que el Desafío Bebras ha tenido un impacto significativo en la promoción del pensamiento computacional y la alfabetización digital en estudiantes de todo el mundo, y ha sido reconocido como una herramienta efectiva para fomentar la colaboración y el aprendizaje autónomo en el aula (Lockwood, 2018; Gouws, 2013; Dagiene, 2016; Hubwieser,

2014; Hubwieser, 2014; Vaicek, 2014).

### **La Metodología**

El enfoque del presente estudio es de tipo mixto, cuantitativo y cualitativo, donde los efectos en el aprendizaje que se van a observar y analizar se estudian una vez que ya han ocurrido (Bernardo, 2000). El estudio realizado utiliza un pre-test y un post-test sin grupo de control que posibilita analizar las habilidades del Pensamiento Computacional de los docentes. El muestreo es no probabilístico, el estudio se realiza con el grupo de docentes que realizaron el curso, con el objeto de monitorear la formación recibida a lo largo de su dictado.

El test utilizado está basado en los *Problemas Bebras*. Para construir el instrumento de evaluación, se seleccionaron y tradujeron al español problemas de complejidad media del *Bebras Australia Computational Thinking Challenge 2018* (Bebras, 2018). Los problemas seleccionados plantean actividades que incluyen encontrar caminos -un camino y el mejor camino-, reconocer patrones, identificar y aplicar algoritmos, ordenar secuencias y utilizar razonamiento lógico.

En el siguiente vínculo se encuentra disponible el pre-test completo,

[https://docs.google.com/forms/d/1KvYKJUoduHUcGkjgbguDPs2TK6cHcQUMiuVx9EFQE\\_M/](https://docs.google.com/forms/d/1KvYKJUoduHUcGkjgbguDPs2TK6cHcQUMiuVx9EFQE_M/).

Y en este otro link el post-test completo,

[https://docs.google.com/forms/d/11NMapxjVB0JtFrQWpcWx1nykKWTi7Z0tx8\\_MxhHb8u0/](https://docs.google.com/forms/d/11NMapxjVB0JtFrQWpcWx1nykKWTi7Z0tx8_MxhHb8u0/).

En la Tabla 1 se listan los problemas seleccionados y las habilidades del PC involucradas en los mismos, las cuales son:

- Descomposición: dividir los problemas en partes (sub-problemas).
- Reconocimiento de patrones: analizar los datos y buscar patrones para que cobren sentido dichos datos.
- Abstracción: eliminar detalles innecesarios y concentrar la atención en los datos importantes.
- Modelado y simulación: crear modelos o simulaciones para representar procesos.
- Algoritmos: crear una serie de pasos ordenados para resolver un problema.

Las actividades fueron presentadas a los docentes en un encuentro sincrónico, para ser resueltas individualmente en 40 minutos. Es importante resaltar que los docentes se enfrentaron por primera vez a un desafío con problemas en este formato. Es decir, las restantes actividades del curso no utilizaron *Problemas Bebras*, pero si se desarrollaron otros tipos de actividades que

implican las habilidades incluidas en los problemas seleccionados.

Como se mencionó anteriormente la evaluación se dividió en dos instancias. Los doce primeros problemas corresponden al primer test y los otros 11 restantes al segundo test. Los resultados de los tests se analizaron de forma cuantitativa en torno a las habilidades asociadas según la identificación realizada en Bebras. Además del análisis cuantitativo, se abordó un análisis cualitativo a partir de la explicitación por escrito de las estrategias utilizadas en la resolución de los problemas por parte de los docentes evaluados. Para el estudio de los textos producidos por los docentes se utilizó el análisis de contenido (Bardin, 1996), los datos fueron triangulados entre sí por al menos dos investigadores.

Habilidad PC	Descomposición	Reconocimiento Patrones	Abstracción	Modelado y Simulación	Algoritmos
Problema					
<b>Test 1</b>					
1 Ordenar Estantes	x	x			x
2 Código Secreto		x			x
3 Pociones Mágicas		x	x		
4 Castores a la Moda		x		x	x
5 Lavadero			x	x	x
6 Torneo de Castores	x		x		x
7 Ciudad Rotonda	x		x	x	x
8 Ferrocarril	x		x	x	x
9 Caramelos	x		x		x
10 Invitados a la Fiesta	x	x	x		x
11 Red Social			x	x	x
12 Torre de Dados	x	x			
<b>Test 2</b>					
1 Clave Numérica	x		x	x	x
2 Tazones		x	x		x
3 Aspiradora Inteligente		x	x		
4 Canguro Saltarín	x		x		x
5 Creando Números	x	x	x		x
6 Autos Eléctricos	x		x	x	
7 Dibujos de Rahul		x			x
8 Castor en la Feria	x	x	x		x
9 Guirnalda de Colores	x	x			x
10 Sistema de Tubos	x		x		x
11 Direcciones Concurrentes	x		x		x

Tabla 1. Problemas seleccionados y habilidades del PC asociadas.

Por último, los docentes tuvieron que realizar una planificación de una actividad áulica para introducir alguno de los temas de programación visto en el curso a sus alumnos. Esta actividad

fue realizada en grupos integrados por dos o tres docentes. Estos grupos fueron conformados por los mismos docentes cursantes. Para este último punto, cada grupo, debía seleccionar uno de los problemas del post-test como eje motivador de la planificación propuesta. En el siguiente link se puede consultar la plantilla utilizada para realizar la actividad:

[https://docs.google.com/document/d/17HHndUqrl3414qUmIF7Ga73v\\_QC6Emwkb7iUMpUOzyU/edit?usp=sharing](https://docs.google.com/document/d/17HHndUqrl3414qUmIF7Ga73v_QC6Emwkb7iUMpUOzyU/edit?usp=sharing).

## Resultados

Se encontró que los docentes respondieron correctamente en porcentajes similares en ambos tests. Aproximadamente un 75% del total de los problemas de los tests tuvieron respuestas correctas. Cabe destacar que la *media* está levemente por debajo de la *mediana* y la *moda*. Aproximadamente, un tercio de los docentes está por encima de la media y una cantidad similar por debajo de la misma. En la Tabla 2 se puede apreciar información de las respuestas correctas discriminadas por test.

Test 1				Test 2			
Respuestas Correctas	Media	Moda	Mediana	Respuestas Correctas	Media	Moda	Mediana
75%	9/12	11/12	9/12	76%	8/11	10/11	9/11

Tabla 2. Respuestas Correctas.

En la Tabla 3 se puede apreciar el porcentaje de respuestas correctas discriminadas por problema.

Test 1		Test 2	
Problema	Correctas	Problema	Correctas
1 Ordenar Estantes	94%	1 Clave Numérica	58%
2 Código Secreto	82%	2 Tazones	77%
3 Pociones Mágicas	87%	3 Aspiradora Inteligente	85%
4 Castores a la Moda	40%	4 Canguro Saltarín	90%
5 Lavadero	69%	5 Creando Números	69%
6 Torneo de Castores	91%	6 Autos Eléctricos	77%
7 Ciudad Rotonda	80%	7 Dibujos de Rahul	100%
8 Ferrocarril	68%	8 Castor en la Feria	87%
9 Caramelos	50%	9 Guirnalda de Colores	52%
10 Invitados a la Fiesta	82%	10 Sistema de Tubos	69%
11 Red Social	88%	11 Direcciones Concurrentes	69%
12 Torre de Dados	68%		

Tabla 3. Porcentaje Respuestas Correctas.

Los mejores resultados se observaron en los problemas asociados a *encontrar un camino*. Aunque, se observa una leve disminución en la efectividad de resolución, en las actividades

donde se debía encontrar *el mejor camino*. Es necesario indagar en las causas que motivan esta diferencia, que pueden ser numerosas y diversas, pero una de las posibilidades es que la mayoría de las respuestas incorrectas pueden deberse a una interpretación errónea de las consignas, a no identificar un dato relevante en el enunciado, o a dificultades relacionadas con las habilidades de abstracción.

Se encontraron tres casos que presentan gran cantidad de respuestas incorrectas. No pudiendo, con la información disponible, dar cuenta de los factores que llevan a estos resultados. Considerando que las causas pueden ser múltiples, tales como, interpretación errónea de los enunciados, falta de comprensión de las consignas, o hasta desgano o falta de interés. Es necesario indagar con mayor profundidad para poder extraer conclusiones fundamentadas sobre estos casos.

Los resultados del test, mostrados en Tabla 4, arrojaron que los docentes resolvieron correctamente el 75% (aproximadamente) de los problemas que involucran habilidades relacionadas con *descomposición, reconocimiento de patrones, abstracción y algoritmos*. Siendo muy similares en ambos tests, los porcentajes difieren en menos del 5%. Lo que permite afirmar que el curso no contribuyó a desarrollar/fortalecer dichas habilidades en los docentes cursantes.

En cuanto a *modelado y simulación*, se puede observar que los docentes respondieron considerablemente mejor en el post-test. Lo que podría indicar que las actividades del curso contribuyeron a mejorar esta habilidad.

Respuestas correctas	Test 1				Test 2			
	Habilidad PC	Rtas. Correctas	Media	Mediana	Moda	Rtas. Correctas	Media	Mediana
Descomposición	77%	5/7	6/7	6/7	73%	6/8	7/8	6/8
Reconocimiento Patrones	76%	5/6	5/6	5/6	78%	5/6	6/6	5/6
Abstracción	76%	6/8	8/8	7/8	76%	7/9	8/9	7/9
Modelado y Simulación	50%	2/5	3/5	3/5	67%	1/2	1/2	1/2
Algoritmos	74%	7/10	9/10	8/10	75%	7/9	8/9	7/9

**Tabla 4.** Respuestas Correctas por Habilidad del PC.

En relación a las descripciones del procedimiento que realizaron los docentes para resolver los problemas, en los escritos realizados en el pre-test se observó que en su gran mayoría presentaron serias dificultades para expresar por escrito en un texto descriptivo el procedimiento. En general, solo nombran una lista de procedimientos, por ejemplo, textualmente expresan: *“observé y descarte”, “leí, razoné y reconocí el patrón”, “observación, reemplazando patrones, reconocimiento”, “por descarte”, “comparé”, “cantidad de 4”, “sume 14+3”, “despejar las opciones”,* entre otros. Solo cinco docentes escribieron un texto descriptivo y explicativo, en el cual consignaron como procedieron paso por paso para resolver las actividades.

En cuanto a los resultados obtenidos en el post-test, se puede visualizar un gran contraste con lo expresado anteriormente. Más de la mitad de los docentes mejoraron la descripción y explicación de los procedimientos utilizados en la resolución de los problemas. Por ejemplo, expresan:

- *“Identifique el patrón y pude establecer que después de 3 A sigue una C”,*
- *“Mirando la forma de las hojas, luego de las hojas con forma de "pino" aspira la que tiene forma de "triángulo",”*
- *“Porque primero está la hoja más larga, después la hoja en forma de corazón, después la hoja con forma de estrella, después la hoja en forma de triángulo, y después empieza de nuevo”,*
- *“Observando el orden de la secuencia. Después de tres hojas tipo A, sigue una hoja C*
- *“le asigné letras a cada hoja y observé la serie(AAA DD BB CC AAA DD BB etc),”*
- *“Observando el orden de la secuencia, después de tres hojas A siguen dos hojas*

*C. La secuencia es DDAAACCBB, como las últimas tres que ingresaron son AAA entonces sigue C”,*

- *“excepto la figura c , todas las figuras tienen una línea que finalizan en más de un punto debiendo levantar el lápiz . En la figura C se comienza dibujando un triángulo luego la línea vertical para dibujar el segundo triángulo, que en un solo trazo se realiza.”,*
- *“En los puntos A y B se requiere pasar por los mismos puntos, en el punto D hay dos dibujos que no se conectan entre sí. En el caso de la figura C si se comienza por el vértice opuesto a la base del triángulo de arriba y se lo dibuja en su totalidad quedamos en el mismo punto de inicio para realizar una recta hacia el triángulo de abajo, que se realiza de igual forma al primero.”.*

Esto conduce a pensar que las actividades propuestas por el curso, el enfoque didáctico,

sumado a las herramientas conceptuales en las cuales focaliza, contribuyen a desarrollar habilidades para la explicitación por escrito de las resoluciones, que hacen posible tomar conciencia sobre los procedimientos y habilidades utilizadas.

Por último, los docentes realizaron 23 planificaciones de actividades, para las cuáles seleccionaron uno de los problemas de los tests como base para dicha actividad. Los docentes tuvieron mayor preferencia por los problemas *Canguro Saltarín* (7 grupos) y *Caramelos* (5 grupos). Ambos son problemas en los que hay que determinar el mejor recorrido e involucran habilidades de *descomposición*, *abstracción* y *algoritmos*. Luego siguen en nivel de preferencia los problemas asociados principalmente al *reconocimiento de patrones*. Estos son *Aspiradora* (4 grupos), *Dibujos de Rahul* (3 grupos) y *Guirnalda de Colores* (1 grupo). Además un grupo seleccionó *Clave Numérica* y otro grupo *Autos Eléctricos*, problemas que están relacionados con *descomposición*, *abstracción* y *algoritmos*.

Cabe destacar que pese a basarse en estas actividades, solo cinco de estas planificaciones trabajan explícitamente y focalizan en habilidades del PC (dos en abstracción, 4 en reconocimiento de patrones, 5 en algoritmos). Además, cuatro de estas cinco propuestas áulicas trabajan sobre la generalización del problema y la generalización de la solución.

El resto de los trabajos se focaliza en la introducción de conceptos de programación o en la mera resolución de los problemas concretos.

Es importante resaltar que entre 8 y 11 grupos centraron sus actividades en explicitar la *estrategia de resolución* y desarrollar *soluciones legibles*. Como así también 14 grupos incluyeron en sus producciones actividades que implican *nociones de algorítmica básica*. Cabe destacar que estos últimos tres conceptos mencionados son fuertemente trabajados en el curso *La Programación y su Didáctica I* y forman parte de las herramientas conceptuales más relevantes del mismo.

### **Conclusiones y Trabajos Futuros**

A partir del análisis de las habilidades de Pensamiento Computacional en los docentes participantes se detecta que un alto porcentaje pudieron resolver problemas de complejidad media que implican el uso de habilidades del PC, tales como, abstracción, reconocimiento de patrones, modelos y simulación, algoritmos y descomposición.

No obstante, los resultados obtenidos no generaron información concluyente que permitan aseverar si el curso contribuyó, o no, a fortalecer las habilidades de los docentes relacionadas al PC evaluadas con las actividades Bebras. Dado que, los resultados del pre-test y post-test son

similares en cuanto a los problemas y las habilidades asociadas a ellos que pudieron resolver correctamente los docentes.

Pero es importante remarcar que si se generó evidencia en cuanto a que el curso aporta considerablemente a mejorar las capacidades de los docentes en la explicitación de estrategias y la descripción de soluciones en la resolución de problemas. Esto se ve reflejado en las descripciones de las soluciones dadas por los docentes a los problemas. Esto, a criterio de los autores, se debe al gran énfasis que tiene el curso en las tres herramientas conceptuales consideradas (estrategias de resolución, legibilidad y algorítmica básica). Se corrobora que las estrategias de solución, la correcta comunicación de soluciones a problemas, y mecanismos elementales de solución de problemas sencillos habilitan y fomentan la formación de capacidades de PC.

El propósito de estas investigaciones es contribuir al avance del conocimiento sobre la formación continua de docentes de diferentes niveles educativos en ciencias de la computación. Además, este trabajo proporciona datos y conocimientos para el debate actual en nuestro país sobre la inclusión de ciencias de la computación y programación en el plan de estudios de la educación obligatoria.

Es necesario complementar estos resultados con otros métodos de tipo cualitativo para obtener una visión más amplia. En particular, se realizarán entrevistas a los docentes sobre los procesos involucrados en la resolución de los problemas. Como así también, se realizarán nuevas evaluaciones usando Problemas de Bebras de mayor complejidad complementando y profundizando el seguimiento de las habilidades del PC en las capacitaciones dictadas. Además se pretende analizar cómo utilizan los docentes este tipo de problemas en sus prácticas áulicas y que aspectos del PC y las ciencias de la computación trabajan con sus estudiantes.

Se considera importante continuar evaluando las habilidades de pensamiento computacional de los docentes utilizando otras metodologías y analizando otras variables y sus relaciones, así como también, evaluar cómo la formación en este tema afecta a las prácticas docentes.

## Referencias

- Aho, A. (2012). Computation and Computational Thinking. *The Computer Journal*, 55(7), 832-835. doi: <https://doi.org/10.1093/comjnl/bxs074>.
- Anderson, L., Krathwohl, D. (editors). (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Allyn and Bacon. ISBN 978-0-8013-1903-7.

- Bardin, L. (1996). *Análisis de contenido*. 2ª e. Madrid, España: Akal.
- Bavera, F., Quintero T., Daniele, M., Buffarini, F. (2019). *Análisis de prácticas de docentes de educación primaria en el marco de una formación en pensamiento computacional*. Aprobado para las 48° Jornadas Argentinas de Informática, SAEI-JAIIO.
- Bavera, F., Quintero, T., Daniele, M., Buffarini F. (2020). *Computational Thinking Skills in Primary Teachers: Evaluation Using Bebras*. Communications in Computer and Information Science book series (CCIS, volume 1184). Springer International Publishing. Computer Science – CACIC 2019. 25th Argentine Congress of Computer Science, CACIC 2019, Revised Selected Papers.
- Bebras Australia Computational Thinking Challenge. (2018). Disponible en <https://www.bebbras.edu.au/wp-content/uploads/2019/02/Bebras-2018-Solution-Guide.pdf>
- Bernado, J., Calderero, J. (2000). *Aprendo a investigar en educación*. Ediciones Rialp, Madrid, España.
- Churches, A. (2009). *Taxonomía de Bloom para la era digital*. Eduteka. Consultado 10/01/19. Disponible en: <http://eduteka.icesi.edu.co/articulos/TaxonomiaBloomDigital>.
- Daniele, M., Quintero T., Bavera F., Buffarini F., Solivellas D., De Dominici C. (2019.) *Análisis de producciones de docentes de educación primaria con formación en didáctica de las ciencias de la computación*. Segundas Jornadas de Didáctica de la Programación, Universidad Nacional de Córdoba.
- Denning, P. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33-39. doi: <https://doi.org/10.1145/2998438>.
- Dagiene V., Stupuriene, G. (2016). In *Bebras-a sustainable community building model for the concept based learning of informatics and computational thinking*. Informatics in education.
- Dostál, J. (2015). *Inquiry-based instruction: Concept, essence, importance and contribution*. PhD thesis, Palacký University, Olomouc, Czech Republic, 2015. ISBN 978-80-244-4507-6, doi 10.5507/pdf.15.24445076.
- Gouws, L., Bradshaw, K., Wentworth, P. (2013). In *First year student performance in a test for computational thinking*. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*. ACM.
- Hubwieser P., Mhling, A. (2014). In *Playing PISA with bebras*. 9th workshop in primary and secondary computing education (wipsce).
- Hubwieser, P., Mhling, A. (2015). In *Investigating the psychometric structure of Bebras contest: towards measuring computational thinking skills*. Learning and teaching in computing and engineering (latice).
- Iniciativa Bebras. <https://www.bebbras.org/>

- Lockwood, J., Mooney, A. (2018). Developing a Computational Thinking Test using Bebras problems. Proceedings of the CC-TEL 2018 and TACKLE 2018.
- Martínez López, P. (2017). Sugerencias para el dictado del curso La programación y su didáctica. Método Program.AR. Complemento al cuaderno Actividades para aprender a Program.AR de la Fundación Sadosky. Universidad Nacional de Quilmes.
- Vaicek, J. (2014). Bebras informatics contest: criteria for good tasks revised. In International Conference on Informatics in Schools: Situation, Evolution, and Perspectives. Springer.
- Wing, J. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35. doi: <https://doi.org/10.1145/1118178.1118215>.

## Una primera aproximación al contexto de implementación en Jujuy de los espacios curriculares de “Robótica y Programación”

Villarrubia, María Fernanda <sup>1</sup>, Echeveste, María Emilia <sup>2,3</sup>  
mvillarrubia@fi.unju.edu.ar, emilia.echeveste@unc.edu.ar

<sup>1</sup> Facultad de Ingeniería  
Universidad Nacional de Jujuy

<sup>2</sup> Facultad de Matemática, Astronomía, Física y Computación  
Universidad Nacional de Córdoba

<sup>3</sup> CONICET

### Resumen

El Sistema Educativo argentino enmarca un cambio de paradigma en relación a la cultura digital que genera propuestas pedagógicas para que los y las estudiantes participen activamente en la sociedad presente y futura profundizando en el ejercicio de la ciudadanía democrática, y el desarrollo económico y social de la Nación. Jujuy no es ajena a estas transformaciones, entre 2018-2019 se pusieron en práctica los Diseños Curriculares en los niveles obligatorios, incorporando la educación digital como parte de los saberes transversales en los nuevos planes de estudio. Con respecto al nivel secundario se aprueba el Diseño Curricular para el Ciclo Básico de Educación Secundaria mediante la Resolución 11134 -E-2018 y el Diseño Curricular para la Formación General y Específica del Ciclo Orientado correspondiente a la orientación en “Robótica y Programación” en el Tomo 16 de la Resolución 182 -E-2019.

En línea con las reglamentaciones mencionadas, el Marco de Organización de los Aprendizajes para la Educación Obligatoria Argentina (MOA) propone avanzar hacia una organización institucional y pedagógica que incorpore instancias de aprendizaje interdisciplinario que integren dos o tres disciplinas. En este sentido resulta interesante observar la organización que asume esta propuesta en las escuelas, por lo que proponemos analizar cómo la Orientación en Robótica y Programación se ha incorporado en escuelas de la provincia de Jujuy.

¿Cuáles son las características del contexto institucional en el que se implementan? ¿Cuáles son las decisiones metodológicas tomadas en su implementación?

¿Cómo se reconfiguran las estructuras curriculares en la provincia? ¿Cómo se desarrolla el PC (pensamiento computacional) a partir de la implementación de los mismos? ¿Qué formas de trabajo integrador evidencian la interdisciplinariedad e integración de disciplinas en los Espacios Curriculares Articulados (ECA) de Robótica y Programación? ¿Cómo se caracteriza la construcción del trabajo entre docentes y estudiantes? ¿Cómo se evalúa la implementación de la integración de los ECA de Robótica y Programación? Estos interrogantes conforman el horizonte

de este proyecto de investigación que responde a una tesis de maestría en construcción sobre los Procesos Educativos Mediados por Tecnologías. En esta oportunidad se identificarán las primeras dimensiones referidas al contexto sociocultural, organización institucional, régimen académico, estructuras curriculares y sus implicancias, disponibilidad de equipamiento informático y de conectividad, a fin de conocer el proceso de implementación de dicha orientación en Jujuy.

**Palabras claves:** Robótica, Espacio Curricular Articulado, Programación, Educación Obligatoria, Jujuy

### **Contexto socioeconómico y educativo de Jujuy**

Actualmente la sociedad de Jujuy se encuentra revolucionada, existen numerosas situaciones que alteraron las dinámicas de la normalidad, las cuales sacaron a la luz una serie de problemáticas que venían ocurriendo de forma silenciosa. Entre estas, y fundamentalmente la que ha sido el factor que atraviesa al sistema educativo de la provincia, se encuentran los salarios bajos en relación al nivel de la canasta familiar básica nacional. Situación que provoca desde el día 6 de Junio de 2023 manifestaciones masivas de docentes de los niveles obligatorios de la provincia. Haciendo fuerza al reclamo docente, se unieron las fuerzas de los pueblos originarios que en defensa del agua, el territorio y los recursos naturales, manifiestan junto a los docentes. Una situación que implica a todos los y las ciudadanas jujeñas, y constituye un punto de quiebre para la sociedad en su conjunto, es la aprobación de la reforma de la constitución provincial<sup>1</sup>, realizada de forma inconsulta a la sociedad en su conjunto, provocando serios episodios desde el día 20 de junio de 2023. A partir de esa fecha la educación viene atravesando por eventos sin antecedentes, como la unión de estudiantes para realizar la toma de colegios, de los docentes, estudiantes y la comunidad en general para hacer escuchar las protestas y las violentas represiones que se dan en los diferentes puntos de la provincia, incluso la fuerza de seguridad llegó a irrumpir en la Universidad Nacional de Jujuy (UNJu), violando su autonomía universitaria.

De esta manera, en los análisis realizados en las instituciones educativas, se refleja un clima enrarecido por los hechos acontecidos en los últimos tiempos. Estos se gestan, ya desde hace años y como se mencionaba anteriormente, por un lado por los bajos salarios, por el otro lado las condiciones de precarización en los cargos docentes, las malas condiciones de los edificios escolares, la falta de tecnología y conectividad en la mayoría de las instituciones educativas. Se destaca como punto de inflexión y de forma positiva la posibilidad de visibilización nacional apelando a la toma de conciencia de la sociedad en general.

### **Análisis Estructural: Tecnología, Conectividad y otras implicancias**

---

<sup>1</sup> Convención Constituyente. (20 de junio de 2023) Constitución de la Provincia de Jujuy. Boletín Oficial. Provincia de Jujuy. <https://boletinoficial.jujuy.gob.ar/wp-content/uploads/2016/Boletines/2023/68-2023.pdf>

El Gobierno de la Provincia de Jujuy, como parte de la estrategia general implementa a través del Préstamo del Banco de Desarrollo de América Latina (CAF) -según “Convenio CFA-11194” del 30 de marzo de 2020-, el “Programa de Mejora del Acceso y la Calidad Educativa” (PROMACE)-Decreto 853/2019. Este programa tiene como objetivo mejorar la equidad y el acceso a una educación en todos los niveles para contribuir a una educación de calidad con igualdad e inclusión plena y democrática, priorizando necesidades y atendiendo demandas de la sociedad como la formación continua y en servicio y fortalecer la pertinencia de las ofertas educativas que permitan contribuir a la inclusión de la población más vulnerable, a la transformación productiva de la provincia de Jujuy y a la formación continua de los docentes de los niveles obligatorios en desarrollo curricular y educación digital.

Entre las acciones que tiene a cargo el PROMACE se destacan la construcción de escuelas, la implementación de un dispositivo de capacitación a docentes y la distribución de ADM (Aulas Digitales Móviles) en las instituciones escolares de todos los niveles en Jujuy. En el Informe “Apoyo al Ministerio de Educación de la provincia de Jujuy a la adopción integral de tecnología en el sistema educativo a través de la implementación de Aulas Digitales Móviles” realizado en este marco por la empresa Telecom (2023), se advierte que: En relación con los desafíos tecnológicos, la Provincia cuenta con retos importantes en lo que refiere al acceso a infraestructura y conectividad. En el primer caso, se advierte un panorama desigual en cuanto a la disponibilidad de dispositivos en las instituciones educativas, como así también disparidades en cuanto al tipo (computadoras, netbooks, notebooks) y uso (administrativo, pedagógico o ambos; estudiantes /docentes). En materia de conectividad, actualmente Jujuy dispone de tres redes de telecomunicaciones que posibilitan la conectividad de 288 de un total de 641 edificios educativos de gestión estatal, en los que se albergan 780 instituciones educativas, según la siguiente distribución:

<b>Tipo</b>	<b>Red</b>	<b>Cantidad de edificios educativos conectados</b>
Redes inalámbricas	Red de Seguridad Provincial	89
	Red ENACOM	60
Red inalámbrica satelital	Red Satelital ARSAT	139

Tabla 1. Redes de telecomunicaciones existentes.  
Fuente: Diseño Global Conectividad-PROMACE (2023).

Los datos reportados también reflejan una estrecha relación entre la desigualdad, el uso de las tecnologías y su posibilidad de enriquecer los procesos de aprendizaje. En tal sentido, destacamos algunos datos relevantes identificados por la Provincia para el nivel secundario.

Aunque previo a la pandemia, el análisis de los usos de los dispositivos vinculados con actividades escolares por parte del estudiantado daba cuenta, por ejemplo, en el caso de la búsqueda de información en internet, de una posible relación con la desigualdad identificada entre regiones en cuanto a la conectividad (Gráfico 1).

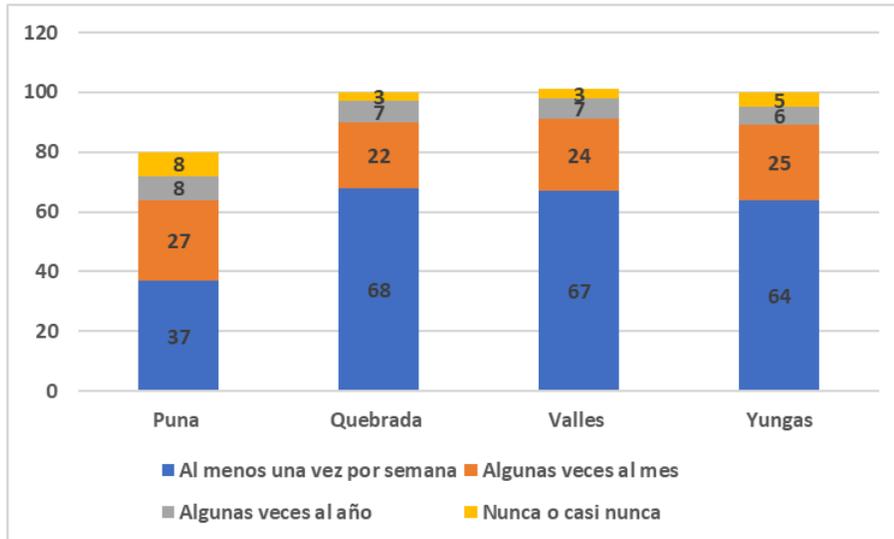


Gráfico 1. Estudiantes que buscan información en internet para realizar tareas escolares según frecuencia de uso. Según ecorregión. Puna – Quebrada – Valles – Yungas.  
Fuente: Ministerio de Educación de Jujuy (marzo, 2020).

Sin embargo, aún con esta desigualdad de tipo tecnológico, no se identifican brechas entre estudiantes de diferentes regiones en cuanto a la elaboración de videos o fotos para trabajos escolares (Gráfico 2).

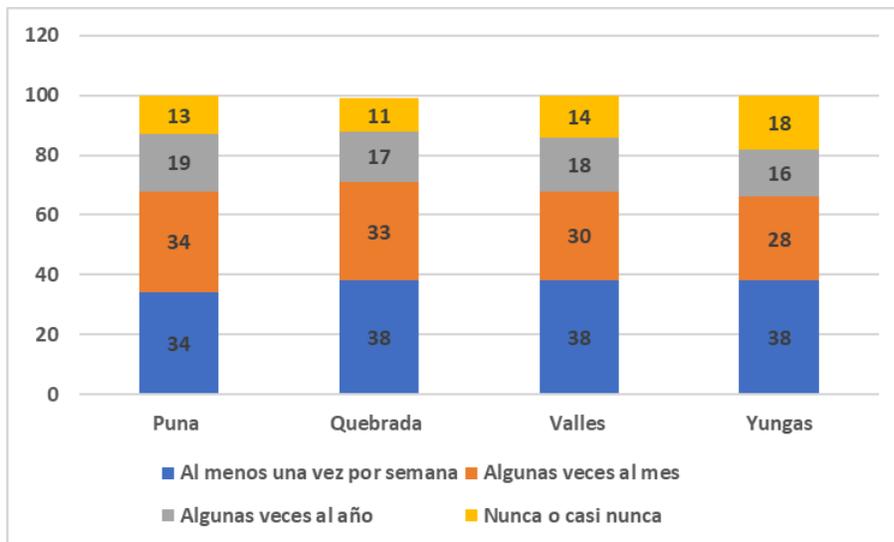


Gráfico 2. Porcentaje de estudiantes que realizan videos o fotos para trabajos escolares según frecuencia de uso. Según ecorregión. Puna – Quebrada – Valles – Yungas.  
Fuente: Ministerio de Educación de Jujuy (marzo, 2020).

Esta paridad de uso, entonces, no podría ser explicada desde la desigualdad en la conectividad, sino que podría decirse que allí intervienen la apropiación de las herramientas computacionales y las habilidades digitales disponibles, otorgándole relevancia a la acción de la escuela para movilizar el aprendizaje por parte de estudiantes. Se trata de una cuestión central en relación con los desafíos pedagógicos del sistema educativo, ya que refiere al valor y peso específico de la tarea de la escuela en relación con la distribución de conocimientos vinculados con la educación digital. Y, en dirección hacia ese objetivo, los desafíos parecen ser de envergadura, si se tiene en cuenta la percepción estudiantil sobre el papel de la escuela vinculado con estos aprendizajes. Tal como señalan los datos, los estudiantes no parecen reparar en la escuela como un lugar de usos y de aprendizajes significativos de las tecnologías (Gráfico 3).

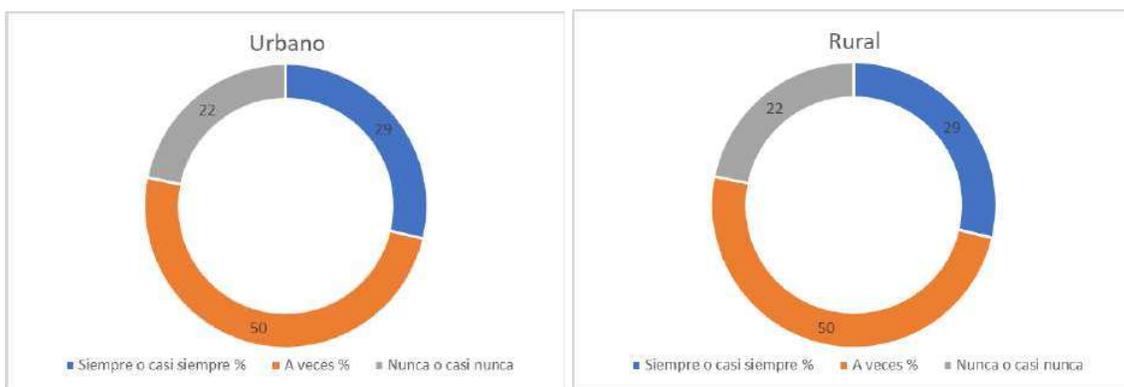


Gráfico 3. Porcentaje de estudiantes según percepción sobre el uso de las computadoras en la escuela “Aprendo más”. Según ámbito. Urbano – Rural.  
Fuente: Ministerio de Educación de Jujuy (marzo, 2020).

En este escenario, la apropiación con calidad y equidad de habilidades digitales – tanto en estudiantes como en docentes – constituye un reto que pone en cuestión la integración de las tecnologías digitales hacia la mejora de los aprendizajes de los y las estudiantes. Las intencionalidades pedagógicas orientadas en esa dirección se ven plasmadas en el diseño curricular de los niveles obligatorios del sistema educativo, donde se incluye la Educación Digital como parte del corpus de conocimientos y habilidades a ser incluidos en la enseñanza para el aprendizaje.

En cuanto a desarrollo profesional, la tarea docente ocupa un lugar central en las prioridades de la política educativa provincial en tanto los y las docentes resultan actores clave para la transformación digital del escenario educativo. Así, la formación docente continua constituye un punto de apoyo para el grupo de docentes de la Provincia que acompaña el cambio de las prácticas en el aula, al tiempo que fortalece las competencias profesionales de quienes están a su cargo.

### **El origen de la Orientación de Robótica y Programación. ¿Cómo se llega a la definición de las 16 orientaciones en Jujuy?**

Según lo establece la Res. 7621-E-2017 todas las escuelas secundarias se organizan en dos ciclos: Básico y Orientado. Esto se implementa progresivamente, a partir de 2018 y desde primer año del ciclo básico, en todas las instituciones con ofertas de Bachilleratos, Comerciales y Polimodales, además de renovar las estructuras curriculares con un criterio de formación general – para garantizar una base común de saberes a partir de la cual pensar la igualdad al acceso a la cultura – y específica, según la orientación y los perfiles esperados.

Este proceso convirtió una secundaria donde regían 122 Planes de Estudio, a una donde conviven 16 Estructuras Curriculares que corresponden a las 16 Orientaciones adoptadas por la provincia de Jujuy. Proceso que continúa con la readecuación en las modalidades artísticas y con ofertas para adultos.

El Departamento de Desarrollo Curricular dependiente de la Dirección de Planeamiento Educativo de la Secretaría de Innovación y Calidad Educativa del Ministerio de Educación de la Provincia de Jujuy, en conjunto con 270 Docentes y Supervisores de todas las regiones, realiza el trabajo permanente y articulado con las áreas de la Secretaría de Gestión Educativa, en 9 meses desde su creación (decreto 5999/18) hasta que se regularizan mediante resolución 11143-E-18 las primeras 13 orientaciones. Luego se aprueban tres orientaciones más donde nace la Orientación Robótica y Programación. (Resolución 182-E-19. Diseño Curricular. Ciclo Orientado. Tomo 16. Robótica y Programación, 2019).

### **Cambios institucionales**

El desafío se extiende a toda la comunidad educativa de la Educación Secundaria, ya que la implementación de las 16 orientaciones, consiste en reorganizar, revisar y promover nuevas formas de hacer para todos los actores institucionales. Se establece un documento organizador llamado Marco General, que brinda criterios generales, objetivos de la Educación Secundaria en Jujuy, Gestión Pedagógico Curricular, Fundamentos Pedagógicos, Orientaciones Didáctico-Metodológicas y Problemáticas Transversales.

En este dispositivo se habilita al equipo de gestión de nivel secundario, compuesto por supervisores -grupo de profesores que supervisa un grupo de escuelas secundarias- y directores de escuelas, a pensar conjuntamente diversas formas de adopción del Diseño Curricular en cada situación particular. De allí la importancia de interpretar la norma curricular en diálogo con lo que se presenta en la realidad educativa escolar, teniendo en cuenta dos aspectos fundamentales: uno relacionado al contexto sociocultural en que la escuela está inmersa, los actores institucionales que la habitan y el perfil de egresados que forma. El otro aspecto, teniendo al anterior como punto

de partida, pero con miras proyectivas de la escuela y con qué elementos dispone para efectivizar la organización curricular que se propone (Diseño curricular para la educación secundaria tomo 1: marco general y ciclo básico, 2018, p. 23).

En este sentido, las escuelas de la Orientación Robótica y Programación se encuentran en diversos contextos y tienen realidades diferentes. A continuación se mencionan las escuelas que tienen esta propuesta:

#	NOMBRE	REGIÓN	LOCALIDAD
1	Bachillerato Provincial N° 4 – Armada Argentina	IV	La Mendieta
2	Colegio Polimodal N° 2	I	Abra Pampa
3	Colegio Secundario N° 1 – Crucero Gral. Belgrano	III	S.S. de Jujuy
4	Colegio Secundario N° 5	VI	Palpalá
5	Colegio Secundario N° 60	IV	San Pedro
6	Colegio Secundario N° 62	III	S.S. de Jujuy
7	Colegio Secundario N° 64	VI	Perico
8	Colegio Secundario N° 65	V	LGSM
9	Escuela Normal República de Bolivia	II	Humahuaca
10	Escuela Normal Superior Juan Ignacio Gorriti	III	S.S. de Jujuy
11	Colegio Privado del Salvador	III	S.S. de Jujuy

Del total de 202 escuelas secundarias de la provincia, se presentan en la tabla 2, las 11 instituciones que eligen la orientación en “Robótica y Programación” (aproximadamente un 5% del total), donde 10 de ellas son de gestión pública y, la décima escuela, es de gestión privada. En el marco de este cambio, la Orientación se encuentra en un proceso gradual de implementación desde el año 2020, año en el cual se dio inicio al nuevo ciclo orientado, concretando la renovación curricular en las escuelas secundarias.

### **Estructuras Curriculares**

Las estructuras curriculares que aparecen en estos nuevos marcos generales, se encuentran organizadas de forma procesual, lo que implica una mirada introspectiva del hacer docente y suponen el replanteo de modelos pedagógicos hacia propuestas diversas siendo en éstas muy importante el trabajo la interdisciplinario, la creación, la innovación y la colaboración con otros, priorizando el contexto y la mirada social.

En esta nueva organización por cada espacio curricular se realiza la presentación tradicional en cuanto a las horas de dictado y se especifican los formatos según los sujetos pedagógicos, los objetivos que se espera alcanzar, la naturaleza de los contenidos a enseñar y aprender, el tipo de vínculo con el conocimiento que se pretende generar, las maneras de abordaje e indagación que se espera favorecer y las capacidades que se desea desarrollar (Diseño curricular para la educación secundaria tomo 1 : marco general y ciclo básico, 2018). Entre ellos se encuentran trabajar desde formatos como Talleres, Ateneos, Laboratorios, Proyectos, Seminarios, los cuales se pueden articular e integrar para poder planificar propuestas multidisciplinares o interdisciplinares centradas en los saberes prioritarios que contribuyan al desarrollo de capacidades.

Atendiendo al marco de referencia antes mencionado y a la resolución del Consejo Federal de Educación 84/09 las formas de aproximación y vinculación con los saberes deben plantear a los y las estudiantes un espacio en el que la confianza y la exigencia permitan asumir la responsabilidad y el esfuerzo que implica un aprendizaje. Siguiendo esta lógica, una vez explicitados los objetivos se presenta una caja de organización de saberes de forma gradual, desde una menor complejidad a una mayor complejidad, por saberes y por año de estudio, en el ciclo superior corresponde 3ro., 4to. y 5to. Teniendo en cuenta esta lógica se presenta la estructura correspondiente a Robótica y Programación, que presentamos más abajo en la Tabla 3.

En relación a la organización del trabajo docente, esta propuesta busca consolidar equipos docentes comprometidos con la vida institucional de la escuela y con oportunidades para el trabajo colectivo. Esto estaría favorecido con una concentración horaria y/o la conformación de cargos para conseguir una participación activa en el proyecto de la escuela, desarrollar un sentido de pertenencia institucional, contar con tiempos y espacios para el trabajo colaborativo y el acompañamiento personalizado a las trayectorias estudiantiles

Esta nueva organización da lugar a cambios en varios sentidos, uno más organizativo y administrativo, es el que da la posibilidad a los docentes de tener un paquete de horas que les permite conseguir mayor pertenencia a la institución, como así también, realizar planificaciones y dar horas frente a los estudiantes en conjunto con algún colega con el que se complementan de acuerdo a los saberes disciplinares. Mientras que otro cambio que aparece está vinculado a la incorporación de las Ciencias de la Computación como parte de la obligatoriedad de la escuela secundaria, que permite llegar a la inclusión de saberes en los diferentes espacios curriculares, brindando a los y las estudiantes habilidades de una nueva alfabetización y un acercamiento al Pensamiento Computacional.

Nº Orden	ESPACIO CURRICULAR		3º	4º	5º
1	Lengua y Literatura		4 hs.	4 hs.	3 hs.
2	Matemática		4 hs.	4 hs.	4 hs.
3	Educación Física		3 hs.	3 hs.	3 hs.
4	Lengua Extranjera*		3 hs.	3 hs.	3 hs.
5	Ciencias Biológicas		3 hs.		
6	Físico-Química		2 hs.		
7	Historia		4 hs.		
8	Geografía		3 hs.		
9	Construcción para la Ciudadanía		3 hs.		
10	Arte ***		3 hs.		
11	Introducción a la Programación		3 hs.		
12	Introducción a la Robótica y Electromecánica		4 hs.		
13	Física			4 hs.	3 hs.
14	Taller Integrador de Ciencias Sociales (1)	Historia (4 hs.)		6 hs.	
		Geografía (4 hs.)			
15	Arte ****			3 hs.	
16	Sistemas de Comunicación			3 hs.	
17	Taller de Programación y Robótica (2)	Programación I (8 hs.) Robótica I (8 hs.)		8 hs.	
18	Química				3 hs.
19	Taller Integrador de Cs. Sociales (3)	Historia (4 hs.) Filosofía (4Hs.)			6 hs.
20	Programación Aplicada a la Robótica				6 hs.
21	Laboratorio de Robótica Aplicada (2)	Programación II (8 hs.) Robótica II (8 hs.)			8 hs.

\* **Lengua Extranjera:** La Institución ofrecerá en el espacio **Lengua Extranjera: Inglés**, por la especificidad de la Orientación, es decir que permita comprender y utilizar textos, con contenidos técnicos, tutoriales, programa, etc.

\*\*\* **Arte.** En este espacio la Institución, por la especificidad de la orientación, deberá ofrecer **Arte Multimedial**. Es decir, el espacio deberá estar a cargo de un Docente de Arte con la formación específica para dicho espacio

\*\*\*\***Arte:** Por la orientación de estas Instituciones, en este espacio curricular se deberá impartir **Arte y diseño asistido por computadora** y deberá estar a cargo de un Docente de Arte con la formación específica para dicho espacio.

**(1) Taller de Cs. Sociales:** El taller de cuarto año consta de 6 hs. cátedras y estará a cargo de: un profesor de historia con 4 hs. cátedras y un profesor de geografía con 4 hs. cátedras. Los docentes impartirán, a los estudiantes, cada uno un módulo en forma individual (2 hs. cat.); y ambos como equipo de enseñanza, impartirán simultáneamente un módulo (2 hs. cat.) frente a los alumnos.

**(2) El Taller de Programación y Robótica y el Laboratorio de Robótica Aplicada** estarán a cargo de dos docentes, uno de Robótica y uno de Programación, con 8 hs. cat. cada uno, que trabajarán en equipo de enseñanza simultáneamente en las 8 hs. cátedras que posee el espacio curricular.

**(3) Taller de Cs. Sociales:** El taller de quinto año posee 6 horas cátedras y estará a cargo de un profesor de historia (4 hs. cat.) y un profesor de filosofía (4 hs. cat.). Los docentes estarán: un módulo en forma individual (2 hs. cat.) con los estudiantes; y ambos como equipo de enseñanza, impartirán simultáneamente un módulo (2 hs. cat.) frente a los alumnos.

Tabla 3: Caja de organización de saberes de Robótica y Programación

Como se observa en la tabla 3, entre los espacios curriculares que tienen esta característica innovadora se encuentran los espacios curriculares n° 14 y 19. Taller Integrador de Ciencias Sociales (1 y 2) y los espacios vinculados a las CC: n° 17. Taller de Programación y Robótica y n° 21 Laboratorio de Robótica Aplicada. Los espacios curriculares interdisciplinarios, articulados o integrados. Esta innovación permite, en relación a la formación específica, la creación de los Espacios Curriculares Articulados (ECA), en formato taller y laboratorio, conformados por dos personas especialistas: una en programación y otra en robótica.

### **Acompañamiento Docente**

Como seguimiento de la implementación, junto a los responsables de las propuestas, el departamento de Desarrollo Curricular organiza regularmente encuentros con docentes y directivos, donde se comparten inquietudes acerca del trabajo que realizan, la forma en que se está llevando a cabo, dificultades, propuestas de mejora y todo lo que pueda contribuir a seguir con la ejecución de las orientaciones en el territorio.

A partir de estas reuniones mantenidas por el departamento de Desarrollo Curricular y los referentes de las escuelas con orientación en Robótica y Programación se registran situaciones problemáticas relatadas por docentes de la orientación, directores y supervisores, que son de gran interés para la investigación. Dentro de los problemas registrados se observan:

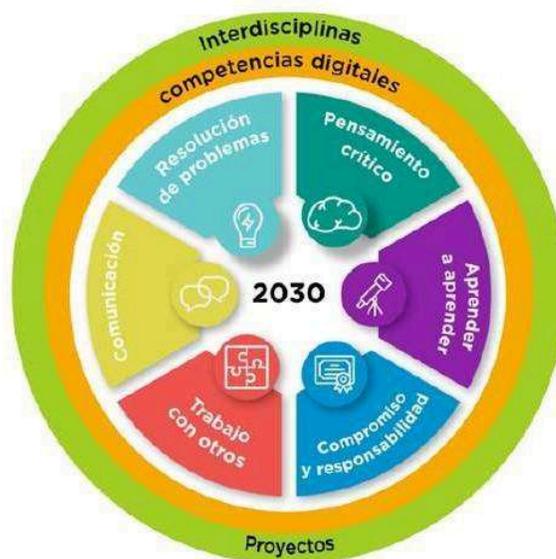
- Una confusión entre tecnología y tecnología informática.
- Estiman que existen graves problemas con los perfiles docentes.
- Consideran que es necesario el equipamiento en las escuelas para la implementación de la orientación en robótica y solicitan espacios de formación docente al registrar no tener suficientes prácticas.
- Expresan lo importante que ven a la orientación para el contexto barrial de alta vulnerabilidad, ya que Robótica y Programación son espacios de construcción colectiva.
- Afirman que cuentan con un piso tecnológico, pero no tienen buena conectividad y estiman necesitar al menos 3 kits de robótica.
- Por último, hablan de la robótica y sus contradicciones, preguntando si es posible la capacitación interdisciplinaria. Pese a que la interdisciplinariedad, la enseñanza mediante saberes prioritarios y el abordaje de temáticas transversales en la enseñanza son centrales en los diseños aprobados, los actores institucionales ponen en duda su posible puesta en práctica.

En este marco y siendo una de las problemáticas expuestas por los actores involucrados en su incorporación a la enseñanza media, nos detendremos en la implementación de los ECA ya que constituye la unidad de análisis de la investigación de maestría mencionada al inicio.

### **Espacios Curriculares Articulados (ECA)**

Los ECA surgen de la innovación propuesta, en la enseñanza partiendo de la re-conceptualización de la idea de aprendizaje integrado. No sólo como la articulación de saberes, sin excluir la enseñanza específica disciplinaria, sino como la creación de estructuras curriculares que propician más concretamente la articulación de saberes, desde la fusión de diferentes espacios disciplinarios. Además, según el “Marco de Organización de los Aprendizajes para la educación Obligatoria Argentina” (MOA) aprobado mediante Resolución CFE N° 330/17, se debe:

“avanzar hacia una organización institucional y pedagógica que incorpore instancias de aprendizaje interdisciplinario que integren dos o tres disciplinas en cada año escolar de la educación obligatoria, desarrollado de acuerdo con los intereses y necesidades de cada contexto (...)



Estas instancias pueden organizarse como módulos o proyectos de aprendizaje interdisciplinario que (...) promueven el diálogo entre las diferentes disciplinas o áreas del conocimiento y apuntan a la resolución por parte de los/las estudiantes –de manera colectiva y colaborativa– de problemas que suponen alguna producción. (...)” (MOA, 2017 pp. 11 – 12)

Este modo de enseñar busca tensionar a la enseñanza tradicional estableciendo conexiones entre los contenidos que sean significativos para la vida de los y las estudiantes, situación que puede

ser más dificultosa si se segregan los conocimientos. Los ECA tratan entonces de fortalecer la triada docentes, alumnos y conocimiento, destacando capacidades fundamentales que plantea el MOA relacionadas al Pensamiento Computacional: resolución de problemas, pensamiento lógico, creatividad, leer y escribir lenguaje de programación, aprender a aprender, trabajar en equipo, capacidad de comunicación, reconocimiento de patrones, abstracción, análisis descendente, construcción de modelos, tolerancia a las frustraciones ante fallos de sistemas de procesamiento de información, construcción y programación de soluciones automatizadas, competentes para la vida en la sociedad del conocimiento.

Estos saberes emergen de la propuesta de integración de la robótica y la programación a las escuelas. En estos confluyen tanto la lógica y la abstracción como la imaginación, la expresión y la capacidad de idear y construir, en forma individual o con otros. Estos campos de conocimiento también propician el trabajo en equipo, la colaboración y el aprendizaje entre pares, dimensiones que deben promoverse en las propuestas de enseñanza y aprendizaje, ya que forman parte de los modos de construcción de conocimiento y de las culturas del mundo del trabajo de la sociedad digital.

### **Algunas consideraciones finales**

Martínez y Echeveste (2020) afirman que un proyecto interdisciplinario incluye el Pensamiento Computacional (PC) cuando se piensa en automatizar una tarea, o procesar y transformar datos. Aborda la participación computacional cuando se torna relevante, se trabaja colaborativamente, y soluciona un problema del entorno. Según esta afirmación y analizando las propuestas de los espacios articulados, se puede evidenciar la intención de ir afianzando y potenciando las competencias de los estudiantes en base a los proyectos interdisciplinarios a favor de una formación integral donde el PC es transversal.

Al respecto Brennan y Resnick (2012) citan a Cuny, Snyder y Wing (2010) y definen al Pensamiento Computacional “como procesos de pensamiento involucrados en formular problemas y encontrar sus soluciones de manera que las soluciones estén representadas de forma tal que puedan llevarse a cabo efectivamente por un agente que procesa información” (p. 3). Agregan a esta definición las tres dimensiones que lo componen: conceptos computacionales, prácticas computacionales y perspectivas computacionales.

Trabajar con habilidades cognitivas para analizar, realizar procesos de abstracción y solucionar un problema por medio del PC, podría forjar en los estudiantes mayores posibilidades de entender, crear y consumir con sentido crítico la tecnología. En relación a lo expuesto Bonello y Schapachnik (2020) en el texto “Diez preguntas frecuentes (y urgentes) sobre Pensamiento Computacional enfatizan “la importancia y urgencia que reviste la incorporación de estos

contenidos a las currículas escolares fundamentado en la relevancia social adquirida por el dominio fluido de estos saberes. Este conocimiento contribuye a garantizar el pleno ejercicio de los derechos ciudadanos en el siglo XXI” (pp. 165 - 166).

La incorporación a la enseñanza obligatoria, pública y gratuita de los ECA de Robótica y Programación es una realidad en el sistema educativo provincial de Jujuy. Hasta el momento hemos realizado una contextualización que enmarca el espacio que actualmente tienen las CC en la provincia jujeña. El sentido del trabajo de la investigación se cumpliría cuando se avance al lograr el análisis de la implementación de estos espacios, lo que representaría a futuro, una contribución para mejorar experiencias similares brindando información significativa.

### **Bibliografía**

- Bonello, M. B. & Schapachnik, F. (2020). Diez preguntas frecuentes (y urgentes) sobre pensamiento computacional. En *Virtualidad, Educación y Ciencia*, 11, 156-167
- Brennan, K. & Resnick, M. (2012). Nuevos marcos de referencia para estudiar y evaluar el desarrollo del pensamiento computacional. American Educational Research Association (AERA) Vancouver, BC, Canada.
- Martínez, C. y Echeveste, E. (2020). “Propuestas de enseñanzas de las Ciencias de la Computación.” Clase 4. Seminario de Pensamiento Computacional. Maestría en Procesos Educativos Mediados por Tecnologías. CEA, Universidad Nacional de Córdoba.
- Ministerio de Educación. Secretaría de Innovación y Calidad Educativa (2017) MOA. Marco de Organización de los Aprendizajes para la Educación Obligatoria Argentina. Secretaría de Innovación y Calidad Educativa. Resolución CFE N° 330/17.
- Ministerio de Educación. Secretaría de Innovación y Calidad Educativa. Dirección de Planeamiento Educativo (2019). “Diseño curricular para la educación secundaria” Diseño Curricular. Ciclo Orientado. Robótica y Programación. Tomo 16. Resolución 182-E-19. Jujuy.
- Ministerio de Educación. Secretaría de Innovación y Calidad Educativa. Dirección de Planeamiento Educativo (2018) “Diseño curricular para la educación secundaria” Tomo 1 : marco general y ciclo básico.
- Telecom Advisory Services (2023) “Apoyo al Ministerio de Educación de la Provincia de Jujuy a la adopción integral de Tecnología en el Sistema Educativo a través de la implementación de Aulas Digitales Móviles (ADM)”. Informe final.

**Festivales de robótica del Gran Río Cuarto:  
alternativa para el aprendizaje en comunidad en contextos no formales**

Ariel Ferreira Szpiniak <sup>1</sup>, Carlos Federico Etcheverry <sup>2</sup>, Carlos Maximiliano Correa <sup>2</sup>  
aferreira@exa.unrc.edu.ar, {maximiliano.c.correa, etcheverry95}@gmail.com

<sup>1</sup> Universidad Nacional de Río Cuarto

<sup>2</sup> Amulen Project

## **Resumen**

Los Festivales de robótica del Gran Río Cuarto (Provincia de Córdoba) son una oportunidad para que las/os jóvenes escolarizadas/os se acerquen a la tecnología desde otra mirada, aprender de una forma lúdica como es la lógica de funcionamiento de los dispositivos tecnológicos, cómo se pueden programar robots y tomar contacto con ellos, experimentando y comprobando su funcionamiento.

Los objetivos propuestos están relacionados con la vinculación entre la universidad y la escuela secundaria; el intercambio de conocimientos y experiencias entre estudiantes universitarios y secundarios; descubrir vocaciones científicas en las/os jóvenes; entender cómo funciona la tecnología que rige al mundo actual; aprender conceptos básicos de las ciencias de la computación; aprender a programar robots, manipularlos, desmitificarlos y compartir saberes.

Los Festivales se realizan de forma articulada con talleres previos, desarrollados a lo largo del año, en los cuales participan estudiantes de 1ro a 7mo año de educación secundaria. El primer Festival se realizó en el año 2016 y en el año 2023 se realizará la sexta edición. En las primeras cinco ediciones participaron más de 600 estudiantes de 25 escuelas de Río Cuarto y la región.

La organización de cada Festival estuvo a cargo de la Secretaría de Extensión y Desarrollo de la Universidad Nacional de Río Cuarto (UNRC), en conjunto con diversas entidades educativas, gubernamentales, profesionales y culturales.

A lo largo de las cinco ediciones anteriores se desarrollaron diferentes actividades, tales como: talleres de robótica (robots bípedos, autos, seguidores de líneas y catapulta); muestra de desarrollos y actividades relacionadas con la informática, la robótica y la programación; visita al Museo Tecnológico Aeroespacial; Copa Turing; talleres de programación, olimpiada de Robótica y Programación del Gran Río Cuarto; stands tecnológicos, informativos y educativos; escenarios de trabajo para estudiantes (N6, Otto, EduKit10, Kits Arduino) y escenarios Demo (drones y autobots).

Para la sexta edición se está articulando con el Cluster Tecnológico Río Cuarto, el Punto Digital Río Cuarto, la Municipalidad de la Ciudad de Río Cuarto, la Fundación por la Cultura y la Asociación Civil La Cultural.

**Palabras clave:** Festival, Programación, Robótica, Educación, Aprendizaje, Comunidad, Jóvenes

## Fundamentación

La utilización de Tecnologías de la Información y Comunicación (TIC) en el ámbito educativo no es algo novedoso, aunque cada vez más importante. Estos recursos buscan generar otras formas de educar, resignificando las configuraciones clásicas del aula. Utilizar TIC para enseñar y aprender sobre ellas y otras disciplinas es un desafío.

La enseñanza de la programación y la robótica mediante el planteo de una situación a resolver, proporciona un papel activo de las/os estudiantes en la búsqueda de soluciones a los problemas planteados, mediante diseño y experimentación, estimulando un aprendizaje significativo. Además, los resultados visibles de la solución planteada en un robot, proveen una retroalimentación positiva que genera emociones y estimula la curiosidad en las/os estudiantes, que pueden reconocer la utilidad de los contenidos de una manera casi instantánea.

La aplicación de la robótica en educación, robótica educativa, puede enmarcarse en varias de las corrientes pedagógicas modernas. Incluso desde el punto de vista de la teoría de las inteligencias múltiples, es decir, apuntar al aprendizaje de un tema desde distintas inteligencias (lógico-matemática, lingüística, espacial, cinético-corporal, musical, etc). La robótica es entendida como un vehículo para mejorar las prácticas de enseñanza y los resultados de las/os estudiantes en otras disciplinas, sumado a comunicar la importancia o utilidad de las TIC para lograr una mejor predisposición del estudiantado hacia el aprendizaje, relacionando las metas académicas con sus proyectos de vida.

Dado su carácter polivalente y multidisciplinario, la construcción y programación de un juguete/objeto como puede ser un robot, permite el abordaje de conocimientos variados de las disciplinas STREAM (ciencia, tecnología, robótica, ingeniería, arte y matemática). En particular, dentro del amplitud del término "ciencia", las ciencias sociales y naturales están presentes de forma muy activa, y las ciencias de la computación juegan un rol fundamental. Los robots estimulan el pensamiento computacional, la imaginación, la innovación, la creatividad, el pensamiento reflexivo y científico de las/os estudiantes (en relación a la formulación de hipótesis, la experimentación y la elaboración de conclusiones), quienes al enfrentarse a un "problema" dado aprenden a experimentar, diseñar y resolver situaciones siendo verdaderos protagonistas de su aprendizaje.

En el proceso de "pensar el robot", tanto para programarlo como para construirlo, se generan las condiciones de apropiación del conocimiento. Se trata de otorgar a las/os estudiantes un rol activo en sus aprendizajes, colocándolos como diseñadores de sus propios proyectos y constructores de conocimientos.

El uso de hardware en el ámbito escolar permite tener control sobre las características del mismo, adaptarlo a las necesidades concretas del ámbito y las realidades sociales, culturales y económicas de la institución.

La programación es una de las áreas más importantes en ciencias de la computación. Éstas

abordan el estudio de las computadoras desde los principios lógicos y formales que hacen posible su funcionamiento, hasta la solución problemas. Es decir, más allá de los detalles que presentan las tecnologías concretas que utilizamos a diario.

Como disciplina, la programación está orientada a la resolución de problemas. Ello requiere del desarrollo de una serie de habilidades de abstracción y de operacionalidad. La abstracción abarca técnicas como la simplificación de problemas, la definición de soluciones generales aplicables a problemas similares y la asignación de nombres significativos a las distintas partes de una solución. La operacionalidad supone la definición de soluciones en términos de un conjunto de pasos que deben ejecutarse en un orden determinado para alcanzar un objetivo.

Las habilidades mencionadas no son propias de la programación, pero condición necesaria para abordar la solución de problemas planteados. Es decir, la programación estimula el desarrollo de la abstracción y la operacionalidad, con lo cual también ayuda a mejorar los aprendizajes.

Desde el punto de vista de las políticas públicas orientadas a la aplicación de TIC en educación, el Programa Conectar Igualdad es el más importante. Fue creado en el año 2010 durante el gobierno de Cristina Fernández de Kirchner y apunta a reducir las brechas digitales de primer, segundo y tercer orden: acceso a los dispositivos tecnológicos (1), formación docente para la incorporación de tecnologías educativas en el aula (2) y comprender cómo funcionan las computadoras y el mundo digital (3). Conectar Igualdad se propuso el objetivo de entregar una notebook a todas/os las/os estudiantes y docentes de educación secundaria y especial de gestión pública, capacitar a las/os docentes e incorporar infraestructura escolar específica (servidor y red interna) para la efectiva utilización dentro de la escuela (piso tecnológico). Hasta fines de 2015, Conectar Igualdad había capacitado más del seiscientos mil docentes de todo el país y entregado más de cinco millones y medio de netbooks. También se puso en marcha el programa para equipar a las escuelas primarias con un laboratorio móvil de netbooks (carro tecnológico). Más allá del equipamiento, uno de sus mayores aportes ha sido su clara mirada soberana, tanto tecnológica como pedagógica, a través del desarrollo del Sistema Operativo Huayra GNU/Linux, todas las aplicaciones educativas integradas al mismo, la capacitación para su aplicación educativa, su inclusión en las netbooks y la posibilidad de utilizarlo en cualquier otro ámbito, gracias a la puesta en valor del software libre como herramienta para que las/os estudiantes aprendan sobre programas, los usen, estudien, modifiquen y mejoren. A la par de los nuevos modelos de netbooks que se fueron desarrollando, Huayra GNU/Linux tuvo una mejora continua, llegando hasta la versión 3.

Entre 2016 y 2019, durante el gobierno de Mauricio Macri, se discontinuó el programa y fue reemplazado por otros de menor envergadura (Escuelas del Futuro, Aprender Conectados) y concepciones políticas muy diferentes, que no lograron superar la barrera del tecnicismo, adquisición y uso de equipamiento importado que tuvo en cuenta el contexto de uso, sin política de mantenimiento y actualización, poca o nula bajada pedagógica. Incluso algunos con software y hardware propietario que no podían utilizarse con el equipamiento disponible en las escuelas.

Luego del parate que se produjo entre 2016 y 2019, el gobierno de Alberto Fernández recuperó el Programa Conectar Igualdad y retomó algunas de sus principales líneas de acción, dentro de las cuales se destaca la entrega de alrededor de un millón de netbooks, nuevas versiones de Huayra GNU/Linux (actualmente está en su versión 6), producción de contenidos y capacitaciones.

Desde 2014, también el Ministerio de Educación de la Provincia de Córdoba avanzó con diferentes programas como Escuelas ProA, Unidad de Educación Digital, Especialización en Enseñanza de la Programación, Cursos y Talleres. Estas acciones que han permitido generar cierto conocimiento respecto de los lineamientos pedagógicos favorables para el aprendizaje de la programación y la robótica. En el año 2022 se presentó el documento sobre "Aportes de Cultura Digital en la Educación Tecnológica", actualización curricular, con la finalidad de fortalecer las prácticas en educación digital. donde se incorporan contenidos de Cultura Digital, Programación y Robótica a fin de promover su alfabetización (Aportes de Cultura Digital en la Educación Tecnológica, 2022).

Para acompañar las diferentes acciones se lograron avances significativos en el marco del Consejo Federal de Educación. En el año 2015 se aprobó la Resolución N° 263/15 donde se establece que "la enseñanza y el aprendizaje de la "Programación" es de importancia estratégica en el Sistema Educativo Nacional durante la escolaridad obligatoria, para fortalecer el desarrollo económico-social de la Nación, conforme lo establecido por el artículo 3° de la Ley de Educación Nacional" y en el año 2018 la Resolución N° 343/18 sobre los Núcleos de Aprendizajes Prioritarios de Educación Digital, Programación y Robótica que plantean la obligatoriedad de la educación digital, la programación y la robótica en todos los establecimientos del país, dando paso a su integración curricular en la educación obligatoria, en consonancia con la Ley de Educación Nacional N° 26.206.

## **Origen**

Los Festivales que describiremos a continuación se enmarcan en las políticas públicas citadas anteriormente, desde el año 2010 hasta la fecha, y la legislación nacional y provincial. Están en correspondencia con "... el desarrollo del conjunto de competencias y capacidades necesarias para que los y las estudiantes puedan integrarse plenamente en la cultura digital" (NAP Educación Digital, Programación y Robótica, 2019).

Los Festivales de robótica del Gran Río Cuarto se plantearon como una alternativa para el aprendizaje en comunidad en contextos no formales, como una oportunidad para que las/os jóvenes escolarizadas/os se acerquen a la tecnología desde otra mirada, aprender de una forma lúdica como es la lógica de funcionamiento de los dispositivos tecnológicos, cómo se pueden programar robots y tomar contacto con ellos, experimentando y comprobando su funcionamiento. La idea surge a partir de los "Festivales Conectar", realizados en la ciudad de Villa María en los años 2012 y 2013, que fueron organizados por el Ministerio de Educación de la Nación y el portal Educ.ar, junto al Ministerio de Educación de Córdoba, el Programa Conectar Igualdad y el Municipio de Villa María. Durante éstos Festivales se realizaron talleres para la utilización creativa de las netbooks entregadas por el Programa Conectar Igualdad. En e año 2012 se llevaron a cabo talleres

de Animación Digital, Tv Digital, Radio Digital, Robótica, Video juegos, Relatos Sonoros y Música Digital. En 2013 los talleres fueron: Periodismo Digital, Radio Digital y TV Digital, Arte y Física, Animación y Ciencias Sociales, Videojuegos y Robótica.

### Objetivos y Articulación

Los objetivos propuestos están relacionados con la vinculación entre la universidad y la escuela secundaria; el intercambio de conocimientos y experiencias entre estudiantes de ambos niveles; incentivar carreras terciarias y universitarias del sector TIC, difundir entre las y los adolescentes la importancia de aprender a programar, acceder y comprender el mundo de la robótica, descubrir vocaciones científicas, entender cómo funciona la tecnología que rige al mundo actual, y poder dominar esa tecnología.

Los Festivales se realizan de forma articulada con talleres previos, desarrollados a lo largo del año, en los cuales participan estudiantes de 1ro a 7mo año de educación secundaria. El primer Festival se concretó en el año 2016 y en el año 2023 se realizará la sexta edición. En las primeras cinco ediciones participaron más de 600 estudiantes de 25 escuelas de Río Cuarto y la región.

### Organización

La organización de cada Festival estuvo a cargo de la Secretaría de Extensión y Desarrollo de la UNRC, como principal responsable, pero se trabajó en conjunto con diversos actores a lo largo de las cinco ediciones. Entre los principales podemos nombrar a la Mesa TIC del Gran Río Cuarto (Municipalidad de la Ciudad de Río Cuarto, Santa Catalina y Las Higueras, el Cluster Tecnológico Río Cuarto), al Ministerio de Educación y al Ministerio de Ciencia y Tecnología de la Provincia de Córdoba, el IPET N° 26 "Juan Filloy", la Asociación Civil La Cultural, al Punto Digital y la Fundación por la Cultura de Río Cuarto.

### Sobre los Festivales

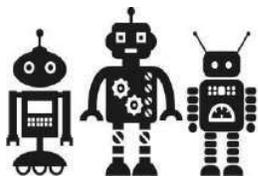


Figura 1: logo del primer Festival

FESTIVAL DE  
ROBÓTICA



FESTIVAL  
ROBÓTICA  
educativa



Figura 2: logo del segundo al quinto Festival

A lo largo de las cinco ediciones se desarrollaron diferentes actividades, tales como: talleres de robótica (robots bípedos, autos, seguidores de líneas y catapulta); muestra de desarrollos y actividades relacionadas con la informática, la robótica y la programación; visita al Museo Tecnológico Aeroespacial; Copa Turing; talleres de programación, olimpiada de robótica y programación del gran Río Cuarto; stands tecnológicos, informativos y educativos; escenarios de trabajo para estudiantes (N6, Otto, EduKit10, Kits Arduino) y escenarios Demo (drones y autobots).

A continuación sintetizamos lo acontecido en cada uno de los cinco festivales realizados.

## I Festival de Robótica del Gran Río Cuarto

El I Festival de Robótica del Gran Río Cuarto formó parte de las actividades llevadas a cabo por la Mesa TIC del Gran Río Cuarto en el marco de Semana Innovadora 2016. Se realizó el día viernes 30 de Septiembre de 2016, por la mañana y la tarde, en el predio del Área de Material Río Cuarto, perteneciente a la Fuerza Área Argentina, más precisamente en el Museo Tecnológico Aeroespacial, Hangar de Exposición e instalaciones del IPET N° 258 "Mayor Ingeniero Francisco de Arteaga". También contó con el apoyo del Ministerio de Educación de la Provincia en el marco del trabajo conjunto con la UNRC para llevar adelante los talleres de robótica educativa en escuelas secundarias de Río Cuarto y la Región.

Participaron más de 150 estudiantes de diferentes centros educativos: IPET 26 "Juan Filloy" de Río Cuarto, IPEM 27 "René Favalaro" de Río Cuarto, IPEAyT N° 186 "Capitán Luis Castagnari" de Las Higueras, IPET N° 258 "Mayor Ingeniero Francisco de Arteaga" de Las Higueras, IPEA N° 215 "Raúl Scalabrini Ortiz" de Santa Catina (Estación Holmberg), Colegio Dante Alighieri de Río Cuarto, Instituto Secundario Charras de Charras e Instituto Tecnológico Leonardo Da Vinci de Río Cuarto.

En dicha oportunidad se realizaron 3 tipos de actividades:

- Talleres de robótica montados en cuatro escenarios denominados: Pehuensat, Capitán Beto, Ar-sat y Tronador.

Para los cuatro escenarios se utilizaron kits del Programa Conectar Igualdad aportados por escuelas, software libre para programar (Arduino + ArduBlock) y hardware libre basado en Arduino. En Pehuensat se utilizaron Robots Bipedos "Otto" y la actividad consistió en enseñarle a caminar y evitar obstáculos. En Capitán Beto se utilizaron Autobots (autitos programables) y la actividad fue resolver un circuito/laberinto. En Ar-sat se utilizaron robots N6 y la actividad fue seguir líneas dentro de un circuito. Finalmente, en el escenario Tronador se utilizó la placa educativa EduKit10, un set de piezas Armamec y jengas. La actividad consistió en construir una catapulta, utilizando la EduKit10 y el Armamec y derribar un castillo de madera. Los Otto fueron elaborados por el equipo del Centro IRC de la Secretaría de Extensión y Desarrollo, la placa EduKit10 y los Autobots por IT10, empresa de base tecnológica de la Ciudad de Río Cuarto conformada por estudiantes y egresados de las carreras de Computación, Telecomunicaciones y Comunicación de la UNRC. El robot N6 fue adquirido a la empresa Robogroup.

Las/os estudiantes participantes rotaron por los 4 escenarios, aprendiendo diferentes contenidos de programación y robótica, sumado a matemática, física y tecnología.



Figura 3: escenario Capitán Beto y escenario Tronador



Figura 4: escenario Pehuensat y Ar-sat

- Muestra de desarrollos tecnológicos realizados en la UNRC, empresas y escuelas. Para ello se armaron 9 stands dentro del Hangar destinado a Exposiciones.

Animatronics y Kamishibai Robotizado (Empresa F8), Arduino para experimentar (IPET N° 26 "Juan Filloy"), Drones y Simulador de Vuelo (Facultad de Ingeniería de la UNRC), Robótica Educativa (Empresa IT10), Impresoras 3D (Empresa Diseño), Drones aéreos (Facultad de Ingeniería de la UNRC), Proyecto "EVA" (Programa Educativo de Adultos Mayores de la UNRC), Expo de Carreras Tecnológicas (Secretaría Académica de la UNRC) y Grooming (Cluster Tecnológico Río Cuarto).



Figura 5: Stands

- Visita al Museo Tecnológico Aeroespacial.

En el museo se brindó una charla y se compartió el almuerzo. Se pudieron apreciar algunas aeronaves que estuvieron en el Área Material, una gran maqueta del IA-63 Pampa y otras de aviones que integraron la Fuerza Aérea Argentina, fotos, el comedor de la tropa que cumplía con el Servicio Militar Obligatorio, y, en el exterior del Museo, los aviones IA-58B o IA-66.





Figura 6: Visita al Museo

Para la realización de las tres actividades se organizaron dos grupos, el primero participó de los talleres (subdivididos en cuatro) y el segundo visitó los Stands de la Muestra Tecnológica y el Museo. Luego del almuerzo se invirtieron los roles.

Para finalizar la jornada las/os participantes se reunieron en la Plaza de Armas donde la Banda Militar de Música Área Material de interpretó varias canciones y diversas autoridades de las organizaciones promotoras del festival pronunciaron palabras sobre el evento, agradecieron a las/os estudiantes participantes y destacaron la importancia de la soberanía tecnológica y la educación pública y gratuita. Por último, el Presidente del Concejo Deliberante de Las Higueras, en representación del Intendente, destacó la organización de esta convocatoria como así también la necesidad de promover en los jóvenes el estudio de carreras vinculadas a la innovación tecnológica.

## II Festival de Robótica del Gran Río Cuarto

El II Festival de Robótica del Gran Río Cuarto se realizó el día 15 de Septiembre del 2017 en el Centro Cultural Viejo Mercado de la Ciudad de Río Cuarto. Se realizó como parte de las actividades de la Semana Innovadora 2017, organizada por la Mesa TIC del Gran Río Cuarto, y contó con el apoyo del Ministerio de Educación de la Provincia.

Participaron de la segunda edición más de 150 estudiantes del IPET 26 "Juan Filloy", IPEM 27 "René Favalaro", IPEAyT 186 "Capitán Luis Castagnari", Colegio Bilingüe Bicultural "Italo Argentino Dante Alighieri", Instituto Tecnológico Leonardo Da Vinci, ATICA, Galileo Galilei, IPET 203 "Dr. Juan Bautista Dichiaro", IPET 389 de la localidad de Sampacho, Colegio La Merced, Colegio Santa Eufrasia y Escuela Experimental ProA (Programa Avanzado de Educación Secundaria), orientación en Desarrollo de Software - sede Río Cuarto.

En dicha oportunidad se realizaron 3 tipos de actividades:

- Talleres de robótica montados en cuatro escenarios denominados: Pehuensat, Capitan Beto,

Ar-sat y Tronador, similares al primer Festival.

- Hubo 10 stands con muestras de trabajos tecnológicos-educativos desarrollados por escuelas secundarias de Río Cuarto y la región, empresas, ONGs y la UNRC.
- Copa Turing (sede Río Cuarto), organizada por la Cámara de la Industria Argentina del Software de Argentina (CESSI), dirigida a jóvenes de escuelas secundarias y primarias de todo el país. Se trata de una competencia para niños y adolescentes



Figura 7: zona de pruebas, stands y escenarios

Luego de sus primeras ediciones (2016 y 2017), el equipo organizador de los Festivales fue invitado a participar del Festival Liber.ar, realizado en Tecnópolis, el cual tuvo una organización muy similar desde el punto de vista de los stands para escuelas, organizaciones, empresas, universidades, e instituciones.

### III Festival de Robótica del Gran Río Cuarto

El Tercer Festival de Robótica del Gran Río Cuarto formó parte de la Semana Innovadora 2018 organizada por la Mesa TIC. El Festival se realizó el día 26 de Septiembre en la Sociedad Italiana de Río Cuarto, ubicada en el centro de la Ciudad de Río Cuarto. El propósito fue incentivar, promover y estimular en los jóvenes de las distintas instituciones de nivel secundario del Gran Río Cuarto y Región el interés por la programación, la robótica y las nuevas tecnologías.

En la apertura de la olimpiada estuvo presente el Rector de la UNRC, Roberto Rovere, quien realizó una encendida defensa de la educación pública, libre gratuita y de calidad, como derecho y herramienta de transformación social. Además, mencionaron aspectos relacionados con el desfinanciamiento educativo e invitaron a todos los participantes a seguir formándose y continuar sus estudios en la Universidad.

El Festival se organizó en tres ejes principales:

- Talleres de robótica y programación
- Primera Olimpiada de Robótica y Programación del Gran Río Cuarto
- Stands tecnológicos, informativos y educativos

En la organización general colaboraron estudiantes universitarios de las carreras de Computación e Ingeniería, y estudiantes secundarios del IPET N° 26 "Juan Filloy". Los estudiantes participaron como talleristas y como veedores en la olimpiada.

- Talleres de programación y robótica educativa:

Participaron más de 200 estudiantes en los 6 talleres simultáneos desarrollados en doble horario. Asistieron 13 escuelas: IPEM N° 28 "Villa de la Concepción de Río Cuarto", IPEAyT N° 186 "Capitán Luis Castagnari" de Las Higueras, IPEMyT N° 257 "Dr. René Favalaro de Laboulaye", Escuela ProA, orientación Biotecnología - sede Río Cuarto, Instituto Santa María Teresa Goretti, Instituto Dante Alighieri, IPEMyT N° 203 "Dr. Juan Bautista Dichiará", Escuela ProA, orientación en Desarrollo de Software - sede Río Cuarto, Colegio San Ignacio, Colegio Santa Eufrasia, Colegio La Merced, Colegio Nuestra Señora del Carmen, y el IPET N° 314 "Libertador General Don José de San Martín".

- Primera Olimpiada de Robótica y Programación del Gran Río Cuarto.

Participaron 45 estudiantes de 8 escuelas: IPEM N° "28 Villa de la Concepción de Río Cuarto", IPEAyT N° 186 "Capitán Luis Castagnari" de Las Higueras, IPEMyT N° 257 "Dr. René Favalaro de Laboulaye", Escuela ProA, orientación en Biotecnología - sede Río Cuarto, Instituto Santa María Teresa Goretti, Instituto Dante Alighieri, IPEMyT N° 203 "Dr. Juan Bautista Dichiará", y Escuela ProA, orientación en Desarrollo de Software - sede Río Cuarto.

La Olimpiada tenía dos partes, una que no requería utilizar conceptos de programación y otra que sí. En los primeros 7 desafíos se evaluaba la parte de "hardware", diseñando soluciones con kits Arduino (encender leds sin programar, encender leds conectados en serie y en paralelo, alarma). Luego se pasaba a 16 desafíos de programación de dichas placas, donde utilizaban el entorno de programación por bloques Ardublock que se venía usando en los talleres (encender leds, parpadear leds, mover servo-motor, alarma de celular, limpiaparabrisas, código morse, semáforo, barrera, pinza, botón, radar, luz nocturna).

Se formaron grupos heterogéneos, conformados por estudiantes de diferentes centros educativos. La Medalla de Oro fue compartida por resultar empatado el primer puesto. Los grupos ganadores estuvieron integrados por estudiantes del IPEM N° 28 "Villa de la Concepción de Río Cuarto", IPEAyT N° 186 "Capitán Luis Castagnari" de Las Higueras, IPEMYT N° 257 "Dr. René Favalaro" de Laboulaye, Escuela ProA, orientación en Biotecnología - sede Río Cuarto, Instituto Santa María

Teresa Goretti de Río Cuarto. La medalla de bronce correspondió a estudiantes del IPEM N° 28 "Villa de la Concepción de Río Cuarto", Escuela ProA, orientación en Biotecnología - sede Río Cuarto e Instituto Santa María Teresa Goretti de Río Cuarto. Cada uno de las/os ganadores de la medalla de oro recibieron como premio un robot Otto, dron terrestre bípedo construido por el equipo organizador. Las/os ganadores de la medalla de bronce recibieron un kit Arduino cada uno. La idea de los premios fue estimular el trabajo y la creatividad con tecnología libre.

- Stands.

Hubo 12 stands con muestras de trabajos tecnológicos-educativos desarrollados por escuelas secundarias de Río Cuarto y la región, empresas y Facultad de Ingeniería de la UNRC. Además de un espacio de co-working del Cluster Tecnológico Río Cuarto.



Figura 8: escenarios

#### IV Festival de Robótica del Gran Río Cuarto

El Cuarto Festival de Robótica del Gran Río Cuarto formó parte de la Semana Innovadora 2019 organizada por la Mesa TIC del Gran Río Cuarto. Se realizó el día 22 de Octubre en el Centro Cultural Viejo Mercado de la Ciudad de Río Cuarto.

El propósito del Festival fue incentivar, promover y estimular en los jóvenes de las distintas instituciones de nivel secundario del Gran Río Cuarto y Región el interés por la programación, la

robótica y las nuevas tecnologías.

El Festival se destinó a estudiantes de escuelas públicas secundarias del Gran Río Cuarto y la región, intentando una regionalización mayor a la registrada en los Festivales anteriores. Para garantizar que las escuelas de la zona pudieran participar, el Festival se organizó en dos momentos: mañana y tarde, con actividades similares. Participaron 18 escuelas, 8 de la Ciudad de Río Cuarto y 10 del Gran Río Cuarto y la región (Santa Catalina -Estación Holmberg-, Las Higueras, Coronel Baigorria, Alcira Gigena, Adelia María, General Deheza y Laboulaye). Asistieron 99 estudiantes, acompañados por 23 docentes de las siguientes escuelas: CITER (Centro Integral de Trabajo, Educación y Recreación) "Ramón Artemio Staffolani" de Las Higueras, Escuela Normal Superior José Manuel Estrada de Alcira Gigena, IPEA N° 215 "Raúl Scalabrini Ortiz" de Santa Catalina (Estación Holmberg), IPEA N° 105 "Alfonsino Storni" de General Deheza, IPEA N° 292 "Agr. Liliam Priotto" de Adelia María, IPEAyT N° 186 "Capitán Luis José Castagnari" de Las Higueras, IPEM N° 27 "René Favalaro" de Río Cuarto, IPEM N° 281 "Dr. Carlos A Lucero Kelly" de Río Cuarto, IPEM N° 91 "Rita Argüello" de Coronel Baigorria, IPEM N° 28 "Villa de la Concepción del Río Cuarto" de Río Cuarto, IPEM N° 95 "Mariquita Sánchez de Thompson" de Río Cuarto, IPET N.º 258 "Mayor Ingeniero Francisco de Arteaga" de Las Higueras, IPET N° 26 "Juan Filloy" de Río Cuarto, IPET N° 314 "Libertador Gral. Don. José de San Martín" de Río Cuarto, IPETyM N° 257 "Dr. René Favalaro" de Laboulaye, PIT (Programa de Inclusión/Terminalidad de la Educación Secundaria para Jóvenes de 14 a 17 años) IPET 26 de Río Cuarto, Escuela ProA, orientación en Biotecnología - sede Río Cuarto y Escuela ProA, orientación en Desarrollo de Software - sede Río Cuarto.

Se montaron cuatro (4) escenarios de trabajo para estudiantes y uno (1) de demostración.

- Escenarios de trabajo para estudiantes:
  - Escenario N6
  - Escenario OTTO
  - Escenario EDUKIT10
  - Escenario KITS ARDUINO
- Escenario Demo: DRONES y AUTOBOTS

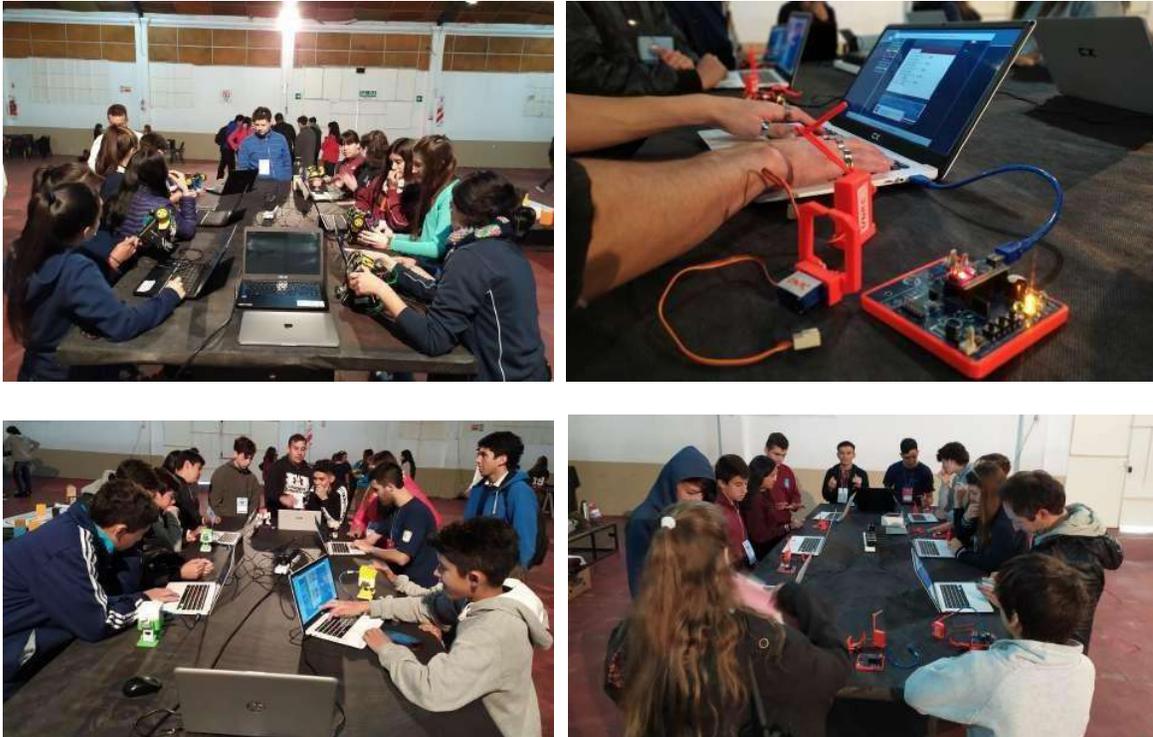


Figura 9: escenarios

## V Festival de Robótica del Gran Río Cuarto

El Quinto Festival de Robótica del Gran Río Cuarto tuvo lugar en el año 2022, luego de la pandemia. En ésta oportunidad fue organizado por la Secretaría de Extensión y Desarrollo de la UNRC, en el marco del Programa de Fortalecimiento de Extensión Universitaria financiado por la Secretaría de Políticas Universitarias del Ministerio de Educación de la Nación, con la participación, la Mesa TIC del Gran Río Cuarto, el IPET N° 26 "Juan Filloy", en el marco de su 25 aniversario, el equipo de trabajo del Proyecto AMULEN, el Punto Digital Río Cuarto y la Fundación por la Cultura de la Municipalidad de la Ciudad de Río Cuarto.

Se realizó el 07 de diciembre de 2022 en las instalaciones del Punto Digital Río Cuarto, ubicado en Centro Cultural del Andino de la Ciudad de Río Cuarto. Estuvo destinado a las/los 150 estudiantes de escuelas secundarias que están participando durante el año 2022 del "Taller Nivel 1 - Aprender programación con placa educativa Edutik10", perteneciente a los Talleres de programación y robótica – 7ta edición.

Tuvo como eje principal la participación en cuatro (4) escenarios rotativos y uno (1) de demostración.

- Escenarios de trabajo para estudiantes:
  - Escenario PICHE
  - Escenario OTTO
  - Escenario EDUKIT10
  - Escenario KITS ARDUINO
- Escenario Demo: DRONES y AUTOBOTS

Como cierre del Festival, Pedro Ducanto y Fernando Sassatelli, Secretario de Extensión y Desarrollo de la UNRC y Presidente de la Fundación por la Cultura, agradecieron a las/os estudiantes participantes, destacaron la importancia de asumir el dominio de la tecnología como Políticas de Estado y de contar con un fuerte sistema de educación pública que garantiza la educación como un derecho humano y lo sostiene con financiamiento estatal. Luego de finalizado el trabajo en los escenarios se realizó un show artístico a cargo de la Murga Mulato Mulé. El show combinó música y baile con participación de diferentes tipos de robots. Finalmente se realizó la entrega de certificados.



Figura 10: escenarios

## Conclusiones

A lo largo de sus cinco ediciones, los Festivales han captado la atención del sector TIC local, la Universidad, y los Gobiernos Municipales de la Ciudad y región. Dentro del ámbito educativo tuvieron excelente repercusión y generaron mucho entusiasmo. Uno de los motivos fue que no existía un ámbito que posibilitara a las/os jóvenes acceder a este tipo de tecnología de manera libre y gratuita. Las opiniones positivas de estudiantes, docentes y directivos de nivel secundario, funcionarios municipales, provinciales y universitarios siempre animó a equipo organizador a redoblar la apuesta. A modo de ejemplo podemos mencionar lo expresado por Agustín, estudiante de séptimo año del IPET N° 258 "Francisco de Arteaga", asistente al Primer Festival, quien celebró la iniciativa y mencionó que *"nos permite interactuar con otros chicos, aprender de programación, guiándonos en el camino sobre qué nos gusta hacer"*.

El Sexto Festival de Robótica del Gran Río Cuarto tendrá características similares al Quinto respecto a los escenarios, y al primero respecto a la metodología. Durante el festival se formarán dos grupos, el primero trabajó dentro del Punto Digital Río Cuarto con cuatro escenarios robóticos que tendrán diferentes desafíos. El segundo grupo lo hará en el Museo Municipal Héroes de Malvinas donde se proyectará una película relacionada a la robótica. Tanto el Punto Digital como el Museo están ubicados en el Centro Cultural del Andino de la Ciudad de Río Cuarto. Terminadas las actividades, los grupos rotarán. Para finalizar se reunirá a ambos grupos, se realizará una puesta en común y se entregarán los certificados de participación al ritmo de un número artístico.

En esta oportunidad la organización está a cargo del equipo de trabajo del proyecto AMULEN, el Punto Digital Río Cuarto, la Fundación por la Cultura de la Municipalidad de la Ciudad de Río Cuarto y el IPET N° 26 "Juan Filloy". Están convocados a participar las/los 200 jóvenes que actualmente concurren a los "Talleres de programación y robótica. Drones Terrestres" que también se desarrollan en las instalaciones del Punto Digital Río Cuarto.

## Bibliografía

- Benítez Larghi, S. (2020). *Desafíos de la inclusión digital en Argentina. Una mirada sobre el Programa Conectar Igualdad*. En Revista de Ciencias Sociales, DS-FCS, vol. 33, n.º 46, pp. 131-154. Universidad de la República. Uruguay.
- Macchiarola, V; Martini, C; Montebelli, A; Mancini, A. (2018). *Inclusión digital educativa en escuelas secundarias argentinas. Un estudio evaluativo*. En Revista Ciencia, Docencia y Tecnología, vol 29, n° 46, pp. 149-175. ISSN 1851-1716.
- Monsalves González, S. (2011). *Estudio sobre la utilidad de la robótica educativa desde la perspectiva del docente*. En Revista de Pedagogía, 32(90), 81-117.
- Willging, P. A., Astudillo, G. J, Bast, S., Occelli, M., Castro, L., & Distel. J. (2017). *Educación con Tecnologías: la Robótica Educativa Aplicada para el Aprendizaje de la Programación*. Memorias WICC 2017, XIX Workshop de Investigadores en Ciencias de la Computación, RedUNCI, 1174-1178. ISBN: 978-987-42-5143-5.

## Documentos

Aportes de Cultura Digital en la Educación Tecnológica (2022). Recuperado el 20 de septiembre de 2023 de:

<https://www.igualdadycalidadcba.gov.ar/SIPEC-CBA/Programas/2022/CulturaDigital/Docs/cultura-digital.pdf>

Núcleo de Aprendizajes Prioritarios de Educación Digital, Programación y Robótica. Educ.ar (2018). Recuperado el 20 de septiembre de 2023 de:

<https://www.educ.ar/recursos/150123/nap-de-educacion-digital-programacion-y-robotica?from=86>

Orientaciones pedagógicas de Educación Digital. Ministerio de Educación de la Nación (2017). Recuperado el 20 de septiembre de 2023 de:

<https://www.educ.ar/recursos/132262/orientaciones-pedagogicas-de-educacion-digital>

Programación y robótica: objetivos de aprendizaje para la educación obligatoria. Ministerio de Educación, Cultura, Ciencia y Tecnología de la Nación (2017). Recuperado el 20 de septiembre de 2023 de:

<https://www.educ.ar/recursos/132339/programacion-y-robotica-objetivos-de-aprendizaje-para-la-educacion-obligatoria>

Resolución CFCyE N° 225/04 (2004). Recuperado el 20 de septiembre de 2023 de:

<https://cfe.educacion.gob.ar/resoluciones/res04/225-04.pdf>

Resolución Consejo Federal de Educación N° 263/15 (2015). Recuperado el 20 de septiembre de 2023 de: <https://cfe.educacion.gob.ar/resoluciones/res15/263-15.pdf>

*Resolución Consejo Federal de Educación N° 343/18 (2018)*. Recuperado el 20 de septiembre de 2023 de: [https://www.argentina.gob.ar/sites/default/files/res\\_cfe\\_343\\_18\\_0.pdf](https://www.argentina.gob.ar/sites/default/files/res_cfe_343_18_0.pdf)

## Enlaces

<https://huayra.educar.gob.ar/>

<https://program.ar/>

<https://www.lariberaweb.com/2018/09/28/mas-de-200-estudiantes-participaron-del-iii-festival-de-robotica-del-gran-rio-cuarto/>

<https://www.telediaridigital.net/2019/10/con-la-participacion-de-18-escuelas-se-realizo-el-cuarto-festival-de-robotica-del-gran-rio-cuarto/>

[https://www.unrc.edu.ar/unrc/n\\_comp.cdc?nota=32467](https://www.unrc.edu.ar/unrc/n_comp.cdc?nota=32467)

<https://www.youtube.com/watch?v=yuJqFJ8CK5o>

<https://www.facebook.com/Robotica-Educativa-UNRC-960760764029834/>

<https://lashiguera.gov.ar/finalizo-la-semana-innovadora-con-el-festival-de-robotica-del-gran-rio-cuarto/>

<https://www.facebook.com/Robotica-Educativa-UNRC-960760764029834/>

<https://www.instagram.com/puntodigitalriocuarto/>

<https://www.instagram.com/roboticaeducativaunrc/>

[https://www.youtube.com/watch?v=yuJqFJ8CK5o \(UniRio TV\)](https://www.youtube.com/watch?v=yuJqFJ8CK5o)

## **Enseñar Ciencias de la Computación en la escolaridad obligatoria. Aportes para un didáctica específica**

Magdalena Garzón<sup>1</sup>, Alfredo Héctor Sanzo<sup>1,2</sup>  
{mgarzon, asanzo}@fundacionsadosky.org.ar

<sup>1</sup>Iniciativa Program.AR  
Fundación Sadosky

<sup>2</sup>Instituto de Ciencias de la Computación  
Universidad de Buenos Aires - CONICET

### **Resumen**

El artículo se inscribe en la tradición para la construcción de una didáctica específica de las Ciencias de la Computación (CC) que en nuestro país ya tiene más de diez años de desarrollo. El presente estudio toma como punto de partida las premisas y horizontes de la propuesta curricular desarrollada por la Fundación Sadosky que plantea un modo de pensar una enseñanza de las CC humanista y emancipadora en pos de formar ciudadanas y ciudadanos que puedan tomar decisiones argumentadas en sus prácticas computacionales y una mirada crítica tanto a nivel individual como social.

Contamos con un enfoque y propósitos claros sobre los cuales continuar construyendo una didáctica de las CC. Pero, ¿en qué consiste la construcción de una didáctica específica? ¿Qué puede aportar la comunidad académica? Para responder a estas preguntas se recurre a los aportes teóricos sobre la naturaleza del aprendizaje, la didáctica general y otras didácticas disciplinares. A partir de allí, se propone un recorrido por tres componentes didácticos: la identificación de conceptos centrales, el desarrollo de recursos didácticos y el conocimiento sobre las concepciones y preconcepciones de estudiantes para visibilizar y poner a disposición referencias teóricas, estudios y el análisis de material didáctico.

**Palabras clave:** didáctica específica, ciencias de la computación, recursos didácticos, representaciones, modelos teóricos, analogías, metáforas, preconcepciones

### **Bases para la construcción de una didáctica específica**

En los estudios didácticos se reconoce una tríada conformada por un docente, estudiantes y los saberes o conocimientos (en un contexto escolar en una época y cultura determinada) que, aunque se representa en una metáfora de triángulo sencilla, cada uno de sus lados involucra una complicadísima trama de relaciones entre sus componentes.



Figura 1. Metáfora inicialmente propuesta por Jean Houssaye (1988) citado en Camilloni, A. (2014).

Este modelo teórico permite explicar por qué la enseñanza no puede ser indiferente ante la opción entre diversas concepciones de aprendizaje. Cada teoría de la cognición supone una concepción sobre cómo se aprende que tiene sus derivaciones en cómo enseñar, en el lugar que se le otorga al saber, y cómo debe darse la relación entre los tres procesos procesos (Litwin, 2005). Asumiendo un acuerdo con teorías cognitivas de base constructivista aprendemos por un proceso de reestructuración. Es decir el aprendizaje es un proceso de construcción de significados a partir de experiencias. La enseñanza entonces se tendrá que ocupar, por ejemplo, de conocer las estructuras previas de sus estudiantes para planificar el andamiaje necesario ya que un nuevo saber puede ser comprendido en la medida en que se conecta con otras ideas ya construidas.

Por otro lado, en sus estudios sobre la formación docente Shulman (1986), un psicólogo educativo dedicado a la enseñanza advierte que cuando se transmite conocimiento didáctico como una lista de recomendaciones, reducidas a lo esencial para simplificar la complejidad de sus principios, esto redundará en un entendimiento incompleto, poco flexible y sin margen para la acción. Por este motivo plantea lo indispensable de recurrir a los marcos teóricos y conceptuales como andamiajes intelectuales argumentando que es justamente el detalle, la historicidad y el contexto de cómo se construyeron esos principios lo que permite que se conviertan en conocimiento que puede ser utilizado en situaciones de práctica auténtica.

Una de las tareas de la comunidad académica dedicada a la construcción de una didáctica de las CC que se deriva de estas premisas es la necesidad de introducirse en el campo de conocimiento de la didáctica general para encontrar las perspectivas didácticas más compatibles con la naturaleza del conocimiento que se enseña y los fundamentos para establecer un orden, relaciones y sentidos propios (Souto, 2014).

No obstante, la construcción de toda didáctica específica, lejos de supeditarse en una relación jerárquica o dependiente a otros campos de dominio se nutre en una relación dialógica y

recíproca de las producciones teóricas de la **didáctica general** y de otras didácticas específicas (W. de Camilloni, 2007; Souto, 2014). La **didáctica** de la **matemática** y de las **ciencias**, que vienen forjando su identidad como disciplina desde hace varias décadas<sup>1</sup> resultan insumos muy interesantes para la didáctica de las CC que ameritan una profunda revisión de la comunidad académica. Se trata de construir conocimiento, como proponen Wollovick y Martínez (2016) en la sinergia entre las Ciencias de la Educación y la Computación.

En síntesis, recurriendo a una metáfora familiar en el ámbito de las Ciencias de la Computación, podemos pensar la construcción del conocimiento didáctico como un modelo de capas. Estudios o experiencias que devienen en teorías del aprendizaje son la capa inicial a partir de las cuales se derivan principios para la enseñanza, otra capa. Grupos de teorías que constituyen enfoques didácticos, una nueva capa construida sobre la base de la anterior. A su vez los enfoques didácticos, derivan en estrategias de enseñanza. En el marco de la didáctica de las CC por ejemplo, tanto Seymour Papert, como luego Mitchel Resnik, y Tim Bell, por citar algunos pioneros en la enseñanza de nociones del ámbito de las CC, recuperan aportes del aprendizaje por descubrimiento (acuñado por Jerome Bruner en 1966) y de la pedagogía auténtica<sup>2</sup>, que a su vez se sostienen en los aportes de la Escuela Nueva<sup>3</sup>, alineados a la psicología constructivista de Jean Piaget<sup>4</sup> y la teoría sociocultural de Lev Vygotsky. Como dice Furman (2023) tenemos muchos *hombros de gigantes* sobre los cuales erigir una didáctica específica.

### ¿En qué consiste la construcción de una didáctica disciplinar?

Desde la didáctica de la matemática, Chevallard (1990) propuso el concepto de **transposición didáctica** como el trabajo que desde la didáctica de una disciplina se realiza para transformar contenidos de saberes “sabios”, en el sentido que provienen de la disciplina académica, a un objeto a enseñar preciso (designado para formar parte de un currículum) y éste en un objeto de enseñanza (tratamiento didáctico). Este autor advierte sobre la distancia obligatoria que separa al primer eslabón (objeto del saber) del segundo (el objeto a enseñar) que es a veces sustituido o reconvertido para constituir el tercer eslabón (el objeto de enseñanza). Entre los ejemplos de esta trasposición que menciona Chevallard se encuentra el proceso que parte de la teoría de los

---

<sup>1</sup> Para profundizar en clave historiográfica sobre el desarrollo de estas disciplinas puede consultarse Contreras Oré, F. (2012). La evolución de la didáctica de la matemática. *Horizonte de la ciencia*, 2(2), 20-25 y Adúriz-Bravo, A., Izquierdo Aymerich, M. (2002). Acerca de la didáctica de las ciencias como disciplina autónoma. *REEC: Revista electrónica de enseñanza de las ciencias*, 1(3), 130-140.

<sup>2</sup> Véase Newmann, F. M., Marks, H. M., & Gamoran, A. (1996). Authentic Pedagogy and Student Performance. *American Journal of Education*, 104(4), 280–312. <http://www.jstor.org/stable/1085433>

<sup>3</sup> La Escuela Nueva representa a un movimiento pedagógico surgido a finales del siglo XIX e inicios del XX que buscaba rever las formas tradicionales de la enseñanza en su época. Algunos de sus referentes principales son John Dewey desde Estados Unidos, María Montessori desde Italia, Ovide Decroly en Bélgica, Celestin Freinet desde Suiza, Antón Makarenko en Rusia, Juana Manso en Argentina.

<sup>4</sup> El término constructivismo se origina en un texto de Piaget (1927) “La construcción de lo real en el niño”.

conjuntos de los matemáticos (objeto de saber matemático) a la designación de la teoría de conjuntos para ser abordada en la escuela primaria (objeto a enseñar) a los diagramas de Venn (una creación didáctica que se constituye en objeto de enseñanza).

Esta noción de reconstruir una disciplina científica para poder ser enseñada, fue adoptada por la didáctica general y otras didácticas específicas como un aspecto central. Cada disciplina específica encuentra sus propios argumentos para las decisiones involucradas en esta reconstrucción. Por ejemplo, desde la didáctica de la Ciencias Naturales se recupera este concepto de Chevallard, y definen sus marcos y condiciones para pensar la transposición didáctica como la acción de “crear y modelar problemas en términos de elementos y operaciones (dominio del problema o elementos y operaciones del dominio) a partir de los cuales los estudiantes pueden razonar a través de modelos. Esto es pensar como “científicos” usando “modelos teóricos.” (Izquierdo-Aymerich y Adúriz-Bravo, 2003, p. 40).

En este sentido, no solo se trata de designar los saberes de las CC a enseñar que se señalan en un currículum, sino de convertirlos en objeto de enseñanza. La didáctica de las CC se debe a sí misma un **análisis de las transposiciones didácticas** que se proponen y que restan construir.

Con otro recorrido, en un contexto donde la formación docente se dedicaba a enseñar la disciplina por un lado y los principios pedagógicos por otro, Shulman (1986) señala que la “enseñabilidad” de un contenido va más allá del conocimiento de la propia disciplina. Al tipo de conocimiento particular que se construye en la interacción del conocimiento experto de la disciplina y del conocimiento didáctico, Shulman lo denomina *conocimiento didáctico del contenido* que caracteriza así:

[el conocimiento de] “...los temas que se enseñan con más regularidad, las formas más útiles de representar esas ideas, las analogías, modos de ilustrar, ejemplos, explicaciones y demostraciones más poderosas- en una palabra, las formas de representar y formular el tema que lo hacen comprensible para los demás [...] También incluye la comprensión de qué hace que el aprendizaje de ciertos temas sea fácil o difícil: las concepciones y preconcepciones que los alumnos de distintas edades y contextos traen con ellos para el aprendizaje de los temas (..) Si esas ideas preconcebidas son creencias erróneas, como ocurre a menudo, los profesores necesitan conocer las **estrategias** que pueden ser más fructíferas para reorganizar la comprensión de los alumnos. (p. 9).

Con los componentes que surgen de esta caracterización se propone una revisión de los aspectos implicados en cada uno y un análisis de ejemplos o estudios desarrollados en el ámbito de las Ciencias de la Computación

## Identificar conceptos y habilidades centrales

En términos generales, la definición de un currículum implica per sé un recorte, una selección de conceptos y habilidades para ser enseñadas. Siguiendo a Chevallard (1990) representa el paso de la definición del *objeto a enseñar* frente a todos los *objetos de saber* académico. Esta transposición es más que definir los títulos de los temas, implica identificar los conceptos estables e indispensables y las habilidades constitutivas de la disciplina. Aquellos que poseen un mayor nivel de generalidad que mantienen unidos y habilitan múltiples conexiones con varios conceptos subordinados (Rollnick et.al, 2017).

Cuando se piensa la enseñanza de una disciplina específica y se busca ofrecer recomendaciones para enseñar mejor, es necesario además hacer explícita la concepción de la disciplina que la subyace y los propósitos que guían su formación (Martinez, C et. al 2022), ya que ésta es la que otorga sentido a la opción por determinados enfoques de enseñanza y coherencia a las decisiones didácticas.

En el campo de las Ciencias de la Computación, en una época donde muchos países están creando o redefiniendo sus diseños curriculares (Bocconi, et.al 2016; Martinez, C. et. al, 2021), en donde convive una falta de claridad conceptual de la disciplina en los espacios educativos (Unesco, 2019; Schapachnik, 2021) resulta clave la claridad y la precisión. Bell (2018) y Denning (2010) advierten sobre la necesidad de facilitar la comprensión sobre los conceptos a desarrollar al conjunto de docentes, no siempre formados en la disciplina. Con esta premisa proponen partir de grandes ideas o ideas clave que permitan conservar la visión de conjunto sin perderse en detalles técnicos y operativos.

En la Argentina contamos con avances en este sentido. Para establecer los saberes que la escuela debe enseñar, la propuesta curricular elaborada por la Fundación Sadosky<sup>5</sup> (Dabbah, et.al, 2023) propone además otros dos criterios especialmente pertinentes en el campo de las CC. Se focaliza en conceptos que promueven **justicia curricular**, es decir que permiten democratizar la participación ciudadana en el mundo computacional y achican diferentes brechas. Y también en conceptos que permiten asumir una **perspectiva crítica** de la existencia, el uso y la producción de la tecnología computacional.

En este marco, en la propuesta curricular citada se encuentran argumentos contundentes para proponer un **enfoque humanístico y emancipador** a partir del cual se identifican los conceptos centrales para abordar la enseñanza de las Ciencias de la Computación en la escolaridad

---

<sup>5</sup> Para no reproducir en este artículo los fines y argumentos se sugiere remitirse al sitio web <https://curriculum.program.ar> y al artículo "Hacia un currículum emancipador de las Ciencias de la Computación" de Martinez, et.al 2002 disponible en <https://revistalataam.digital/article/220322/> que profundiza en los diferentes paradigmas que guiaron los debates sobre la inclusión de las CC en el currículum.

obligatoria.

**Desarrollar recursos didácticos (modelos teóricos, analogías significativas, metáforas poderosas, representaciones, ejemplos).**

Cuando desde la Psicología educativa o desde la Ciencia Cognitiva, se habla de “representación”, se refiere a la idea de que apropiarse de cualquier aspecto de la realidad supone representarlo a nivel interno, es decir, construir un modelo mental de esa realidad (Izquierdo, 1999; Medina, 2008 y Velez 2008 citados en Portela et.al, 2022). Pensando en la enseñanza, Shulman (1986) destaca la necesidad de buscar formas de representación de las ideas y conceptos de una disciplina de un modo en que puedan ser comprendidos por otros. Dicho autor señala que “como no existe una única forma de representación poderosa, el docente debe tener a mano un verdadero arsenal de formas de representación alternativas, algunas de las cuales se derivan de la investigación, mientras que otras tienen su origen en la sabiduría de la práctica” (p.9). Para favorecer representaciones desde la didáctica se recurren a representaciones concretas, modelos teóricos, analogías y metáforas.

**Representaciones concretas**

En términos de Galagovsky y Adúriz Bravo (2001), referentes argentinos en la didáctica de las Ciencias Naturales, una representación concreta, refiere a:

[...] representaciones visuales de ciertas imágenes asociadas a algún modelo científico en particular; pueden ser dibujos, proyecciones bidimensionales u objetos tridimensionales. El referente de este tipo de representación es también un concepto científico, reconstruido mediante artificios pictóricos, generalmente simplificadores del concepto más complejo. Ejemplos de este tipo serían: el dibujo de un orbital, el esquema de una célula. (p. 236)

En las CC existen innumerables ejemplos de representaciones concretas: diagramas de red, de contexto, de flujo, de objetos, de clases, de arquitectura, de compuertas lógicas, de estados, autómatas, etc. En la didáctica específica de la disciplina, uno de los espacios en donde se juegan las representaciones es en la construcción de entornos de programación. Como señala Gómez (2020) el desarrollo de lenguajes y entornos para la enseñanza de la programación permitieron promover su enseñanza en la escolaridad obligatoria. Las decisiones que se adoptan al crear un nuevo lenguaje de programación y su correspondiente entorno de desarrollo integrado (IDE) para enseñar son parte del último eslabón de la transposición didáctica, son objeto de enseñanza. Ejemplo de cómo impacta el concepto de representación en un IDE es la coherencia con la que en el lenguaje Gobstones<sup>6</sup> (Martínez Lopez 2013) se decidió representar

---

<sup>6</sup> <https://gobstones.github.io/>

la diferencia conceptual que existe entre expresiones (elementos del lenguaje que *denotan valores*) y comandos (que denotan *acciones*) a través de la **forma** y el color.



Figura 2: Representación en el Gobstones IDE: Todos los comandos son azules y tienen conexiones arriba y abajo. Todas las expresiones son verdes y tienen conexiones sólo a izquierda.

Un caso de representación confusa es la elegida en la biblioteca Blockly para crear y modificar parámetros. En ella, los comandos y los parámetros tienen formas muy similares a pesar de ser conceptos radicalmente diferentes.



Figura 3: Representaciones confusas en Blockly: el comando imprimir y el parámetro (o “entrada”) cantidad tienen la misma forma.

En Argentina existe un acuerdo en la comunidad de la didáctica de las CC que se refleja en la construcción de sus IDE fundamentado no solo en la importancia de las representaciones sino también en la gradualidad de lo que se muestra a las y los estudiantes.

### Modelos teóricos

En su propuesta de una didáctica de la ciencia escolar de Izquierdo- Aymerich y Adúriz- Bravo (2003) señalan:

[Es necesario que] los modelos teóricos escolares se ajusten al mundo de los alumnos y, por tanto, tengan un significado para ellos, manteniendo como objetivo final el conocimiento científico aceptado. Si los modelos científicos escolares no chocan con las analogías y las metáforas, sino que por el contrario concuerdan con ellos, el lenguaje científico escolar adquiere la profundidad necesaria para convertirlo en un instrumento de comunicación y comprensión. En las antípodas de esta propuesta solemos encontrar una ciencia escolar que impone teorías científicas nulas, sin relación con los fenómenos que salen de los libros de texto y atiborradas de un lenguaje

científico excesivamente preciso, que hay que aprender de memoria. (p.40).

La didáctica de las CC comparte con su par de las Ciencias Naturales, el propósito de presentar una ciencia que se acerque al conocimiento dinámico sobre el mundo atravesado por computadoras, más que a una tecnología sofisticada y oscura (Dabbah et.al, 2023).

Un ejemplo valioso de las CC es el abordaje del modelo de máquina programable. Esto es, comprender -en una primera instancia- que una computadora es una máquina programable que recibe información, la procesa (realiza operaciones, comparaciones, modificaciones, o la almacena) y genera nueva información (una acción, una imagen, un texto) sobre la base de un programa. Este modelo, que sostiene la visión de sistema y permite abordar la arquitectura de las computadoras, se constituye como saber necesario a partir del cual se puede avanzar hacia la arquitectura de von Neumann, a comprender cómo los componentes de esta arquitectura participan en la ejecución de un programa<sup>7</sup>, hacia un modelo que subyace a las técnicas de aprendizaje automático, a problematizar el derecho a reparar<sup>8</sup> y el impacto ambiental, es decir, habilita múltiples puertas de entrada y diferentes grados de complejidad para su enseñanza en distintos niveles escolares. Lo valioso es que aunque se simplifique para ajustarse al mundo de los alumnos, como sostiene Izquierdo y Adúriz (2003) esta transposición didáctica no choca con la vigilancia epistemológica<sup>9</sup> necesaria que advierte Chevallard (1990). Se dispone de un interesante tratamiento didáctico de esta noción en la colección de Manuales de Ciencias de la Computación para el aula y otras propuestas más recientes<sup>10</sup>, así como en propuestas de enseñanza elaboradas por el ISEP de Córdoba<sup>11</sup>.

## Analogías

Portella y otros (2022) en un estudio sistemático sobre el uso de analogías como recursos

<sup>7</sup> Gisela Confalonieri propone un abordaje de este tema para la secundaria en la secuencia didáctica [¿Qué es lo que hace más rápida a mi computadora?](#)

<sup>8</sup> Nicolas Wollovick y Cristián Rojo, hacen una interesante propuesta para abordar el derecho a reparar llevada a la práctica en sucesivas oportunidades en la secuencia didáctica [Desarmar para aprender](#), publicada por la Fundación Sadosky.

<sup>9</sup> La vigilancia epistemológica, según Chevallard, “es la herramienta que permite al didacta analizar el saber tal como es enseñado, interrogar las evidencias, cuestionar las ideas simples, desprenderse de la familiaridad engañosa de su objeto de estudio, aún sabiendo [...] que el objeto deberá haber sufrido ciertas deformaciones que lo harán apto para ser enseñado.”(1990;16).

<sup>10</sup> En <https://repositorio.curriculum.program.ar/> se pueden encontrar las secuencias didácticas de toda la colección clasificadas por su contenido y otras propuestas recientes. Para introducir el modelo de máquina programable en el 1º ciclo de primaria, en el capítulo 6 del Manual [Ciencias de la Computación para el aula. 1º ciclo primaria](#), se encuentran diversas propuestas. La propuesta [Les quitamos el velo a las computadoras](#) para abordar en 2º ciclo de primaria; el modelo de computadora como punto de partida para introducir el Aprendizaje automático en [¿Qué necesita la IA para ser IA?](#) y [Un modelo de aprendizaje automático por dentro: entre datos y opiniones](#) en nivel secundario.

<sup>11</sup> El ISEP de Córdoba en su sitio <https://hacemosescuela.cba.gov.ar/> publica propuestas para abordar el modelo de máquina programable en diferentes niveles. En nivel inicial, [“Computadora ¿sos vos?”](#), [El Celular. una computadora de bolsillo](#). Para el primer ciclo de primaria [Los componentes y aplicaciones de la computadora](#), [Las computadoras de nuestro entorno](#).

didácticos recurren a la perspectiva de la Psicología cognitiva (Gentner, 1983; Holyoak, 1984 citados en Portella, 2022). Concluyen que las analogías deben tener las suficientes correspondencias entre sus elementos, sus roles y relaciones para poder formular inferencias que permitan transferir de la situación conocida por las y los estudiantes a la situación nueva que se busca enseñar. En el mismo sentido, desde la didáctica de las Ciencias Naturales, Galagovsky y Adúriz-Bravo (2001), proponen el concepto de *análogo concreto* para referirse a las analogías que apelan a conceptos de significación ya conocida por los alumnos. Los autores destacan que la calidad de una analogía está también en su capacidad de **suscitar discusiones ricas sobre las similitudes y diferencias** con el modelo científico asociado para favorecer reflexiones metacognitivas. La adjetivación de “concreto” la remiten a la teoría piagetiana, en referencia al “pensamiento operatorio concreto”<sup>12</sup>. Según estos autores, la clave está en las estrategias didácticas que busquen que sean los estudiantes quienes hacen explícitas las relaciones entre la analogía y el contenido disciplinar. A este abordaje lo denominan “Modelo didáctico analógico”.

Desde la didáctica de las CC un ejemplo rico para suscitar discusiones es la analogía del viaje de un paquete por correo postal<sup>13</sup> para introducir la capa física del diseño de internet.



Figura 4: Analogía: envío de un paquete por correo postal y envío de un mensaje de Whatsapp

Por un lado, el mecanismo de envíos de paquetes físicos está más relacionado al pensamiento operatorio concreto dentro de la realidad de las y los estudiantes, y por el otro, permite suscitar discusiones sobre las similitudes entre el paquete y la información, entre los edificios de distribución y las antenas, entre los cables y las camionetas, etc. y las diferencias entre las velocidades en ambos casos, la cantidad de paquetes en ambos casos, el salto internacional en el caso de internet, etc.

Un ejemplo de analogía actualmente en debate, es el tratamiento didáctico dado al término de

<sup>12</sup> Piaget establece una serie de estadios del desarrollo cognitivo, desde el estadio sensoriomotor, el preoperacional, el de las operaciones concretas hasta el estadio de las operaciones formales.

<sup>13</sup> Un ejemplo de esta analogía puede verse en el programa [¿Por donde viajan los whatsapps?](#) de Canal Encuentro, realizado en el marco del Programa Seguimos educando (2020).

algoritmo. Una analogía muy difundida ha sido la de las instrucciones para reproducir actividades de la vida cotidiana con la noción de algoritmo. Esta analogía buscaba transmitir la importancia del orden de las instrucciones en un algoritmo, con la consecuente equiparación de evocar una



serie de pasos para realizar cualquier tarea (Schapachnik y Bonello, 2022).

Figura 5: Ejemplo de analogía en debate: los pasos para atarse los cordones en Ciencias de la Computación para el aula - Manual para docentes. 2° segundo ciclo primaria <http://bit.ly/2uWUDij>

En principio, podemos decir que el análogo se encuentra correctamente cercano a las significaciones conocidas por las y los estudiantes, y hay espacio para cierta discusión sobre la similitud entre los pasos de una tarea y las instrucciones de un programa. Sin embargo, la comunidad ha encontrado los límites de esta analogía. En principio, su enseñanza en la escuela supone a las y los estudiantes poner un nuevo nombre (algoritmo) a la noción de textos instructivos que vienen construyendo desde nivel inicial sin que este cambio de terminología aporte ningún valor adicional al concepto de secuencia de pasos. De hecho la asociación con un contenido tan familiar en el contexto escolar, puede explicar su rápida pregnancia en el ámbito educativo y la dificultad para diferenciar el análogo del nuevo concepto. Esto fue señalado por Schapachnik y Bonello (2022) al advertir sobre “el riesgo de trivializar la explicación y que la transposición didáctica deje de dar cuenta del objeto de estudio real que se pretende abordar” (p.47). Es que pensar un algoritmo exclusivamente como una serie de pasos presenta por lo menos dos problemas. Por un lado, no permite discutir la capacidad de un algoritmo de proponer una única solución para resolver múltiples instancias de un problema. Por ejemplo, un algoritmo de ordenamiento puede operar con muchas listas distintas, pero la serie de pasos para atarse los cordones no considera esa multiplicidad. Por otro lado, rompe con el modelo de la computación como sistema o transformación de información: los pasos hablan de un proceso, pero ¿dónde está la información?, ¿dónde está la entrada?,

¿dónde la transformación de información? ¿dónde la salida?. Al no existir suficientes similitudes no se pueden suscitar discusiones interesantes con las y los estudiantes.

### Metáforas

Gracias a Lakoff y Johnson (1989) pioneros de la Lingüística Cognitiva (citados en Portela et.al, 2022), “la metáfora pasa a ser considerada como un mecanismo cognitivo, que impregna no sólo

el lenguaje, sino también el pensamiento y la acción” (p. 92). Wigdorovitz de Camilloni (2014) analiza el uso de las metáforas como recursos didácticos y las define como “modos de operación mental que sobre la base de una analogía iluminan una de las interpretaciones que se proponen y logran constituir una elucidación profunda del aspecto mentado en la metáfora” (p.19). En este sentido, las metáforas constituyen una herramienta que puede favorecer representaciones internas. Al igual que en las analogías, se busca posibilitar conexiones entre lo que las y los estudiantes conocen para permitir la proyección de conceptos desde un dominio accesible y familiar hacia otro nuevo. Estas proyecciones se concretan en la búsqueda de correspondencias, es decir en poner de manifiesto las relaciones analógicas entre ambos dominios. (Portella, et.al, p.87). Cuando utilizamos metáforas como recurso didáctico, una vez que se hace explícito el significado de lo que interesa proyectar al nuevo concepto, tiene el poder de convertirse en el objeto en su nuevo contexto de significación.

Existe un género de metáforas que pueden ser representadas visualmente para ser descritas y revisadas. En las CC, utilizamos por ejemplo la metáfora del árbol de directorios.

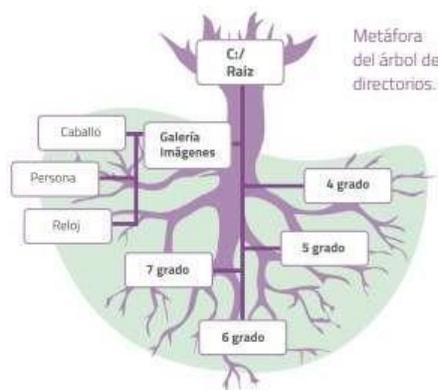


Figura 6: Metáfora de un árbol de directorios en SD “Archivos, carpetas y almacenamiento”<sup>14</sup>.

En esta metáfora, se entremezcla la idea estructural de un árbol (elemento de la naturaleza) con la idea jerárquica de organización de un sistema de archivos, y tanta es la correspondencia que en la propia disciplina se utiliza la palabra “raíz”. La potencia de este tipo de metáforas estará nuevamente en la explicitación y el trabajo sobre las proyecciones. Por ejemplo mediante las preguntas ¿Por qué se dice “carpeta raíz”?

¿Cuáles son las ramas en la compu? ¿Y de qué juegan los archivos? ¿El árbol está al derecho o al revés?

<sup>14</sup> <https://repositorio.curriculum.program.ar/repositorio-diseno-curricular/archivos-carpeta-y-almacenamiento/>

### **Conocimiento de las concepciones y preconcepciones de estudiantes.**

El conocimiento didáctico del contenido, según Shulman (1986), incluye la comprensión de qué temas de la disciplina son más difíciles o más fáciles de aprender. Como esta dificultad no es intrínseca de cada tema de la disciplina, sino que depende de las concepciones y preconcepciones que los alumnos de distintas edades y contextos traen con ellos para el aprendizaje de los temas, surge la necesidad de conocerlas.

Gracias a la psicología constructivista sabemos que el sujeto que aprende no es un envase vacío que debe ser llenado con conocimiento, sino que va construyendo significados a partir de su experiencia. A partir de ello, un gran número de teorías didácticas se ocuparon de analizar cómo promover a través de la enseñanza el conflicto cognitivo (término acuñado por Piaget) o el cambio conceptual (en palabras de Astolfi y otros 1998 citados en W. Camilioni, 2007) cuando esas concepciones intuitivas o alternativas son erróneas. El cambio conceptual no es nada sencillo y el modo de promoverlo no es trivial, ya que no corregimos nuestras concepciones sólo con una buena explicación. En este contexto, resultan clave los estudios sobre la concepciones y preconcepciones de las y los estudiantes que permiten construir saber sobre su influencia en el aprendizaje y las condiciones didácticas necesarias para superar y transformar esas concepciones iniciales.

Gran parte de las estrategias didácticas más difundidas se apoyan en este paradigma. De allí surgen, por ejemplo, dos principios didácticos. Por un lado la importancia de recuperar las ideas previas de los alumnos antes de comenzar un tema. Por otro lado, las estrategias buscan que las y los estudiantes hagan explícitos y argumenten los pasos que dan para resolver un problema, y así, dar cuenta de su proceso de pensamiento y sus concepciones iniciales.

En el ámbito de las CC, en un estudio llevado adelante por Mertala (2019) al indagar en las concepciones de niñas y niños de 6 años, encontraron cuestiones como que una laptop es el único ejemplo de computadora, mientras que otros dispositivos como tablets (que les podrían resultar también familiares), no son vistos como computadoras. Basados en este estudio por ejemplo una derivación para la enseñanza es lo valioso de abordar el modelo de máquina programable en los primeros años de la escolaridad.

En otro estudio llevado adelante por Parra y otros (2021) entre las las misconceptions que se indagaron a estudiantes de 10 años, se concluye que los estudiantes *piensan a "La nube" como un espacio de almacenamiento "etéreo" de datos más que un medio por el cual se envía y recibe información*. En estudios con adolescentes (Kodama et al. 2017), frente a la pregunta "¿cómo funciona por detrás Google?", concluye que en los mejores casos se incluyen dibujos y explicaciones de cables y el edificio de una compañía, pero casi nunca aparecen menciones a

bases de datos o algoritmos de búsqueda. Este tipo de estudios, nos da la pauta de la necesidad de abordar en el aula las representaciones que la metáfora de “**la nube**” habilita, mostrando lo tangible de los cables submarinos y los data center para favorecer comprensiones de cómo internet es una red de computadoras interconectadas que intercambian datos entre ellas<sup>15</sup>.

Otro aspecto importante al abordar la CC en la escuela, es la necesidad de desnaturalizar la idea intuitiva sobre la objetividad, que deviene de una concepción de la tecnología computacional basada en que las máquinas no tienen intereses, ni prejuicios, “no se equivocan”. Al respecto, Schapachnik y Bonello (2022) proponen llevar al aula los debates técnico-sociales sobre los sesgos algorítmicos, la neutralidad de la red y las plataformas en simultáneo con los conocimientos técnicos sobre los que se sustentan. Por ejemplo, en un inicio haciendo hincapié en la presencia de personas encargadas de escribir los programas que las máquinas ejecutan, luego en las personas que establecen sobre qué conjunto de datos se entrenará a un modelo de aprendizaje automático<sup>16</sup>, también considerando los intereses y motivaciones detrás de la “gratuidad” de algunos servicios.

Por último, es necesario revisar las propias concepciones culturales que tenemos como docentes respecto de lo que es difícil o fácil, y sobre lo que es adecuado o no. Esto es especialmente relevante para atender las **desigualdades de género**. Un estudio llevado adelante por Echeveste et al. (2021) señala que “las estudiantes muestran una confianza y autopercepción de eficacia que es menor a la de los estudiantes varones, considerando que ‘me altero porque no me sale, entonces que se encarguen los chicos que son los que más saben’”. En el mismo sentido, Labhart (2023) concluye que “la menor exposición de las mujeres al uso de las computadoras genera menor seguridad en su vínculo con las CC, en especial en culturas en donde el acceso a juguetes y videojuegos están atravesados por prejuicios culturales de género”. Esta sensación de inseguridad y falta de confianza, permite explicar las autopercepciones sobre la computación como algo difícil. Albert Bandura (1977, citado en Furman, 2021), un psicólogo dedicado a la educación, muestra en sus investigaciones el modo en el que el sentido de autoeficacia, entendida como la confianza en las propias posibilidades, afecta el comportamiento. Furman lo sintetiza de un modo muy claro “sentirnos autoeficaces hace que abordemos las tareas difíciles como desafíos que podemos resolver, en lugar de amenazas que debemos evitar.” (pp.102-3).

---

<sup>15</sup> Una propuesta didáctica que aborda esta noción en segundo ciclo de primaria se puede encontrar en la secuencia [El viaje de la información por internet](#), desarrollada por la Fundación Sadosky para el Plan Ceibal y adaptada para el contexto argentino.

<sup>16</sup> Un abordaje interesante de esta noción se propone en la secuencia [Un modelo de aprendizaje automático por dentro: entre datos y opiniones](#), publicada en el sitio [curriculum.program.ar](http://curriculum.program.ar)

## Conclusiones

Fruto del análisis realizado en torno al desarrollo de modelos teóricos, analogías y metáforas, se advierte un denominador común para la construcción de estos recursos didácticos: Resulta irrenunciable la necesidad de indagar las concepciones previas de las y los estudiantes de diferentes edades y contexto. En términos de la psicología cognitiva, las significaciones que construyeron a partir de sus experiencias y que traen para el aprendizaje de nuevos conceptos. Por otro lado, el análisis sobre la construcción de representaciones, sugiere apoyarse en objetos concretos para construir a partir de ellos conceptos abstractos. Y en cuanto al desarrollo de modelos teóricos la importancia de que, aún respetando el principio de gradualidad, mantengan como objetivo final el conocimiento científico aceptado.

A su vez, se evidencian lineamientos para continuar en la construcción de una didáctica específica de las CC. El recorrido busca abrir una línea de trabajo que apela a algunos aportes teóricos de la didáctica general, de la didáctica de las matemáticas y de las ciencias naturales como insumos para construir conocimiento didáctico de las CC.

Otra línea de trabajo de la comunidad académica que podría resultar de interés es la construcción de las transposiciones didácticas necesarias para promover los saberes a enseñar identificados en la Propuesta curricular de la Fundación Sadosky. Al respecto se señalan dos cuestiones a considerar. Una, que las nociones y conceptos que tienen sentido y se comprenden en un contexto informático, requieren de una cuidadosa transposición didáctica que contemple el contexto escolar que maneja otros significados para los mismos términos. Esto se señaló a propósito de comprender un algoritmo exclusivamente como una secuencia de pasos, pero puede generalizarse a la resolución de un problema en términos generales y sus diferencias con la resolución de un problema computacional, o las diferencias que existen en la significación del concepto de abstracción en el ámbito educativo y en el ámbito de la programación. Otro riesgo a considerar es que aunque se simplifique un concepto para ajustarse al mundo de los alumnos, la transposición didáctica debe dar cuenta del objeto de estudio que se pretende abordar. Es decir realizar una vigilancia epistemológica, en términos de Chevallard.

Como señala Shulman, el conocimiento didáctico también se construye a partir de la sabiduría de la práctica docente, pero éstas prácticas muchas veces permanecen implícitas o invisibilizadas. En este sentido, el campo de la didáctica de las CC tiene como desafío visibilizar y poner a disposición experiencias docentes valiosas que pueden aportar nuevos saberes sin ocultar las bases en las que se apoya para construir ese conocimiento.

## Referencias

- Bell, T., Tymann, P., y Yehudai, A. (2018). *The Big Ideas of K-12 Computer Science Education*. EATCS, 124, 1-18.
- Bocconi, S., et. al, (2016). *Developing computational thinking in compulsory education - Implications for policy and practice*, JRC Science for Policy Report.
- Bonello, B., Schapachnik, F. (2020). [10 preguntas frecuentes \(y urgentes\) sobre Pensamiento Computacional](#). Revista Virtualidad, Educación y Ciencia, 20(20-2020), 156-167.
- Chevallard, Yves (1990) *La transposición didáctica. Del saber sabio al saber enseñado*. Editorial Aique.
- Dabbah J., Garzón M., Gómez M., Martínez C., Martínez López P. E. (2023) [Propuesta curricular para la inclusión de las Ciencias de la Computación en la educación obligatoria de Argentina](#). Fundación Sadosky.
- Denning, P. J. (2010). [The great principles of computing](#). American Scientist, 98(5), 369-372.
- Echeveste, E., Gómez, M., Martínez, C, Benotti, L. (2021). *La escuela y la brecha de género en la enseñanza de las Ciencias de la Computación*. Actas JADiCC 2021, pp 21-30.
- Furman, Melina (2021). *Enseñar distinto. Guía para innovar sin perderse en el camino*. Colección educación que aprende. Siglo veintiuno editores.
- Galagovsky, L. R., & Adúriz-Bravo, A. (2001). [Modelos y analogías en la enseñanza de las ciencias naturales. El concepto de modelo didáctico analógico](#). Enseñanza de las Ciencias. Revista de investigación y experiencias didácticas, 19(2), 231-242.
- Gómez, M. J. (2020). [Aspectos de adquisición de lenguaje en la enseñanza de programación](#) [Tesis de doctorado, Universidad Nacional de Córdoba].
- Izquierdo, M. et.al (1999), *Caracterización y Fundamentación de la Ciencia Escolar*, Enseñanza de las Ciencias, Special issue, 79–92.
- Izquierdo-Aymerich, M., Adúriz-Bravo, A. (2003). *Epistemological Foundations of School Science*. Science & Education, 12(1), 27-43.
- Kodama, C., et.al (2017). *There's a creepy guy on the other end at Google!: engaging middle school students in a drawing activity to elicit their mental models of Google*. Information Retrieval Journal, 20, 403-432.
- Labhart, Mariana (2023). *¿Cómo y por qué enseñar Ciencias de la Computación desde una perspectiva de género en las escuelas?* Fundación Sadosky.
- Litwin, Edith (2005). *La tecnología educativa en el debate didáctico contemporáneo*. Paidós.
- Martinez, C. y Borchardt, M. (2021). [Enfoques y perspectivas didácticas globales en la enseñanza de la computación](#). Simposio Argentino de Educación en Informática (SAEI) en el marco de las Jornadas Argentinas de Informática Virtuales (JAiVO).

Anales de SAEI 2021.

- Martínez, C., et.al. (2022). [Hacia un currículum emancipador de las Ciencias de la Computación](#). Revista Latinoamericana de Economía y Sociedad Digital, Agosto(3), 1-25.
- Martínez Lopez, Pablo E (2013). *Las bases conceptuales de la Programación: Una nueva forma de aprender a programar*.  
[http://inpr.web.unq.edu.ar/el-libro-de-gobstones/?dl\\_id=35](http://inpr.web.unq.edu.ar/el-libro-de-gobstones/?dl_id=35)
- Mertala, P. (2019). *Young children's conceptions of computers, code, and the Internet*. International journal of child-computer interaction, 19, 56-66.
- Parral L., Schinca H., Schapachnik F., Czemerinski H.(2021) [Misconceptions de Ciencias de la Computación en niños/as escolarizados/as](#). Buenos Aires: Departamento de Computación, Universidad de Buenos Aires, Fundación Sadosky. JADiCC 2021.
- Portela, M. P., Salic, M. A., Ceccacci-Sawicki, L., Fernández, C., Olguín, M. V. (2022). [Las analogías y las metáforas como recursos para la enseñanza y aprendizaje: una revisión sistemática](#). Perspectivas En Psicología, 19, 84-106.
- Rollnick, M., Mavhunga, E. (2017). [Pedagogical Content Knowledge](#). International Encyclopedia of Education, Third Edition, January, 656-661.
- Schapachnik, F. (2021) [Programación. Robótica. Ciencias de la Computación. Despejando la ensalada epistemológica que la Informática trajo a la Educación](#). IV Congreso Municipal de Educación, Secretaría de Educación de la Municipalidad de Córdoba.
- Schapachnik, F y Bonello, M.B (2022) *Ciencias de la computación en la escuela. Guía para enseñar mucho más que a programar*. Colección educación que ladra. Siglo veintiuno editores.
- Shulman, L. S. (1986). *Those Who Understand : Knowledge Growth in Teaching*. Educational Researcher, 15(2), 4-14.
- Souto, M (2014), *Didáctica general y didácticas especiales* en Malet, A.M y Monetti (comp) Aportes para la discusión en Debates universitarios acerca de lo didáctico y la formación docente Colección Universidad, Noveduc.
- UNESCO (2019). *Coding, Programming and the Changing Curriculum for Computing in Schools*. Reporte UNESCO/IFIP TC3 Meeting at OCCE – June 2018, Linz, Austria.
- Wigdorovitz de Camilloni, A. R. (2007). *El saber didáctico*. Colección. Cuestiones de educación. Editorial Paidós.
- Wigdorovitz de Camilloni, A. R. (2014). [Las metáforas conceptuales en la construcción del discurso pedagógico](#). Revista de Educación, 7, pp17-32.
- Wolovick, N., & Martínez, M. C. (2016). [Enseñar a Programar y Programar para Aprender](#). Virtualidad, Educación Y Ciencia, 7(12), pp. 8–10.

## Hola Mundo. Tus primeros pasos en programación

Jorge Rodriguez <sup>1</sup>, Silvana Hubinsky <sup>2</sup>, Viviana Sanchez <sup>1</sup>, Guillermo Guerrero <sup>1</sup>,  
Valeria Flores <sup>1</sup>, Aldo Matamala <sup>1</sup>, Federico Amigone <sup>1</sup>, Celeste Ramos <sup>1</sup>,  
Pablo Kogan <sup>1</sup>, Jorge Navarro <sup>3</sup>, David Torchinsky <sup>1</sup>

j.rodrig@fi.uncoma.edu.ar, funyder.diplo@gmail.com, viviana.sanchez@fi.uncoma.edu.ar,  
guillermo.guerrero@est.fi.uncoma.edu.ar, valeria.flores@fi.uncoma.edu.ar,  
aldo.matamala@fi.uncoma.edu.ar, fede.amigone@fi.uncoma.edu.ar,  
celeste.ramos@fi.uncoma.edu.ar, pablo.kogan@fi.uncoma.edu.ar, jorge.navarro@fi.uncoma.edu.ar,  
david.torchinsky@est.fi.uncoma.edu.ar

<sup>1</sup> Facultad de Informática  
Universidad Nacional del Comahue

<sup>2</sup> Fundación de la Universidad Nacional del Comahue para el Desarrollo Regional  
Universidad Nacional del Comahue

<sup>3</sup> Consejo Provincial de Educación  
Ministerio de Educación de la Provincia de Neuquén.

### Resumen

La escasez de programadores representa un desafío significativo con implicaciones que trascienden el ámbito tecnológico. Abordar esta problemática requiere esfuerzos colaborativos de la educación, la sociedad y los gobiernos para asegurar un abastecimiento adecuado de profesionales en programación y para adaptarse a un entorno tecnológico en constante evolución.

El presente trabajo trata de evidenciar una metodología implementada en la provincia de Neuquén en el año 2022, tendiente a fortalecer el desarrollo regional en el ámbito tecnológico, a partir de una oferta académica en línea que permita a personas jóvenes construir habilidades básicas en el área de algoritmos y programación que promueva la opción por profesiones relacionadas a la computación. La propuesta busca brindar herramientas necesarias para desarrollar piezas de software simples. Además, dar más elementos para los empleos del futuro, pensar proyectos de vida, intervenir el mundo, resolver problemas con pensamiento lógico (pensamiento computacional) y ciudadanía digital.

El resultado neto de esta experiencia confirma la vitalidad del tema como área de vacancia en razón de que su implementación genera habilidades que refuerzan las condiciones de empleabilidad e incentivan a quienes participan a la prosecución de otros trayectos formativos, entre los que podrían incluirse a algunos de carácter académico, lo cual requiere garantizar de manera progresiva una base más amplia de participación y nuevas estrategias de retención.

**Palabras clave:** Programación, diplomatura, educación tecnológica, aprendizaje virtual.

## Introducción

En la década del 90, con la llegada de las computadoras personales a la vida diaria, la enseñanza de computación en las escuelas se enfocó en el uso de herramientas del tipo planilla de cálculo y procesadores de texto, tendencia que aún hoy continúa en Argentina e incluso en el resto del mundo (Sadosky, 2013; Royal Society, 2017). Este tipo de enseñanza enfocada en la ofimática (enseñanza para la oficina) se basa en la visión de esa época que consideraba que en el futuro habría una diferencia en el campo laboral entre quien supiera usar una computadora y quien no.

Hoy en día, cuando la tecnología es parte de la vida diaria de niñas, niños y jóvenes desde sus primeros años, y usar las aplicaciones casi no les representa desafíos, la profundidad de las TIC propicia el desarrollo de capacidades a través del aprendizaje de programación, bajo el título de Pensamiento Computacional. Dichas capacidades son: la generación de modelos mentales, la abstracción, la generalización, la identificación, la identificación de patrones y formulación de problemas y la prueba de soluciones a través de un algoritmo, además de habilidades blandas como la reflexión sobre la propia tarea (autocrítica) y el trabajo colaborativo (Denning, 2019; Wing, 2006)

Por otro lado, hoy en día la industria y la economía necesitan contar con profesionales especializados para apoyar su crecimiento, pero el sector informático se enfrenta a una alta escasez de profesionales. De acuerdo a cifras del Ministerio de Educación, en Argentina la tasa de inscripción en carreras universitarias de computación está estancada hace aproximadamente 15 años, con alrededor de 20.000 ingresantes por año y sólo 3600 personas egresadas, lo cual representa la mitad de los profesionales que la industria requiere (Rabosto, 2019). Esto no ocurre sólo en Argentina: también Estados Unidos y los países de Europa Occidental, entre otros, enfrentan el mismo desafío (Uman, 2017)

Como parte de las estrategias realizadas para fomentar el aumento de las inscripciones a carreras informáticas a nivel nacional, la Fundación Sadosky en conjunto con un colectivo de Universidades Nacionales impulsa el programa “Vocaciones en TIC”, del cual la Facultad de Informática de la Universidad Nacional del Comahue participa, cuyo objetivo es que estudiantes de escuelas secundarias tengan una primera experiencia real programando computadoras. De este programa a nivel local, han participado más de 30000 estudiantes en talleres, involucró la visita a más de 605 escuelas secundarias y articuló esfuerzos con más de 60 universidades públicas de la Argentina (Llambi, 2023).

Además, con objetivos similares, los últimos años la Facultad de Informática ha llevado adelante varios proyectos y actividades de extensión que fomentan la enseñanza de programación en los últimos años de las escuelas secundarias de la región.

A escala provincial, la propuesta se enmarca en el Plan Quinquenal de Desarrollo de la Provincia del Neuquén 2019 – 2023 lanzado por el gobierno provincial donde se sintetizan, en un futuro

deseable y a mediano plazo, políticas, planes, programas y proyectos, que generarán los resultados necesarios para alcanzar la provincia que se planifica.

Como parte de estas acciones, durante el año en curso, el gobierno de la provincia del Neuquén firmó un convenio con CFI<sup>1</sup> donde se acordó la realización de acciones conjuntas para llevar adelante el Programa de Reducción de la Brecha Digital Educativa a efectos de llevar adelante la ejecución de distintas iniciativas para garantizar la inclusión digital en la provincia del Neuquén.

El objetivo general del Plan es promover el desarrollo de la sociedad-economía del conocimiento con el objetivo de impulsar territorios inteligentes, que promueva la capacidad de innovación de las personas y las organizaciones, el apoyo de las tecnologías digitales para hacer más eficientes las ciudades e interconectar a los actores territoriales, a fin de mejorar las condiciones y la calidad de vida de los territorios y sus habitantes.

La presente iniciativa se enmarca en una política provincial destinada a la reducción de la brecha digital y al desarrollo de la economía del conocimiento, buscando generar, en el largo plazo, un polo de Formación, Capacitación y Desarrollo con base en la Ciencia, la Tecnología y la Innovación.

### **Contexto**

En el mundo actual, la programación se ha convertido en una habilidad esencial que impulsa la innovación y el progreso en todos los aspectos de la sociedad. Desde la creación de aplicaciones móviles hasta la automatización de procesos industriales, la programación está en el corazón de las soluciones tecnológicas. Sin embargo, la enseñanza tradicional de la programación a menudo se enfrenta a limitaciones geográficas, de recursos y de acceso, lo que dificulta que un amplio grupo de estudiantes adquieran estas habilidades vitales.

En respuesta a este desafío, la Facultad de Informática de la Universidad Nacional del Comahue, el Consejo Federal de Inversiones, la Fundación de la Universidad Nacional del Comahue para el Desarrollo Regional, la Secretaría de Estado de Planificación y Acción para el Desarrollo y el Gobierno de la Provincia de Neuquén desarrollan de forma colaborativa un programa de enseñanza de programación virtual que busca democratizar el acceso a la educación en programación y empoderar a individuos de todas las edades y antecedentes para que se conviertan en creadores y solucionadores de problemas digitales.

### **Contenido**

Se conforma de un plan a implementar para la realización de las tareas cuyo fin es diseñar, planificar y publicar, los contenidos y las actividades que se van a dictar en el transcurso de la diplomatura. El resultado de la ejecución de cada una de las tareas que a continuación se detallan, va a permitir contar con todo el material y las actividades que se van a realizar en la plataforma de educación a distancia.

---

<sup>1</sup> <https://www.cfi.org.ar/>

N°	Módulo	Hs cátedra
M0	Plataforma del cursado: herramienta, uso y administración	4
M1	Hola Mundo, Conocer el Entorno - Variables, Tipos de	24
M2	Modularización.	24
M3	Estructuras de Control - Alternativas.	24
M4	Estructuras de Control - Repetitivas.	24
M5	Tipo de Datos complejos, Arreglos.	24

Tabla 1: Estructura de contenidos

**El Módulo 0.** Este módulo estuvo dirigido a diseñar los encuentros, armar el material de cada uno de los recorridos, planificar el cronograma con los temas a abordar, organizar los horarios de cada tipo de encuentro, y coordinar la estrategia de organización interna del equipo en función del número de participantes.

Cada módulo incluyó los conceptos teóricos, herramientas de prueba que se pueden utilizar para ir implementando lo aprendido y videos cortos que abordan cada uno de los conceptos. Además, se incluyeron dos secciones de Desafíos: Una, presentó los conceptos aprendidos como herramienta de refuerzo y otra, denominada Desafío Dipli, se corresponde con un proyecto transversal a la Diplomatura, donde los y las participantes trabajaron de manera evolutiva en cada etapa propuesta en los módulos. Finalmente se encuentra la sección de Adicionales en las que, si algún participante quiere realizar más desafíos lo puede hacer, pero no se plantearon como obligatorios.

**El Módulo 1.** Los contenidos para esta etapa de la Diplomatura fueron:

- Hola Mundo. Conocer el Entorno - Variables, Tipos de Datos Básicos, Secuencia
- Presentación del entorno de programación CodeSandbox, entorno online, colaborativo y con soporte al lenguaje de programación HTML y JavaScript. Implementación del primer "Hola Mundo". Entrada y Salida.
- Definición de variables y ámbito de una variable. Tipos de datos básicos de una variable que permiten almacenar números y cadenas de caracteres. Definición de las operaciones e instrucciones de entrada y salida. Operadores Aritméticos, de Asignación, de Comparación y Lógicos.

**El Módulo 2.** Los objetivos de los contenidos para el Módulo 2 "Modularización", refieren a conocer los principios para la implementación de módulos simples que permiten leer, escribir y visualizar

información. La modularización es una práctica para organizar una base de código en partes con acoplamiento bajo y elementos independientes. Cada parte es un módulo. Cada módulo es independiente y tiene un propósito claro.

**El Módulo 3.** Los objetivos de los contenidos para el Módulo refieren a conocer las estructuras de control alternativas, identificación de los problemas a los que brindan solución, su sintaxis y su utilización para la implementación de soluciones.

**El Módulo 4.** Los objetivos de los contenidos para el Módulo 4 refieren a conocer las estructuras de control repetitivas, identificar los problemas a los que brindan solución, conocer su sintaxis y utilizarlas para la implementación de soluciones.

**El Módulo 5.** Los objetivos de los contenidos para el Módulo 5 , refieren a conocer un tipo de estructura que permite almacenar y gestionar una colección de datos de un mismo tipo básico, conocer la definición de Arreglos, identificar las situaciones donde se requiere utilizar la estructura de datos, su sintaxis e implementar soluciones que la requieran.

### **Metodología**

**Actividades áulicas virtuales.** El dictado de los 6 módulos (0-5) de la diplomatura se extenderá durante 5 meses, con actividades sincrónicas y asincrónicas. Cada módulo se desarrollará a partir de encuentros semanales y tendrán una duración de 4 semanas o su equivalente en horas de 24 horas mensuales. Se debe exceptuar el módulo inicial (0) que solo tiene una duración de 4 horas.

Los encuentros se desarrollarán de manera on-line, con actividades sincrónicas o asincrónicas según lo planificado. Se prevé que cada semana cuente por un lado con 2 encuentros sincrónicos: desarrollados los días lunes con una duración de 2 hs y los viernes de 1 hr y por otro lado con 1 encuentro asincrónico: con una duración de 3hs. Cada encuentro será grabado y accesible desde la plataforma educativa que se va a utilizar para el dictado de la tecnicatura. Además el material teórico / práctico correspondiente a cada módulo, también se encontrará publicado y accesible por los y las participantes desde la plataforma.

**Sistematización, registro y evaluación.** Se prevé a lo largo de la diplomatura y de manera semanal realizar actividades evaluativas vinculadas con los contenidos impartidos en ese momento. Al final del curso, los y las participantes procederán a presentar y exponer una

actividad, donde se integran y ponen en práctica, todos los conceptos vistos en cada uno de los módulos.

En cada encuentro se realizará un registro de los y las participantes que se encuentran en la sala, con el fin de llevar un control de asistencia de cada uno de los y las inscriptas.

### **Herramientas utilizadas**

- Zoom<sup>2</sup>: Es una de las principales aplicaciones de software para videoconferencias. Permite interactuar virtualmente con compañeros de estudio cuando no es posible reunirse en persona, y también ha tenido un gran éxito en eventos sociales.
- Discord<sup>3</sup>: Discord es una aplicación gratuita de comunicación que te permite usar chat de voz, video y texto con tus amigos, comunidades de juegos y desarrolladores. Tiene cientos de millones de usuarios, lo que la convierte en una de las formas más populares de conectarse con personas en línea. Discord se puede usar en casi todas las plataformas y dispositivos más populares, entre ellos Windows, macOS, Linux, iOS, Android y navegadores web.
- Codesandbox.io<sup>4</sup>: Es un entorno de desarrollo para aplicaciones web al que se puede acceder con un navegador. No se requiere configuración, por lo que se puede comenzar a usar de inmediato. Los espacios aislados se pueden compartir fácilmente con equipos o compañeros y se pueden crear todos los que se necesiten. Con la funcionalidad de actualizaciones en vivo, los códigos se ejecutan de inmediato, lo que permite ver los resultados directamente en el espacio aislado. Los espacios aislados se pueden vincular a GitHub.
- Página oficial de la diplomatura: Sitio web donde se publican los recorridos a transitar por los facilitadores y participantes durante todo el trayecto de la Diplomatura. Cada módulo fue publicado en la siguiente plataforma donde se fueron habilitando a medida que fue avanzando el recorrido de los mismos, <https://holamundo.fi.uncoma.edu.ar>.
- Quizizz<sup>5</sup>: es un sitio web que nos permite crear cuestionarios online que nuestros alumnos pueden responder de maneras distintas. Permite a los docentes promover espacios para evaluar el aprendizaje de los estudiantes, proporcionar una retroalimentación inmediata a las respuestas entregadas por los estudiantes, y de gamificar en los procesos de enseñanza y de aprendizaje generados en un curso.

### **Destinatarios**

La propuesta de Diplomatura fue dirigida principalmente a jóvenes (mayores de 18 años) que han egresado recientemente o se encuentran en los últimos años del secundario en la Provincia de

---

<sup>2</sup> <https://zoom.us/>

<sup>3</sup> <https://discord.com/>

<sup>4</sup> <https://codesandbox.io/>

<sup>5</sup> <https://quizizz.com/?lng=es-ES>

Neuquén.

### **Inscriptos.**

- Total pre-inscriptos/as: 1724 (mil setecientos treinta y nueve)
- Total pre-inscriptos/as con inconsistencias: 62 (sesenta y dos)
- Total de pre-inscriptos/as que se consideraron para la ponderación 1661(mil seiscientos sesenta y uno)

### **Caracterización de la población acreditada.**

- El proceso de selección, entre otras variables, considera la **paridad de género**.
- La mayoría de las personas acreditadas reside en la microrregión denominada Confluencia<sup>66</sup>, donde se ubica el área metropolitana de la ciudad capital Neuquén
- Rango de edad: La población mayoritariamente alcanzada por el sistema de ponderación, resultó ser el grupo de las edades más jóvenes : entre 18 y 24 años.
- Educación alcanzada: Especial importancia tuvo en la selección la consideración de aquellas personas interesadas en participar que estuvieran cursando su último año de nivel medio y cumplieran 18 años en 2022. Del total, 67 (37%) responden a esta característica.
- Acceso a conocimientos y/o saberes previos en programación.
- Tiempo disponible para estudiar.
- Situación laboral actual: El 87% de quienes se acreditaron, conforman el grupo de participantes que son estudiantes (48%); desempleados (39%); a ellos se suma un 10% aproximado que declararon estudiar y trabajar. El 4% mencionó tener trabajo.
- Acceso a servicios de conectividad: El 8% de los y las acreditadas declaró no contar con dispositivos adecuados.

### **Resultados**

Para tener una información más próxima al análisis de resultados, se proporcionó a los diplomados una encuesta, tanto para los que iniciaron y luego abandonaron, como a los que cumplieron con el 80% de asistencia. Por lo tanto, la observación del comportamiento que asumen las variables

---

<sup>66</sup> La microrregión “Confluencia” se constituye como el espacio más poblado del sur argentino. Respecto a los servicios, concentra la mayor cantidad de instituciones educativas de nivel universitario y los centros de salud de mayor complejidad. Se trata de la región con mayor dinamismo económico. Se destaca por ser el principal centro logístico, comercial y financiero en la provincia y también de la Patagonia. En ella se desarrollan la actividad hidrocarburífera, sus servicios asociados y la administración pública. La Ciudad de Neuquén es la de mayor diversidad de actividades económicas y de equipamiento y la que posee mayor jerarquía urbana. Fuente: <https://www.copade.gob.ar/microrregiones/>

representadas de manera comparativa entre quienes abandonaron y quienes certificaron y aprobaron la Diplomatura, se destaca:

- El recorrido total de la diplomatura registró un 79% de desgranamiento (de 188 acreditados, 40 finalizaron) / aun así, si se considera que las vacantes fueron 100, el porcentaje de aprobados es del 40%.
- Se registra que el 20% de quienes iniciaron y manifestaron ser estudiantes y/o trabajar, abandonaron el recorrido de la Diplomatura.
- En razón del rango de edad, se registra que las personas más jóvenes (18 a 24 años) en su mayoría abandonaron el recorrido.
- Por otro lado también, entre quienes dejaron de participar, se registran casos de abandono previo en los niveles medio(3%), terciarios (5%) y universitarios (11.5%). Sin embargo, en el conjunto de quienes certifican también registran casos de abandono previo (22,5%), por lo tanto esta variable no estaría impactando directamente sobre el proceso de desgranamiento.
- En cuanto al tiempo transcurrido desde la aprobación del secundario; el grupo que registró mayores bajas fue el de los y las jóvenes que se encuentran cursando el último año de estudios en ese nivel, la baja fue del 90%: de 61 personas, 6 certificaron.
- La disminución respecto de la distribución geográfica de quienes iniciaron y quienes finalmente terminaron podría estar indicando que existen factores exógenos a la Diplomatura que impactan en el número de participantes.
- Es importante el acceso a las herramientas óptimas para una conexión virtual, para el desarrollo de las actividades y desafíos propuestos. Quienes no contaban con estos recursos, en particular una computadora abandonaron la Diplomatura (10% de los 148)
- El 65% de quienes completaron la Diplomatura no tenía experiencia de aplicación de lenguajes de programación con anterioridad.
- Se identifica un importante registro respecto al acceso a herramientas de programación recibidas en el cursado del nivel medio de estudios (56% de los acreditados) y también un porcentaje importante de personas que son autodidactas (23%).

## **Conclusiones**

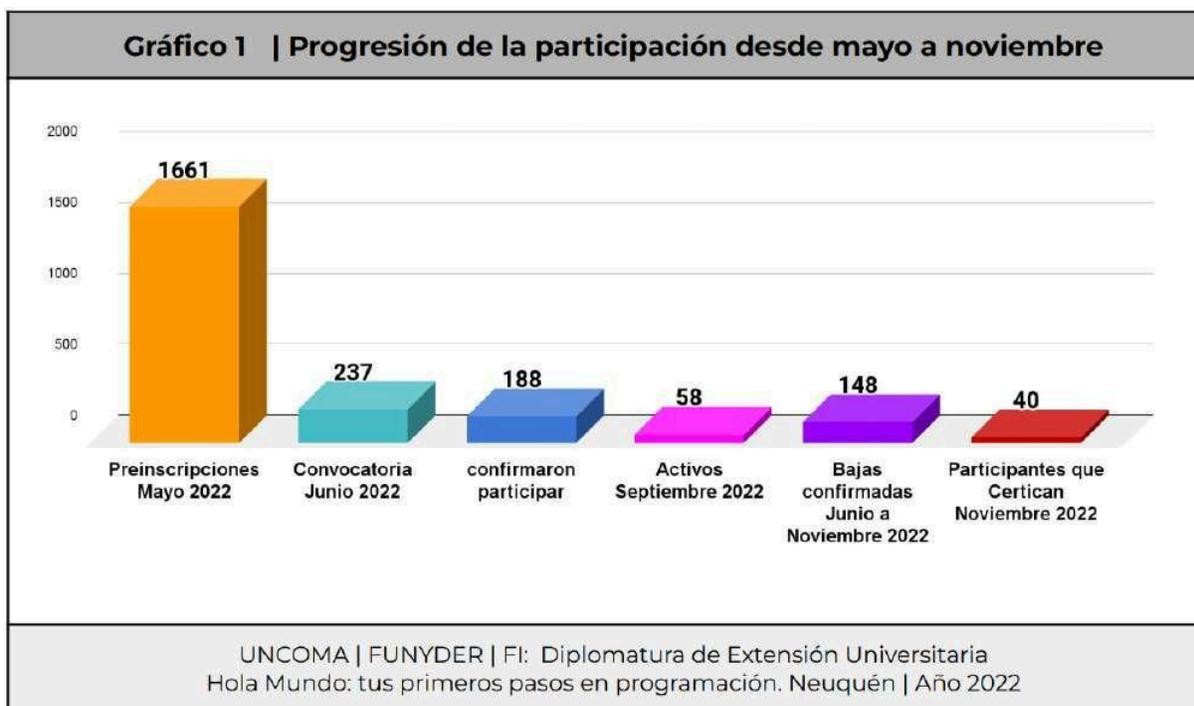
Son reconocidos los procesos de desgranamiento en las matrículas y el problema del sostenimiento de las trayectorias estudiantiles en niveles terciario y universitario en nuestro país. De acuerdo a investigaciones (Alcoba et al., 2019), se trata de fenómenos que generalmente tienen lugar en los tramos iniciales de las carreras; la relevancia de estos problemas impactan en la población, sus capacidades de desempeño personal y otras limitantes que impactan en el mundo del trabajo y los

niveles de desarrollo humano.

Así pues, al momento del cierre del primer informe de avance, uno de los tres informes a presentar, se encontraban participando un total de 110 personas cuya distribución según género autopercebido fue de paridad entre varones (50,9%) y mujeres (49,1%).

Por otra parte, el comportamiento de la distribución geográfica, respondía al comportamiento que también asume la variable “residencia” según la distribución poblacional en la provincia: La mayor parte de los habitantes neuquinos viven en la microrregión denominada Confluencia, donde se ubica el área metropolitana de la ciudad capital Neuquén: 82,6% de los y las participantes residen en esa microrregión, luego con un 17,4% participaban residentes de las microrregiones Este y Centro.

Al momento del cierre del Segundo Informe de Avance se registraron un total de 58 participantes activos; Finalmente, de este grupo 38 participantes han logrado completar el Trayecto de la Diplomatura y certificarse. El siguiente Gráfico (gráfico 1) de barras refleja la progresión desde la fase de preinscripciones hasta las certificaciones.



Se propone comprender el desgranamiento que, en la Diplomatura Hola Mundo, se ha manifestado con el 79%; para ello se considera relevante para este informe presentar gráficamente las características generales de los 148 participantes que confirmaron su participación y abandonaron en algún punto del cursado en contraste con las 40 que sí certificaron y aprobaron.

El orden de presentación de las variables responde a factores (Ruiz Ramirez et al., 2014) que se registran en diversos ámbitos académicos en relación a las causales posibles de abandono y que son sostenidos en diversas investigaciones dedicadas a este y otros temas que preocupan tanto en

nuestro país como así también en Latinoamérica.

Desde este punto de vista y, en el contexto que aborda el informe final, se asume que los factores que impactan en la deserción o no de los y las estudiantes se perciben y manifiestan en cada persona de manera combinada, resultan en aspectos que vale la pena identificar desde las instituciones a los fines de sostener, desde una perspectiva aún más inclusiva de la educación pública.

## Referencias

- Alcoba, J., Garatte, M. L., & Hernando, G. (2019, October 23). *Desgranamiento y dificultades de integración a la vida universitaria en Ciencias de la Educación de la Universidad Nacional de La Plata*. SEDICI. Retrieved August 21, 2023, from <http://sedici.unlp.edu.ar/handle/10915/83878>
- Denning, P. J., & Tedre, M. (2019). *Computational thinking*. Mit Press.
- Furber, S. (2012). Shut down or restart? The way forward for computing in UK Schools. London, England: The Royal Society.
- Llambi, C., Borchardt, M., Klinkovich, V., Locca, N., Martínez, C., & Scasso, M. (2023). Aprendizajes y desafíos para la enseñanza de las Ciencias de la Computación en las escuelas.
- Rabosto, A. N., & Zukerfeld, M. (2019). El sector argentino de software: desacoples entre empleo, salarios y educación. *Ciencia, tecnología y política*.
- Royal Society. (2017). After the reboot: Computing education in UK schools. Policy Report.
- Ruiz Ramirez, R., Ayala Carrillo, M. d. R., & Zapata Martelo, E. (2014, julio 1). *ESTEREOTIPOS DE GÉNERO EN LA DESERCIÓN ESCOLAR: CASO EL FUERTE, SINALOA*. Redalyc. Retrieved August 21, 2023, from <https://www.redalyc.org/pdf/461/46132451012.pdf>
- Sadosky, F. (2013). Una propuesta para refundar la enseñanza de la computación en las escuelas Argentinas. *Reporte de la Fundación Sadosky*, 23.
- Uman, I. (2017, March 2). *Universidad de Buenos Aires*. Universidad de Buenos Aires. Retrieved August 20, 2023, from <https://www.uba.ar/noticia/16788>
- Wing, J. M. (2006). *Computational Thinking* (3rd ed., Vol. 49). Communication of ACM. <https://doi.org/10.1145/1118178.1118215>

## Nuevo contexto en las carreras de informática. ¿Con qué estrategias lo enfrentamos?

Gladys Dapozo, Cristina Greiner, Raquel Petris, Ana María Company, María Cecilia Espíndola  
{dapozo,cgreiner,rpetris,anamacom,mcespindola}@exa.unne.edu.ar

Universidad Nacional del Nordeste

### Resumen

La alta demanda de profesionales informáticos ha llevado a un notable incremento de la matrícula de las carreras de informática en los últimos dos años. En este contexto, la enseñanza de la programación inicial en la universidad ha requerido adecuaciones para mitigar las dificultades de la masividad. En este trabajo se describen los resultados de un conjunto de estrategias aplicadas y se analizan sus fortalezas y debilidades, contrastando con el desempeño final de los estudiantes. Para completar el análisis, mediante una encuesta se recabó la opinión de los estudiantes acerca de las modalidades implementadas. Respecto de la programación por pares, los alumnos opinaron favorablemente sobre la estrategia porque les permitió debatir las propuestas de solución, contar con un compañero de estudio y organizarse mejor para estudiar. Como dificultades señalaron cuestiones logísticas, como acordar lugares y horarios para reunirse a estudiar. El equipo de cátedra considera que esta modalidad tiene como ventaja reproducir situaciones de desempeño laboral, disminuir la numerosa cantidad de trabajos a corregir, favoreciendo la devolución de las correcciones, incrementar el aprendizaje gracias al debate entre los pares, sin embargo, se aprecia que requiere algunos ajustes para un mejor aprovechamiento.

**Palabras claves:** Enseñanza de la Programación. Programación por pares (pair programming), pensamiento computacional.

### Introducción

La Industria IT argentina es uno de los actores de la economía que ha demostrado, en los últimos 30 años, mayor nivel de expansión global en los mercados más exigentes. Por esta razón, es considerada por todos los gobiernos como un sector de importancia estratégica para el desarrollo económico del país<sup>1</sup>. Esta industria demanda permanentemente profesionales del área IT.

En este contexto, las carreras de informática del país han incrementado su matrícula en forma notable, debido a la aceleración de la virtualización provocada por la pandemia, la masificación de la modalidad de trabajo remoto, mejores retribuciones salariales para los trabajadores del

---

<sup>1</sup> <https://redargentinit.com/porque/>

sector. Respecto de este último punto, cabe destacar que los sueldos de la industria del software duplican la Canasta Básica y los salarios del sector privado en general, según informe del Observatorio Permanente de la Industria del Software y Servicios Informáticos de la República Argentina (OPSSI)<sup>2</sup>.

Por otra parte, además del aumento del interés por la formación, se observa también que los estudiantes quieren aprender rápido para empezar a trabajar. Por tanto, se orientan preferentemente al uso de herramientas y minimizan el esfuerzo en las actividades de profundización conceptual y desarrollo del pensamiento computacional que se requiere para ser un buen solucionador de problemas mediante la programación de computadoras.

En este trabajo se describen las distintas estrategias implementadas en una materia de programación del primer año de una carrera de informática que tuvo en el año 2023 más de 1.000 alumnos inscriptos, tres veces más que el promedio de los últimos 5 años. Para lograr los objetivos de aprendizaje el equipo docente ajustó la metodología de enseñanza, incluyendo las siguientes estrategias: a) introducción de los conceptos básicos de programación mediante actividades lúdicas utilizando un método de resolución de problemas, b) programación por pares para el desarrollo de las actividades prácticas que consisten en resolución de problemas codificando con el lenguaje C, c) laboratorios móviles, ante la imposibilidad de continuar con las tradicionales clases en los laboratorios de informática, los estudiantes realizaron las actividades prácticas en aulas comunes utilizando sus computadoras personales o sus móviles, d) evaluación por pares: el primer parcial y su correspondiente recuperatorio fue realizado por pares utilizando sus propias computadoras. El segundo parcial fue realizado en forma individual, utilizando el soporte papel. En esta decisión influyó la aparición de chatGPT.

A continuación, se detallan las distintas estrategias aplicadas y los resultados obtenidos.

## **Estrategias aplicadas**

### **1. Introducción de los conceptos básicos de programación mediante actividades lúdicas utilizando un método de resolución de problemas**

Se diseñó una estrategia de enseñanza con el propósito de mejorar la comprensión de los conceptos básicos de programación que consiste en:

- a) Definición de un método de resolución de problemas
- b) Desarrollo de un conjunto de actividades lúdicas, basadas en la programación por bloques, aplicando el método de resolución de problemas.

---

<sup>2</sup> <https://cessi.org.ar/wp-content/uploads/2023/06/Infografia-Salarios-Software-enero-2023.pdf>

- c) Aplicación del método de resolución de problemas a lo largo del desarrollo de las actividades de la asignatura.

El método de resolución de problemas, basado en [1] combina herramientas conceptuales y herramientas del lenguaje, promoviendo el pensamiento computacional que se requiere para la programación. Consiste en los siguientes pasos: a) Idear una estrategia de solución, y explicitarla; b) Expresar la estrategia mediante una división en bloques o módulos; c) Declarar y nombrar adecuadamente cada bloque de modo que exprese la tarea que realiza; d) Definir cada uno de los bloques que expresan tareas, mediante instrucciones primitivas.

Las actividades diseñadas con este propósito, tomadas de [2], se desarrollaron durante tres semanas con dos clases de tres horas cada una. Se presentaron desafíos que los estudiantes debían resolver utilizando la herramienta lúdica PilasBloques, siguiendo el método de resolución de problemas propuesto. En la Figura 1 se muestra, a modo de ejemplo, la actividad denominada “María la come sandía” y en la Figura 2 el código en bloques de la solución lograda. El desafío consiste en hacer que María coma todas las sandías de la cuadrícula. Para lograr la solución, se solicita a los estudiantes que apliquen el método de resolución de problemas, enfatizando los conceptos de “abstracción”, referido a pensar una estrategia antes de codificar la solución, la “descomposición del problema en partes”, traducida en la creación de bloques, la “detección de patrones” secuencias que se repiten o regularidades que se pueden enseñar a un autómata reutilizar, y en la “legibilidad” de la solución, mediante la definición de nombres representativos de los bloques. Para resolver la actividad ejemplo, se utilizaron herramientas del lenguaje, tales como, primitivas, procedimientos y la estructura de repetición simple.



Figura 1. Actividad “María la come sandía”.

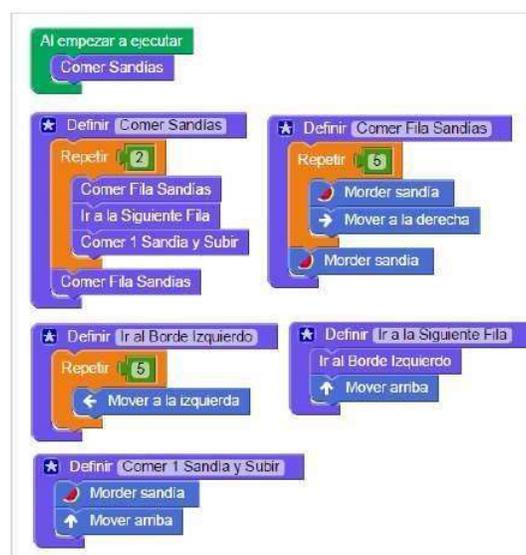


Figura 2. Código en bloque que resuelve el desafío

En el siguiente ejemplo se describen los pasos del método, aplicado en una actividad denominada “Tito enciende las luces”, cuya interfaz se muestra en la Figura 3.

El método comprende dos etapas, que se expresan en una grilla que es proporcionada a los alumnos a efectos de tener una guía para la aplicación del mismo. Estas etapas bien diferenciadas enfatizan también la independencia de la resolución del problema de la posterior implementación en una herramienta de programación determinada (Ver Tabla 1 y 2).



Figura 3. Actividad “Tito enciende las luces”.

La primera etapa involucra el uso de las herramientas conceptuales para resolver un problema de programación. Los pasos se muestran en la Tabla 1, en la columna 1 se indican los pasos lógicos y tareas a realizar, y en la columna 2 el resultado que se obtiene en el caso ejemplo.

Pasos	Ej.: Tito enciende las luces
<ul style="list-style-type: none"> <li>Analizar el problema (Ver el escenario)</li> <li>Determinar el QUE (objetivo)</li> <li>Pensar una solución global</li> </ul>	Encender todas las luces
<ul style="list-style-type: none"> <li>Pensar un nombre para la solución</li> </ul>	EncenderLuces
<ul style="list-style-type: none"> <li>Detectar “patrones” (<b>reconocer regularidades que se pueden enseñar – a un autómatas - y reutilizar</b>)</li> </ul>	Las dos diagonales son iguales
<ul style="list-style-type: none"> <li>Pensar una estrategia</li> <li>Dividir el problema en tareas (procedimientos)</li> </ul>	<ul style="list-style-type: none"> <li>Ubicarse al inicio de la primera diagonal</li> <li>Encender la diagonal</li> <li>Ubicarse al inicio de la segunda diagonal</li> <li>Encender la diagonal</li> </ul>

<ul style="list-style-type: none"> <li>• Pensar nombres representativos para las tareas</li> </ul>	<ul style="list-style-type: none"> <li>• UbicarseDiagonal_1</li> <li>• EncenderDiagonal</li> <li>• UbicarseDiagonal_2</li> <li>• EncenderDiagonal</li> </ul>
--	--

Tabla 1: Herramientas conceptuales para la resolución de problemas

La segunda etapa se muestra en la Tabla 2. Consiste en la aplicación de las herramientas del lenguaje para obtener una solución ejecutable en alguna herramienta de programación. En la columna 1 se muestran los pasos a realizar y en la columna 2 la resolución de la actividad utilizando la herramienta PilasBloques. En el primer paso se define un bloque general de solución denominado “EncenderLuces” (1) que contiene los bloques que representan la estrategia de solución (2).

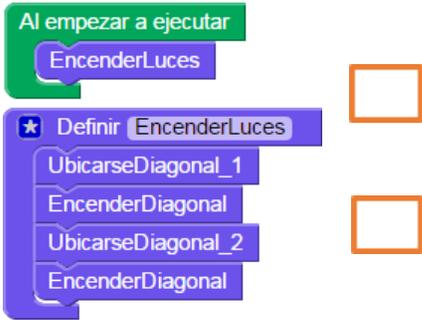
En el segundo paso se desarrolla (define) cada bloque que conforma la solución, en base a las primitivas, necesarias para cumplir el objetivo (3). Las primitivas son las acciones predefinidas en la herramienta.

En el último paso se verifica el logro del objetivo (Encender todas las luces), ejecutando la solución.

De este modo se van incorporando las distintas estructuras, respetando los nombres de los bloques y estableciendo la secuencia lógica de ejecución, y no menos importante, dividiendo el problema principal en partes más pequeñas (módulos) y fomentando el reuso.

Una ampliación de la aplicación de esta estrategia se detalla en [3].

Esta estrategia está en consonancia con investigaciones recientes que concluyen que la programación visual y el aprendizaje basado en juegos pueden mejorar el pensamiento computacional y las habilidades de resolución de problemas en los estudiantes y pueden usarse para introducirlos en la programación [4], [5].

Pasos	Ej.: Tito enciende las luces
Escribir la solución global en la herramienta	 <p>The image shows a Scratch script for a function named 'EncenderLuces'. It starts with a green flag icon 'Al empezar a ejecutar' block, followed by a purple 'EncenderLuces' block. Below this is a 'Definir EncenderLuces' block containing a sequence of four blocks: 'UbicarseDiagonal_1', 'EncenderDiagonal', 'UbicarseDiagonal_2', and 'EncenderDiagonal'. To the right of the code blocks are two orange square boxes representing lights.</p>

<p>Desarrollar (definir) cada bloque</p> <ul style="list-style-type: none"> <li>• Analizar las primitivas</li> <li>• Determinar el <b>COMO</b></li> <li>• Armar la solución final con:             <ul style="list-style-type: none"> <li>○ las primitivas</li> <li>○ los procedimientos</li> <li>○ las repeticiones</li> </ul> </li> </ul> <p>-&gt;<b>Programar</b></p>	
<p><b>Verificar:</b> ejecutar y comprobar que se haya logrado el objetivo (es necesario conocer de antemano el resultado esperado)</p>	<p>Se encendieron todas las luces</p>

Tabla 2: Aplicación del esquema de solución a una actividad con PilasBloques

Finalizada esta etapa, denominada “Introducción a la programación” en la planificación de las actividades de la asignatura, se iniciaron las actividades formales del curso. Para la resolución de las actividades prácticas se utilizó el mismo método aplicado en la resolución de las actividades lúdicas. Para la introducción de cada uno de los conceptos básicos, propios de un curso de programación inicial (modularización, estructuras de control, uso de parámetros), se referenció previamente la actividad lúdica que incluye el concepto. En la Figura 4 se muestra un ejemplo de la vinculación entre la programación por bloques y la programación imperativa.

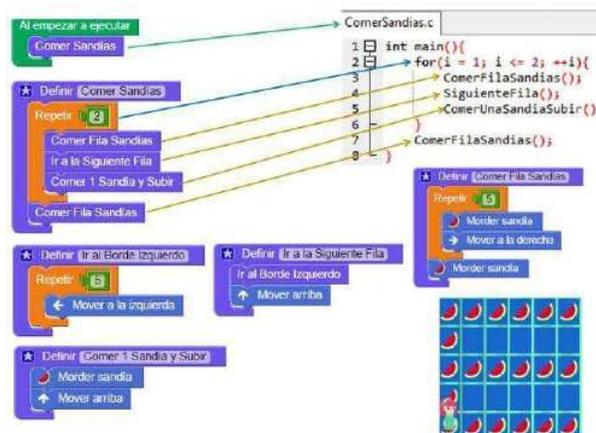


Figura 4. Vinculación entre la programación por bloques y la programación imperativa.

Esta etapa finalizó con una actividad de evaluación obligatoria en la que los alumnos debían resolver un desafío siguiendo el método propuesto. Como resultado, en la Tabla 3 se puede apreciar que el 80% aprobó la actividad (95% de los que rindieron).

Aprobados	799	80%
Desaprobados	43	4%
No rindieron	163	16%
	1005	100%

Tabla 3. Resultados actividad de evaluación

Si bien la evaluación consistía en una actividad muy sencilla, la aprobación de la misma era obligatoria para cumplir las condiciones de acreditación, para dejar en claro la importancia que la cátedra le confiere a los conceptos que consolidan el pensamiento computacional.

## 2. Programación por pares para el desarrollo de las actividades prácticas

La masividad de alumnos del ciclo lectivo 2023 exigió al equipo docente reorganizar el uso de los recursos, tanto materiales (aulas, equipamiento) como humanos (docentes), y optar por estrategias pedagógicas de autocontención y permanencia de los alumnos en las aulas.

En particular, se eligió la modalidad de programación por pares (*pair programming*) dado que es una metodología de desarrollo ágil que se utiliza en el mundo del trabajo.

Esta nueva mirada es una forma de enriquecer la formación de los futuros profesionales informáticos y permite acercar de manera temprana al aula procedimientos que se utilizan con éxito en la industria del software [6].

La programación por pares es un enfoque ágil en el que dos personas trabajan juntas de manera sistemática para desarrollar aplicaciones informáticas en tiempo reducido. Ha sido utilizada tanto en la industria del software como en la enseñanza de la programación y ha demostrado su efectividad en ambos escenarios [7].

Cada pareja debe adoptar e intercambiar los roles de conductor y navegador periódicamente. El conductor es la persona que utiliza el teclado en un momento determinado para escribir el código del programa. El navegador es la persona que retroalimenta al conductor a través de indicaciones, sugerencias y correcciones [8].

Para la implementación de la estrategia se solicitó a los alumnos de las 7 comisiones de aproximadamente 150 alumnos cada una, que gestionen e informen sus grupos de pares a través de un formulario digital.

Las clases de modalidad teórico-práctica conllevan una serie de ejercicios que permiten

comprender, reforzar y aplicar los contenidos conceptuales, y la correspondiente elaboración del código en lenguaje C que proponga una posible solución a los ejercicios planteados. Luego las soluciones de algunos pares de alumnos son expuestas para todos los asistentes, propiciando el debate y otras posibilidades o propuestas alternativas de soluciones. Si bien la metodología sostiene los roles de conductor y navegador, en esta propuesta no se puso el foco en estos roles, sino más bien en el trabajo colaborativo entre pares. El éxito de la metodología se basa en potenciar las relaciones interpersonales, promoviendo el trabajo en equipo, el aprendizaje de los desarrolladores, y un ambiente de trabajo cordial.

De este modo, se genera un espacio interactivo, de debate, que invita a los alumnos a pensar y analizar otras propuestas de solución, lo cual demanda conjugar esfuerzos, conceptos y competencias, mediante una serie de transacciones que les permita elaborar las soluciones de manera consensuada. Entender el trabajo colaborativo como una forma de trabajo que respeta las contribuciones individuales de los miembros del grupo [9].

Una comparación de la programación individual y en pareja reveló que la programación en pareja es eficaz y se recomienda utilizarla con más frecuencia en entornos educativos [10].

### **3. Uso de laboratorios móviles**

Tradicionalmente, la asignatura utilizaba un esquema de clases teóricas, clases prácticas y clases de laboratorio. La dinámica consistía en brindar los conceptos teóricos en las clases de teoría, consolidar estos conceptos mediante la resolución de guías de trabajos prácticos en las clases prácticas en aulas comunes, y realizar la codificación de las soluciones en los laboratorios de informática. Ante la imposibilidad de mantener esta estructura por la gran cantidad de alumnos e insuficiencia de recursos como aulas, laboratorios (y docentes), las clases se impartieron en la modalidad teórico-práctica, esto es, en aulas comunes se impartían los conceptos y se resolvían las actividades prácticas, incluyendo la codificación de las soluciones. Para esto los estudiantes concurrían con sus notebooks y otros utilizaron sus teléfonos móviles.

### **4. Modalidad de evaluación**

En consonancia con este nuevo esquema de trabajo, y con el propósito también de optimizar el tiempo de corrección de los numerosos exámenes y favorecer la devolución de las correcciones a los estudiantes, importante instancia de aprendizaje, en la planificación se consideró que el primer parcial sería por pares y el segundo parcial individual, entendiendo que mantener la modalidad por pares en todas las instancias de evaluación requeriría un mayor seguimiento de la actuación interna de los pares, imposible de realizar con una relación docente alumno de 1/75.

El primer parcial se realizó en la modalidad por pares en aulas comunes donde cada par contaba

con una notebook. Los alumnos estaban en un contexto “en línea”, con wifi a disposición para el acceso al aula, y a cualquier otro recurso, generando el riesgo de intervenciones externas en la resolución de las consignas. Con el agravante de que el examen se tomaba en el mismo momento a toda la comisión. Cada comisión contaba con más de 150 estudiantes.

Al momento de la corrección de estos exámenes, se encontraron códigos generados con algún software de AI generativa. Consultados los autores, admitieron el uso de ChatGPT. Esta situación era fácilmente detectable debido a que la solución que plantean no se regía por el método de resolución de problemas trabajado en las clases.

Ante la aparición de esta situación no prevista, con las dificultades de la atención a gran cantidad de alumnos, se decidió que los exámenes se tomaran en papel.

### Resultados obtenidos

a) Considerando el desempeño

Al finalizar el cursado se evaluó el desempeño de los alumnos según haya trabajado de a pares o no. En la Tabla 4 y la Tabla 5 se muestran los resultados.

Se aclara que la condición final de los estudiantes, de acuerdo con el régimen de acreditación de la asignatura, puede ser alguna de las siguientes categorías:

- Regular: 75% de asistencia a las clases y aprobar los 2 parciales obligatorios y aprobar actividad de PilasBloques. Este alumno continúa el cursado de las siguientes materias del plan, pero debe rendir un examen final teórico en los turnos de exámenes del año académico.
- Promociona: Si cumple con las condiciones anteriores y si el promedio de los 2 parciales obligatorios es 7 o superior está habilitado a un Tercer Parcial sobre los temas conceptuales de la asignatura. Aprobando este tercer parcial el alumno aprobó la materia.
- Libre: Son aquellos que no aprobaron las parciales o abandonaron el cursado.(\*). se cuenta el que aprobó

Condición Final	Cantidad	
Ambos regulares	46	
Ambos promocionados	104	
Alguno de los dos quedó regular y el otro promocionó	108	
Alguno de los dos aprobó y el otro quedó libre	254	127 (*)
Ambos libres	208	
TOTAL	720	

Aprobados (Regulares o Promocionados)	385	53%
Libres	335	47%
<b>TOTAL</b>	<b>720</b>	<b>100,00%</b>

(\*) se cuenta el que aprobó

Tabla 4: Desempeño de los estudiantes que trabajaron de a pares

<b>Condición Final</b>	<b>Cantidad</b>	
Promocionados	31	
Regulares	29	
Libres	224	
<b>TOTAL</b>	<b>284</b>	
Aprobados o Promocionados	60	21%
Libres	224	79%
<b>TOTAL</b>	<b>284</b>	<b>100,00%</b>

Tabla 5: Desempeño de los estudiantes que no trabajaron de a pares

En los resultados se puede apreciar que los estudiantes que trabajaron de a pares tuvieron un mejor desempeño.

b) Considerando opiniones recabadas mediante encuesta

Al finalizar el cursado se realizó una encuesta a los estudiantes, respecto de la modalidad del trabajo por pares. Respondieron 165 estudiantes. Consultados sobre su opinión respecto a la modalidad de trabajo por pares, un 83% indicó que le pareció buena o muy buena (83%).

Respecto de los aspectos positivos de la modalidad (Ver tabla 6), la mayoría destaca como positivo el debatir ideas y conceptos. Esto es particularmente interesante en la enseñanza de la programación dado que no existe una solución única para los problemas que se plantean en las guías de trabajos prácticos. Este debate entre pares contribuye significativamente en el aprendizaje.

Otro aspecto importante en la universidad, principalmente en el primer año, es tener un compañero de estudio. La modalidad por pares permitió que este acercamiento a otro, dado por condiciones académicas, facilitara el contacto para lograr un compañero de estudio.

Con esta modalidad se pretendía también facilitar el acceso al equipamiento necesario para la realización de los trabajos prácticos. Esta modalidad permitió que quienes no tenían una notebook pudieran realizar las actividades prácticas con los pares que sí contaban con el equipo.

Otro aspecto señalado fue la creación de hábitos de estudios, aspecto importante para el avance en la carrera. En alumnos ingresantes es bastante común que carezcan de metodologías de

estudio que les permita organizar mejor su tiempo. Esta modalidad puede contribuir al desarrollo de hábitos de estudio.

En menor medida, los alumnos consideran que la modalidad por pares les permitió aclarar dudas sobre los prácticos, llevar la materia al día y adaptarse a la vida universitaria.

Aspectos positivos	Cant	%
Debatir ideas y conceptos	66	40%
Tener un compañero de estudio	27	16%
Como no tengo notebook, pude trabajar con la de mi par	18	11%
Me obligó a crear un hábito de estudio junto a mi par	13	8%
Aclarar dudas sobre los prácticos	10	6%
Llevar la materia al día	10	6%
Me facilitó adaptarme a la vida universitaria	10	6%
Otras	11	7%
	165	100%

Tabla 6: Aspectos positivos de la modalidad por pares

Consultados acerca de las dificultades o inconvenientes que tuvieron en la modalidad por pares, la mayoría (61%) destaca como dificultades cuestiones de organización, como días, horarios y lugares para estudiar.

Por otra parte, un 12% indica como inconvenientes la falta de acuerdo para la resolución de los problemas y la inflexibilidad de alguna de las partes. Sobre estos aspectos, propios del trabajo en equipo, debería realizarse una capacitación previa hacia los estudiantes a fin de facilitar el cumplimiento de los objetivos de la modalidad.

Inconvenientes o dificultades	Cant	%
Adaptarme a los días y horarios de mi par	61	37%
Tenemos problemas con los horarios y lugar donde estudiar	31	19%
Pierdo tiempo en juntarnos para estudiar y hacer las tareas	9	5%
Discutimos mucho el ejercicio y no nos podemos poner de acuerdo	14	8%
Mi par siempre quiere hacer según su idea o estrategia	7	4%
Ningún inconveniente	28	17%
Otras cuestiones	15	9%
	165	100%

Tabla 7: Dificultades de la modalidad por pares

c) Opiniones respecto a la modalidad de evaluación

Se consultó a los estudiantes su opinión acerca de volver a la actividad individual en el segundo

parcial. En la tabla 8 se indican los resultados, que indican que en general, estuvieron de acuerdo con el retorno al examen individual.

Debimos seguir trabajando de "a pares"	29	18%
El cambio de modalidad me afectó negativamente	11	7%
El cambio de modalidad me afectó positivamente	26	16%
Me pareció apropiado para poder individualizar mis problemas	86	52%
Otros	13	8%
	165	100%

Tabla 8: Volver al examen individual

En la encuesta se les consultó a los estudiantes acerca de qué soporte consideraban más apropiado para el examen, si la notebook o el papel. El 84% consideró más apropiado usar una notebook.

Los que eligieron el papel, argumentaron lo siguiente:

*Me sentí más seguro de lo que hacía y no estaba con ese "miedo" al compilar. Me ayudó a pensar mejor el objetivo y la estrategia antes de codificar.*

*No tengo notebook, y ya me acostumbré al papel.*

*Te ayuda y obliga a aprender mejor las estructuras.*

*En papel, aunque dificultad más lograr un código que funcione, ayuda mejor a poder aprender las estructuras y funcionamiento.*

*El tener que rendir en papel me obligo a comprender cosas que antes me costaban y me favoreció a que pueda resolver yo sola y con lo que sabía el programa que me pedía el parcial, siento que de lo contrario seguiría sin comprender esas pequeñas cosas que luego en otras materias me haría falta y sobre todo en un futuro para trabajar porque la lógica y entender la estructura para armar un programa lo es todo.*

*Porque es la única forma de poder realmente adquirir conocimiento sólido, en netbook me confié con las estructuras ya dadas por el C. Me fue contraproducente.*

*Aunque prefiera más en notebook siento que es más apropiada en papel ya que no todos cuentan con una netbook y en papel es más justo para todos*

*Con netbook solo nos preocupamos de que funcione, mientras que en papel, siendo más difícil, me aseguré de pensar más en el funcionamiento de las partes formando un mapa mental.*

*Me resulto más práctico realizarlo en papel, ya que me exigió más práctica y estudio, y eso favorece a la hora de programar, así como también me permitió tener más presente el método de resolución de problemas y visualizar el escenario.*

*Es una buena estrategia ya que nos hace pensar, y como no sabemos si está bien, como para verificar, para mí este es el mejor método, para estar alerta.*

Los que eligieron la notebook, argumentaron por las ventajas que otorga, mayor rapidez para escribir el código, la posibilidad de verificar el funcionamiento, la facilidad para corregir el programa, reproducir el ambiente laboral, entre otros.

### **Conclusiones**

La enseñanza de la programación en contextos de masividad exige implementar nuevas estrategias que permitan cumplir con los resultados de aprendizaje esperados. En esta asignatura la suma de estrategias implementadas ha permitido, con las actividades lúdicas iniciales, consolidar un método de resolución de problemas que incorpora las herramientas conceptuales y las herramientas del lenguaje necesarias para promover el pensamiento computacional que se requiere para la programación, y hacer que el método se convierta en el eje que sustenta la programación convencional en el paradigma procedural. La programación por pares permitió que los estudiantes confronten sus ideas, compartan dudas y consoliden soluciones de a dos, optimizando el tiempo de corrección de exámenes y devolución, por parte de los docentes. Permitiendo de esta manera implementar en el aula metodologías que se utilizan con éxito en la industria del software.

Estimula y prolonga la permanencia de los estudiantes en el contexto universitario, según sus propias manifestaciones, en el sentido que les facilitó la adaptación.

Los laboratorios móviles permiten que los alumnos puedan verificar inmediatamente sus soluciones, las compartan en clase y realicen las modificaciones en el mismo momento, reproduciendo la actividad que se realiza en el mundo del trabajo y visualizando otras estrategias de sus compañeros pudiendo analizar y optimizar las soluciones.

Continuar con esta modalidad de trabajo y evaluación queda sujeta a un debate y análisis hacia el interior de la cátedra, en función de los resultados obtenidos. Si bien los resultados, en términos de desempeño, se mantuvieron en los indicadores de años anteriores, queda por resolver cómo sumar la IA a la metodología de enseñanza. Asimismo, los beneficios para los alumnos que realizan honestamente las actividades evaluativas y las respectivas prácticas de a pares encuentran en esta modalidad ventajas que favorecen y afianzan sus aprendizajes.

**Referencias bibliográficas**

- [1] Martínez López, P. E. (2013). *Las bases conceptuales de la Programación. Una nueva forma de aprender a programar*. Universidad Nacional de Quilmes. <http://www.gobstones.org/bibliografia/Libros/BasesConceptualesProg.pdf>
- [2] Factorovich, P. M. y Sawady O'Connor, F.A. (2017). *Actividades para aprender a Program.AR*. Fundación Sadosky. <http://program.ar/descargas/manual-docente-descarga-web-v2017.pdf>
- [3] Dapozo, G., Greiner, C., Petris, R., Godoy Guglielmone, M.V. (2019). *Enseñanza de programación en la universidad. Estrategia basada en programación por bloques*. En S. Pech, M. Prieto, J. García, & E. Orozco, "Innovation and Practice in Education" Págs. 89-98. Publisher: CIATA.org | UCLM | HIU.
- [4] Adnan, A., Romli, R. (2022). A Comparative Evaluation on Methods of Teaching Computer Programming. In: Saeed, F., Mohammed, F., Ghaleb, F. (eds) *Advances on Intelligent Informatics and Computing. IRICT 2021. Lecture Notes on Data Engineering and Communications Technologies*, vol 127. Springer, Cham. [https://doi.org/10.1007/978-3-030-98741-1\\_47](https://doi.org/10.1007/978-3-030-98741-1_47)
- [5] Kanika, K., Chakraverty, S., Chakraborty, P. (2020). Tools and Techniques for Teaching Computer Programming: A Review. *Journal of Educational Technology Systems*. DOI: 10.1177/0047239520926971
- [6] Zanga, A. M., Aubin, V. I., Dejean, G., Blautzik, G., Blautzik, L. J., Nisi, Z. J., Salica, M. E., Pafundi, F. R. (2015). *Mejoras en el proceso de enseñanza-aprendizaje de programación utilizando metodologías propias de la industria del software como caso particular de las metodologías activas*. Universidad Nacional de La Matanza. <http://repositoriocy.t.unlam.edu.ar/handle/123456789/400>
- [7] Roque Hernández, R. V., González Morales, R., Muñoz Castellanos, S. P. (2022). *La programación por pares: un análisis de la producción científica en Web of Science*. *Forum International Journal of Social Sciences and Humanities*, 4(7), e22474. [doi.org/10.35766/j.forhum.22474](https://doi.org/10.35766/j.forhum.22474)

- [8] Dalton, J. (2019). *Great Big Agile*. Apress Berkeley, CA. doi.org/10.1007/978-1-4842-4206-3
- [9] Maldonado, M. (2007). *El trabajo colaborativo en el aula universitaria*. Laurus, vol. 13, no. 23, pp. 263–278. Universidad Pedagógica Experimental Libertador Caracas, Venezuela.
- [10] Demir, Ö., Seferoglu, S. (2021). *A Comparison of Solo and Pair Programming in Terms of Flow Experience, Coding Quality, and Coding Achievement*. Journal of Educational Computing Research, 58(8), 1448-1466. doi.org/10.1177/0735633120949788

## Pasayo

### Una plataforma extendida para enseñar a programar a personas en el espectro Autista

Jorge Rodríguez, Federico Amigone, Guillermo Ariel Guerrero  
j.rodrig, fede.amigone,gillermo.guerrero@fi.uncoma.edu.ar

Facultad de Informática  
Universidad Nacional del Comahue

#### Resumen

La rápida inserción de las Ciencias de la Computación en los ámbitos sociales, políticos, productivos y económicos plantea nuevos desafíos educativos para personas con autismo.

Comprender sobre prácticas y conceptos fundamentales de la computación es una pieza clave para el ejercicio de la ciudadanía. Desarrollar capacidades de comprensión sobre las tecnologías y las Ciencias de la Computación mejora las posibilidades de las personas en el espectro para participar de las actividades sociales. Por otro lado, el conocer sobre programación se ubica como una plataforma consistente sobre la que pueden construirse proyectos de vida profesional en el campo de las tecnologías, favoreciendo el acceso al trabajo para las personas jóvenes en el espectro autista.

En este trabajo se presenta PASAYO, un marco de trabajo destinado a apoyar los procesos de enseñanzas y aprendizajes de conceptos sobre algoritmos y programación para personas con autismo. La iniciativa PASAYO instrumenta un enfoque didáctico que despliega múltiples modos de programación: tangible, basados en bloques y basados en texto, que son recorridos a partir de trasposiciones de narrativas y organizados en trayectos PASAYO; una plataforma digital para la enseñanza y un esquema de enseñanzas y aprendizajes que integra a la familia en el escenario de facilitación. Se describen los resultados obtenidos del trabajo de campo desarrollado con 17 familias.

**Palabras clave:** Enseñanza de las Ciencias de la Computación, Enseñanza de la Programación, Autismo.

#### Introducción

La irrupción de la tecnología en la constitución de los procesos sociales, políticos, productivos y económicos plantea profundos desafíos y oportunidades para las juventudes en el espectro autista. Por un lado, comprender cómo funcionan las computadoras y el software ya no es opcional, es requisito para gozar de una ciudadanía plena en el siglo XXI.

Sin una mínima capacidad de comprensión sobre las tecnologías y las Ciencias de la Computación, las personas en el espectro son relegadas a los márgenes de la actividad social, quedando como espectadores de un suceder que escapa a su comprensión e intervención. Se obtura así su derecho a la participación de los debates públicos; son situados forzosamente a una condición pasiva en la cual no se encuentran habilitados a opinar, quedando así relegados a las opiniones de otros. Por otro lado y más allá de la comprensión ciudadana del mundo, el conocimiento en programación es una plataforma sobre la que pueden construirse, impulsarse y defenderse derechos de acceso al trabajo para las y los jóvenes en el espectro autista.

Existen experiencias aisladas en otros países, tanto en el ámbito de la formación en conceptos relacionados a la programación como iniciativas que buscan ampliar la participación de las personas en el campo laboral tecnológico [6][8]. Estos estudios e iniciativas sugieren que existe afinidad entre el autismo y la programación y que esto resulta satisfactorio al menos en tres dimensiones: mejora las posibilidades de inserción laboral, contribuye a hacer efectivo el derecho a conocer que le asiste a todas las personas y mejora los procesos de socialización y empoderamiento [9][10].

No obstante, actualmente, el acceso al conocimiento tecnológico se encuentra fuera del alcance de las infancias, adolescencias y juventudes en el espectro autista [7]. Las instituciones educativas, de todos los niveles, que ofrecen formación en este campo no cuentan con dispositivos y enfoques didácticos específicos para las personas en el espectro. Y por otro lado, los diseños curriculares para estos trayectos formativos suelen estar sujetos a rigideces normativas y conservadoras que si bien pueden funcionar para personas situadas en la media funcional, habitualmente no se acomodan a jóvenes en el espectro autista [1][2].

Desde la perspectiva epistemológica, la programación y el autismo configuran un espacio de amplia vacancia. No existen suficientes iniciativas de investigación que hayan logrado consolidar conclusiones teóricas y prácticas basadas en evidencia, ni tampoco la demarcación de los contornos para la definición de una didáctica específica posible [7]. Las iniciativas en investigación que tengan por objeto de estudio y desarrollo a los dispositivos, mecanismos y didácticas específicas para producir aprendizaje en programación son, por tanto, tan necesarias como escasas. Los dispositivos no solo deben diseñarse, sino que deben construirse y experimentarse en el campo de estudio. La construcción de plataformas didácticas, costosa en recursos económicos, humanos y temporales, requiere de aproximaciones de especialistas y del contacto efectivo con la población autista interesada.

Este trabajo se enmarca dentro de la Línea de Investigación y Desarrollo destinada a producir recursos didácticos para enseñar Ciencias de la Computación y evaluar su efectividad en el ámbito de la educación.

El resto del artículo está organizado de la siguiente manera. A continuación, describimos la línea de investigación y desarrollo. En la Sección 3, presentamos los resultados obtenidos y esperados. Finalmente, en la Sección 4, comentamos aspectos relacionados a la formación de recursos humanos en el marco de esta línea, así como del proyecto de investigación en su conjunto.

### **Trabajos relacionados**

Existen experiencias aisladas en otros países, tanto en el ámbito de la formación en conceptos relacionados a la programación como iniciativas que buscan ampliar la participación de las personas en el campo laboral tecnológico.

En general, cuando se examina la literatura existente en relación al autismo y programación, incluyendo estudios que investigan habilidades cognitivas y de procesamiento en individuos en el espectro autista, los autores concluyen que hay una correlación positiva entre las habilidades de programación y los rasgos del espectro autista, y sugieren que la programación podría ser una herramienta útil para mejorar la empleabilidad y la calidad de vida de personas con autismo. En [12] se investiga la conexión entre el autismo y las habilidades de programación, sugiriendo que los individuos con TEA pueden tener una aptitud particular para la programación.

En [13], es posible encontrar un trabajo que examina la accesibilidad de la programación informática para individuos neurodiversos, incluidos aquellos con autismo, y se sugieren estrategias para hacer que la educación en programación sea más inclusiva. Los autores también señalan que los individuos en el espectro autista pueden enfrentar desafíos únicos en el entorno de la programación, como la necesidad de interacción social y la flexibilidad para trabajar en equipo. En general, las observaciones epistémicas en relación a la producción computacional de la población autista sugiere que se necesitan enfoques de enseñanza específicos que aborden estas áreas de necesidad más allá de que exista potencial productivo en el ámbito del autismo y la programación, y por ello esta área de investigación merece mayor atención y exploración.

Así mismo, existen estudios que examinan las mejoras que se producen al enseñar programación a niños con autismo en lo relativo a sus habilidades sociales y de comunicación. La programación puede ser una herramienta efectiva para mejorar estas habilidades, debido a su naturaleza estructurada y predecible, y al hecho de que puede involucrar tareas repetitivas y detalladas que pueden ser de interés para los niños con autismo. En [15] se realiza una revisión sistemática de la literatura que explora la conexión entre el TEA y las habilidades de codificación y proporciona una visión general de la investigación existente en esta área.

Algunos autores también señalan que enseñar programación a niños con autismo puede ser un desafío debido a su dificultad para interactuar socialmente y trabajar en equipo. Sin embargo, la enseñanza de programación puede ser adaptada para abordar estas dificultades y proporcionar una oportunidad para mejorar las habilidades sociales y de comunicación. En [14] se analiza cómo las organizaciones de desarrollo de software pueden apoyar la neurodiversidad, incluidas las personas con autismo, y destaca los beneficios potenciales que pueden aportar al campo. En [16] se revisa la literatura relacionada con el autismo y la tecnología de la información (TI), incluida la programación informática, para comprender los desafíos y las oportunidades para las personas con TEA en los campos relacionados con la TI. En general, los estudios sugieren que la programación puede ser una herramienta efectiva para mejorar las habilidades sociales y de comunicación en niños con autismo, y que es importante considerar el enfoque de enseñanza adecuado para asegurar que los niños obtengan los mayores beneficios posibles.

### **Enfoque PASAYO**

El enfoque PASAYO es un abordaje epistémico que persigue ensayar las condiciones, recursos, prácticas y metodologías que generen las condiciones necesarias para que se produzca aprendizaje en el área de la programación para la población autista interesada. Dichas condiciones de posibilidad requieren ser construidas colectivamente, habida cuenta de que las características cognitivas y conductuales de la población autista conforman un espectro amplio.

El desafío que PASAYO aborda, es el de generar las condiciones que permitan el aprendizaje de los conceptos de instrucción y su composición en programas algorítmicos; los tipos de datos, variables, constantes; las estructuras que permiten controlar los flujos de ejecución tales como las repetitivas y las alternativas y finalmente, las estructuras de datos en memoria como los arreglos y las colecciones. Todos estos recursos cognitivos son utilizados para la producción de artefactos computacionales que resuelvan determinados desafíos con significación individual y situada para cada estudiante.

En este sentido, el enfoque PASAYO parte de seis premisas que intentan configurar una superficie sobre la que ensayar prácticas basadas en evidencia, a saber:

- **Acceso Temprano:** PASAYO busca propiciar oportunidades de incorporación temprana a las prácticas de exploración y aprendizaje de la programación como objeto de estudio. El promedio de edad de la población estudio del presente trabajo es de 11 años.
- **Dominio Tecnológico:** Aunque algunos estudios relacionados con programación y autismo señalan sus ventajas a la hora de producir habilidades sociales, el objeto de estudio

sobre el que el enfoque PASAYO se centra en la producción de saberes y habilidades sobre la tecnología computacional en general y la programación en particular.

- **Rol Productivo:** Los y las estudiantes que abordan el enfoque PASAYO, no lo hacen en calidad de consumidores de tecnología. Sus trayectos formativos no tienen por objeto el aprendizaje de tecnologías que son ofrecidas como servicios o productos, sino la producción de artefactos computacionales ejecutables desde el mismo inicio de su recorrido.
- **Rigurosidad Disciplinar:** Si bien el proceso de abordaje al objeto disciplinar regula el nivel de abstracción con el que se lo presenta, el contenido es impartido en toda su rigurosidad disciplinar. Las gramáticas textuales de los lenguajes utilizados en los modos Bloques y Texto, no sufren alteraciones ni adaptaciones, lo cual permite que las producciones resultantes puedan leerse por cualquier programador que domine el lenguaje.
- **Competencia Profesional:** El proceso formativo tiene por objetivo la formación de capacidades altamente valoradas en el mercado tecnológico.
- **Entorno Ciudadano:** Los dispositivos que soportan los trayectos formativos no se limitan a los contextos educativos normativos como la escuela o la universidad, sino que exploran y explotan el potencial que ofrecen las comunidades, la ciencia ciudadana y fundamentalmente los apoyos familiares.
- **Trayecto Situado:** Todo el proceso que atraviesa el estudiante es didácticamente intervenido a efectos de adaptarlo estratégicamente dentro de los límites de intereses propios y fortalezas individuales del estudiante.

### Enfoque Didáctico

En ésta sección, describiremos los elementos constitutivos del enfoque PASAYO, a saber, la narrativa traspuesta y el trayecto.

El concepto basal de la didáctica del enfoque de PASAYO es la transposición de narrativas PASAYO. Se trabaja con el estudiante una narrativa (coelaborada entre el facilitador y el estudiante), específicamente adaptada a su agenda de intereses personales y a su capacidad cognitiva. Ésta narrativa, se representa con un párrafo corto y con soportes visuales en caso de ser necesario, exponiendo alguna situación concreta que pueda ser representada o solucionada algorítmicamente. De esta forma, el rol del estudiante, será transponer la narrativa a su equivalente algorítmico computacional.

Una narrativa es una enunciación verbal o escrita muy corta y que representa un desafío en el formato de cuento. Las narrativas PASAYO poseen un dominio específico, el cual se espera que

esté sujeto a la agenda de intereses singulares del estudiante y una complejidad determinada. Cada narrativa es traspuesta, esto es, partiendo del relato, se construye un artefacto computacional ejecutable que lo resuelva.



El concepto que vehiculiza el tránsito a través de PASAYO es el de trayecto. Un trayecto es un conjunto, siempre mayor o igual a 3, de narrativas PASAYO. En la figura 01 puede observarse el diseño general de un trayecto PASAYO. Todas las narrativas de un trayecto deben coincidir en su graduación de complejidad y su dominio. El enfoque busca generar indicadores relativamente objetivos respecto de la experiencia de aprendizaje con el estudiante.

El grosor de las flechas en cada narrativa representa el indicador cuantitativo del nivel de regulación que el facilitador del trayecto va seteando en cada narrativa. Así, es de esperar que ese nivel de regulación comience a descender una vez que se avanza en el trayecto PASAYO. Si el indicador de regulación no tiende a cero durante las iteraciones de trasposición de una narrativa, es probable que el seteo de complejidad, dominio o apoyos sea incorrecto y se requiera reingeniería didáctica sobre el trayecto.

Solo cuando se logró que el indicador de regulación converga cero, esto es, el estudiante logró trasponer la narrativa sin la intervención de su facilitador/a, se avanza a la siguiente narrativa dentro del trayecto.

Por otro lado, se espera que los trayectos utilizados soporten una complejidad incremental, de manera que sobre un mismo dominio argumental es posible transitar un recorrido que incorpore complejidad en forma regulada y sin pérdida de contexto semántico.

Otro aspecto importante del enfoque didáctico de PASAYO es que el proceso formativo está guiado por una curricularización dinámica. No existen tiempos prefigurados para ningún concepto disciplinar. Es decir, no existen preconceptos respecto del tiempo que debería tomarle a un estudiante transitar determinados contenidos disciplinares. Esta desestructuración curricular propicia un flujo de aprendizaje adaptativo a las capacidades y en los tiempos de elaboración de fortalezas cognitivas de cada estudiante.

Finalmente, PASAYO hace del potencial de la producción colaborativa un componente fundacional de su estratégica didáctica. Las narrativas que se producen dentro de la plataforma retroalimentan la comunidad de usuarios y permiten potenciar las posibilidades de producción y aprendizaje.

### **Plataforma digital**

El proceso formativo del enfoque PASAYO está soportado por su propia plataforma didáctica específica. La experiencia de los estudiantes está pensada como un pasaje de complejidad incremental entre tres modos de programación: tangible, bloques y texto.

- **El modo Tangible**

Una de las formas más amigables para introducir las primeras nociones sobre algoritmia, en niños desde los 3 años de edad, es la Programación por medio de *Interfaces de Usuario Tangibles* o simplemente *Programación Tangible*. Su éxito radica en que los niños en general aprenden de una manera activa: a través de experiencias con otras personas y utilizando objetos manipulables que les permiten interactuar con su mundo. Estas particularidades también son valoradas en la población de niñas y niños en el espectro autista, población que suele responder mejor a entornos visuales y concretos.

Con los entornos de Programación Tangibles los niños desde edades tempranas, tienen la oportunidad de manipular directamente objetos, ensamblando, encadenando y/o conectándose. Estos objetos tangibles, en realidad, representan instrucciones precisas y generan, al secuenciarlos, un programa. Así, los niños interactúan con objetos físicos y transforman la lógica

del mundo real en la lógica del programa. De este modo, se logra hacer que las manipulaciones simbólicas y abstractas involucradas en los procedimientos creativos, se vuelvan más concretas y manejables para los niños, a través de tecnologías sin teclados. Así, la programación tangible permite la utilización de piezas físicas como componentes de programación. Cada pieza tiene características y funcionalidades específicas que al secuenciarse se genera un programa.

Los LPT asumen que piezas físicas pueden constituirse en elementos sintácticos de un lenguaje (variables, instrucciones, operadores, entre otros). La colección de piezas físicas proporciona al programador, posiblemente un sujeto no alfabetizado, la posibilidad de secuenciar u ordenar elementos físicos para construir un algoritmo. Es posible pensar en LPT simple compuesto por un conjunto reducido de primitivas, por ejemplo que no involucra el uso de variables o de algunas estructuras de control, como un subconjunto de un LPT más complejo. Esta característica se convierte en andamiaje del proceso de aprendizaje, en el sentido que es posible sumar complejidad al lenguaje a medida que los nuevos conceptos sobre programación tienen posibilidades de integrarse a las estructuras cognitivas del sujeto.

Así, PASAYO implementa su modo de programación tangible con un entorno web desde el que es posible descargar una colección de micromundos y elementos del lenguaje para cada narrativa propuesta. Adicionalmente, es posible intervenir el micromundo modificando algunas cuestiones tales como la temática del micromundo o el avatar utilizado, a efectos de adaptar la narrativa a los intereses singulares del estudiante. Como constante didáctica PASAYO en todos sus modos, el entorno permite partir de una narrativa y trasponer su solución en un código ejecutable de instrucciones de tipo tangible.

- **El modo Bloques**

Los lenguajes de programación por bloques son una forma de programar empleando bloques gráficos o visuales para crear algoritmos ejecutables. En éstos entornos los usuarios pueden seleccionar bloques y arrastrarlos a una interfaz gráfica para construir su programa.

Los entornos de programación con bloques, son utilizados en entornos de programación educativa, donde se busca introducir conceptos de programación a estudiantes jóvenes o principiantes sin la necesidad de aprender una sintaxis compleja o una gramática de programación. Los bloques suelen representar comandos, acciones, operaciones matemáticas, entre otros, y están diseñados para ser fácilmente comprensibles e intuitivos.

El modo de programación por Bloques de PASAYO, se implementa en un entorno web que integra Google Blockly. Como constante didáctica PASAYO en todos sus modos, el entorno permite partir de una narrativa y trasponer su solución en un código ejecutable de instrucciones de tipo bloque.

- **El modo texto**

PASAYO utiliza un lenguaje textual de amplio uso en la industria: javascript. El entorno de programación textual incorpora múltiples características tales como:

**Recorridos:** son colecciones de narrativas que desarrollan un mismo aspecto curricular ordenadas por complejidad.

**Soporte para colaboración en tiempo real:** a través de un link privado, es posible compartir la sesión de programación con múltiples usuarios.

**Corrección:** las sesiones quedan almacenadas para su posterior corrección.

Es importante señalar que las narrativas de los trayectos PASAYO definidos en la sección anterior, deben ser compatibles en los modos adyacentes, es decir, una narrativa en el modo de programación tangible debe seguir siendo válida en el modo bloques, y cualquier narrativa del modo de programación con bloques debe ser equivalente en el modo de programación textual.

### **Esquema de facilitación**

PASAYO funciona con un modelo de facilitación experimental para el diseño de intervenciones didácticas. Esta facilitación se organiza bajo un esquema de ternas compuestas por las figuras del estudiante en el espectro autista, el facilitador familiar y el facilitador PASAYO. En general, el facilitador PASAYO no toma contacto directo con el estudiante, sino que este proceso es mediado por los facilitadores familiares. Sin embargo, en caso de que el estudiante así lo desee, puede establecerse un modelo directo entre el estudiante y el facilitador PASAYO.

Si bien el proceso comienza tomando contacto y registrando las características del estudiante, las cuales pueden ser aportadas por el estudiante, su familia o alguna figura terapéutica, el esquema de facilitación adquiere una forma circular, en tanto se intenta hacer intervenir en el proceso de aprendizaje todas las fortalezas y debilidades del estudiante. Comprende los siguientes momentos:

- **Recuperación de la experiencia anterior:** este momento busca registrar conclusiones sobre la experiencia anterior a efectos de lograr una mejor observación y registro de los resultados epistémicos obtenidos. Por otra parte, contar con la mayor información posible respecto de las características del estudiante es también un factor estratégico a la hora de retroalimentar el diseño de las intervenciones.
- **Presentación de la narrativa:** el material de trabajo es presentado al facilitador familiar, y en este punto se concreta el primer momento didáctico ya que no es posible asumir ninguna

capacidad técnica en la materia por parte de la familia o el terapeuta que ejerce el rol de facilitador familiar. En este momento, no se produce aprendizaje en el estudiante, sino en su entorno de facilitación.

- **Acomodaciones:** en este momento se realizan adaptaciones a la narrativa y al material de trabajo a efecto de acomodarlo a las características del estudiante. Estas acomodaciones pueden ser reemplazos temáticos, cambio de avatares, etc.
- **Diseño de la intervención:** en este momento, bajo la suposición de que el facilitador familiar ya tiene dominio técnico de la narrativa y que la misma fue adaptada a los intereses y capacidades del estudiante, se diseña el momento en que el estudiante tomará contacto con la misma. En este punto, el facilitador PASAYO propone la secuencia didáctica que desarrollará la narrativa y que buscará tener éxito en los registros emocionales tanto del estudiante como de la familia, independientemente de los resultados técnicos que se alcancen.
- **Coordinación del próximo encuentro:** dentro del ciclo de facilitación, se reserva el último momento para la planificación semanal del encuentro entre el estudiante y su facilitador familiar. Esto consolida el compromiso de sostener la frecuencia de trabajo necesaria para la producción de habilidades en programación.

Los resultados de cada experiencia son volcados en una base de datos epistémica sobre la cual es posible realizar conclusiones sobre la efectividad del modelo propuesto, evaluar el impacto en el proceso de aprendizaje y construir métricas de desempeño del proceso formativo.

### **Experiencia**

La experiencia del enfoque PASAYO tuvo una duración de 12 meses, en los cuales se procedió a realizar una convocatoria a familias interesadas a través de organizaciones civiles vinculadas a los derechos de las personas en el espectro autista.

El equipo de facilitación estuvo conformado por dos becarios, estudiantes avanzados de las carreras de Licenciatura en Ciencias de la Computación, dos docentes facilitadores ad honorem, dos investigadoras ciudadanas vinculadas a los derechos de la mujer y el equipo de coordinación compuesto de dos investigadores FAI-UNCo.

La extracción geográfica de las familias registró aproximadamente en un 80% de la provincia de Neuquén, y el resto de puntos geográficos varios como la provincias de Río Negro y Buenos Aires. Se priorizaron aquellos estudiantes con capacidades de lectoescritura e interés por la computación.

## **Desafíos**

El despliegue en campo del enfoque PASAYO en la población autista estudiada, sugiere que no existen grandes dificultades de orden cognitivo para la adquisición de habilidades en programación.

Las dificultades observadas parecen indicar que existen complejidades en el sostenimiento de la infraestructura necesaria para dar continuidad al enfoque así como en las dificultades que representa la incorporación de los espacios de facilitación en la organización cotidiana de la familia; organización que en general se encuentra ya cargada de actividades relativas a las terapias clínicas de los estudiantes.

## **Conclusiones**

En el contexto de las actividades desarrolladas en este trabajo, se logró desplegar los primeros experimentos de aplicación de enfoques y plataformas en la enseñanza de la programación a personas en el espectro autista. Estos experimentos de campo incorporan las modalidades de programación tangible, bloques y texto a lo largo de un trayecto con la participación de un niño en el espectro. Al momento, y luego de un año de trabajo con 17 familias, el enfoque propuesto verifica la producción de artefactos computacionales en distintos modos de programación, tangible, bloques y texto. Esta producción aporta indicios alentadores acerca de que el enfoque habilita la construcción de las estructuras cognitivas que permiten el dominio progresivo de la algoritmia en estudiantes dentro del espectro autista. En general, a la mayoría de las familias les tomó aproximadamente un mes con dos estímulos semanales, realizar las propuestas de programación tangible y esto las ubicó en un estadio cognitivo satisfactorio para comenzar a programar en el paradigma de bloques.

Los resultados sugieren que el modelo de trabajo propuesto resulta efectivo. En el plano didáctico específico, no se observaron dificultades significativas en el abordaje inicial a las complejidades algorítmicas bajo el modo tangible. Se verificó que la flexibilidad de la propuesta funcionó bien singularizando las actividades a la agenda propia de los y las estudiantes. Se produjo conocimiento computacional en niñas, niños y adolescentes que no tenían ningún conocimiento algoritmo previo.

Respecto del modelo de facilitación, las ternas demostraron ser una arquitectura posible y potente. En los casos donde la familia logró regularidad en los espacios de facilitación, se observaron progresos incrementales en las soluciones construidas por los y las estudiantes. El pasaje del paradigma tangible al paradigma bloques resultó ser apropiado, habilitando la producción de soluciones sin mayores complicaciones habida cuenta de que los conceptos basales de instrucción, algoritmo y lenguaje fueron bien modelados bajo el paradigma tangible.

Por otra parte, el enfoque de diseño participativo parece una respuesta satisfactoria al problema de producción de recursos situados. La plataforma PASAYO resulta ser un medio valioso para la producción de recursos y su socialización.

## Referencias

- [1] Gribble, J., Hansen, A., Harlow, D., y Franklin, D. (2017). Cracking the code: the impact of computer coding on the interactions of a child with autism. En Proceedings of the 2017 conference on interaction design and children (pp. 445–450).
- [2] Wille, S., Century, J., y Pike, M. (2017). Exploratory research to expand opportunities in computer science for students with learning differences. *Computing in Science & Engineering*, 19 (3), 40–50.
- [3] Sapounidis, T., Stamelos, I., y Demetriadis, S. (2016). Tangible user interfaces for programming and education: A new field for innovation and entrepreneurship. En *Innovation and entrepreneurship in education* (Vol. 2, pp.271–295). Emerald Group Publishing Limited.
- [4] Weintrop, D. (2019). Block-based programming in computer science education. *Communications of the ACM* , 62 (8), 22–25.
- [5] Weintrop, D., y Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education (TOCE)*, 18 (1), 1–25.
- [6] Krzeminska, A., Austin, R. D., Bruy`ere, S. M., y Hedley, D. (2019). The advantages and challenges of neurodiversity employment in organizations. *Journal of Management amp; Organization*, 25 (4), 453–463. doi: 10.1017/jmo.2019.58
- [7] Munoz, R., Villarroel, R., Barcelos, T. S., Riquelme, F., Quezada, A., y Bustos-Valenzuela, P. (2018). Developing computational thinking skills in adolescents with autism spectrum disorder through digital game programming. *IEEE Access*, 6 , 63880–63889.
- [8] Zubair, M. S., Brown, D. J., Hughes-Roberts, T., y Bates, M. (2021). Designing accessible visual programming tools for children with autism spectrum condition. *Universal Access in the Information Society*, 1–20.
- [9] Grandin, T. (1999). Choosing the right job for people with autism or asperger’s syndrome. Indiana Resources Center for Autism.
- [10] Eiselt, K., y Carter, P. (2018). Integrating social skills practice with computer programming for students on the autism spectrum. En 2018 IEEE Frontiers in Education Conference (FIE) (pp.

1–5).

[11] Vohland, K., Land-Zandstra, A., Ceccaroni, L., Lemmens, R., Perelló, J., Ponti, M., . .

. Wagenknecht, K. (2021). *The science of citizen science*. Springer Nature.

[12] Xavier Huysman, Lore Desmet, Tinne De Laet. "Exploring the Link between Autism Spectrum Disorder and Computer Programming Skills". Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 2016.

[13] Jamie Gower, Shah Rukh Humayoun, et al. "Programming for All: A Study of Computer Programming Accessibility for Neurodiverse Individuals". Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE), 2021

[14] H. A. Feiler, A. Oestreicher-Singer "Supporting Neurodiversity in Software Development" Published in Proceedings of the 2018 International Conference on Software Engineering (ICSE), 2018

[15] Khaled Badran, Mohammed Anbar, et al. "Understanding the Link between Autism Spectrum Disorder and Coding: A Systematic Literature Review" IEEE Access, 2020

[16] Antti Knutas, Pekka Kallioniemi "Autism and IT: A Literature Review", Proceedings of the 2019 ACM Conference on International Computing Education Research (ICER), 2019

**PRENDER:****Una propuesta didáctico-pedagógica para la enseñanza de las Ciencias de la Computación**

Christian Cossio-Mercado <sup>1,2</sup>, Gonzalo Pablo Fernández <sup>1,3</sup>  
{ccossio, gpfernandez}@dc.uba.ar

<sup>1</sup> Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires.

<sup>2</sup> Instituto de Ciencias de la Computación  
CONICET - Universidad de Buenos Aires

<sup>3</sup> Universidad Nacional de Quilmes.

**Resumen**

La enseñanza de programación y robótica se pusieron en primer plano en la Argentina con la aprobación, en 2018, de los Núcleos de Acceso Prioritarios (NAPs) por parte del Consejo Federal de Educación. Esto ocurrió luego de varios años de trabajo por parte de organizaciones públicas y privadas, como fundaciones y universidades, y hasta desde el ámbito de la representación gremial docente. Así, actualmente se busca que se trabaje sobre distintos contenidos de las Ciencias de la Computación, con su propio espacio curricular, y no sólo sobre temas de programación. A partir de la experiencia de los últimos siete años en el desarrollo de cursos y talleres de didáctica de la programación, se detectaron varias falencias en las bases teóricas-conceptuales de las prácticas de enseñanza de estos temas y, además, en el enfoque seguido en la formación docente de base, así como en cursos y talleres relacionados. Así, se vio necesaria la definición de un marco didáctico-pedagógico integral para la enseñanza de programación, pensamiento computacional y otros conceptos de las Ciencias de la Computación, el cual incluye definiciones para la planificación de actividades de clase, así como el seguimiento de las mismas, enfoques para el acercamiento al conocimiento, por medio de la indagación y el trabajo con problemas, la organización de cursos por medio de aprendizaje combinado, y un enfoque de la evaluación como una instancia más de enseñanza. Esta propuesta puede ser de utilidad para docentes que desean acercarse a estos temas como parte de su formación didáctica, como también para aquellas personas que cumplan funciones de coordinación y gestión, y deban organizar cursos y talleres, o, inclusive, planes de estudio en el marco de profesorado o formación docente complementaria. Adicionalmente, puede ser de utilidad para distintos niveles educativos, más allá de la enseñanza obligatoria, como puede ser la educación universitaria o para cursos o formación profesional.

**Palabras clave:** Enseñanza de la Programación

**Introducción**

En este trabajo se resumen las principales ideas y la definición de un marco conceptual y práctico para la enseñanza de contenidos de programación, aunque puede ser utilizado para otras áreas relacionadas a la resolución de problemas, denominada PRENDER (PRopuesta de ENseñanza DuradERa).

La enseñanza de programación y robótica se pusieron en primer plano en la Argentina con la aprobación, en 2018, de los Núcleos de Acceso Prioritarios (NAPs) por parte del Consejo Federal de Educación, luego de varios años de trabajo. Complementariamente, desde hace años se realizan distintos esfuerzos desde organizaciones público-privadas, como la Fundación Sadosky, universidades, y hasta desde el ámbito de la representación gremial docente, donde se busca que se trabaje sobre distintos contenidos de las Ciencias de la Computación, con su propio espacio, y no sólo sobre temas de programación, ya que estos últimos son sólo una parte de todos los contenidos que forman parte de una currícula integral en el área.

A partir de la experiencia de los últimos siete años en el desarrollo de cursos y talleres de didáctica de la programación, se detectaron varias falencias en el enfoque con el cual se desarrollaban las prácticas de enseñanza en las escuelas y, además, en el enfoque seguido en la formación docente de base, así como en cursos y talleres relacionados. De esta forma, se vio necesaria la definición de un marco didáctico-pedagógico para la enseñanza de programación, pensamiento computacional y otros conceptos de las Ciencias de la Computación, el cual incluye definiciones para la planificación de actividades de clase, así como el seguimiento de las mismas, enfoques para el acercamiento al conocimiento, por medio de la indagación y el trabajo con problemas, la organización de cursos por medio de aprendizaje combinado, y un enfoque de la evaluación como una instancia más de enseñanza.

Este trabajo está motivado por la necesidad de integrar herramientas, enfoques y técnicas conocidas en la formación docente y en el ámbito educativo, pero con un foco en la enseñanza de las Ciencias de la Computación, y no sólo vistos como contenidos que habitualmente forman parte de las materias de didáctica general en la formación de profesorado. Esta propuesta puede ser de utilidad para docentes que desean acercarse a estos temas como parte de su formación didáctica, como también para aquellas personas que cumplan funciones de coordinación y gestión, y deban organizar cursos y talleres, o, inclusive, planes de estudio en el marco de profesorado o formación docente complementaria. Adicionalmente, esta propuesta puede ser de utilidad para distintos niveles educativos, más allá de la enseñanza obligatoria, como puede ser la educación universitaria o para cursos o formación profesional.

El aporte esencial de esta propuesta es la de incluir en el mismo marco distintas propuestas relacionadas con la enseñanza de las ciencias y la resolución de problemas, siempre con el foco en la enseñanza de la programación y el pensamiento computacional.

En las siguientes secciones se resumen los principales componentes de este enfoque, de forma de destacar las partes que son relevantes a efectos de nuestra propuesta.

## **Bases conceptuales de la propuesta**

### **Enseñanza para la Comprensión**

Algo trascendental al momento de realizar actividades de enseñanza es definir cómo sabemos cuál es el grado de comprensión de los temas por parte de las y los estudiantes. Existen varias escalas para poder analizar esto, entre las que están la denominada taxonomía de Bloom original (Bloom et al, 1956) y su versión revisada (Anderson et al, 2001), en las cuales se define una jerarquía para el grado de aprendizaje que una persona puede tener sobre algún tema. Por otro lado, existen otros modelos, como el de la enseñanza para la comprensión (Stone Wiske et al, 1999), donde se define a la *comprensión* como la medida elegida. Así, a diferencia de la taxonomía de Bloom y otros modelos similares, no se opta por un modelo jerárquico y secuencial (con sus consiguientes problemas, como plantean Ritchhart y otros (2014), pp. 38), sino que se plantea que la comprensión se define como la capacidad de desempeñarse de manera flexible con el saber aprendido. Así, el aprendizaje no frágil (i.e., duradero y robusto) requiere comprensión.

Para trabajar con enseñanza para la comprensión, entre otras cosas, se requiere formularse y responder tres preguntas: 1) ¿Qué quiero que aprendan mis estudiantes?, 2)

¿Cómo sé que comprenden?, y 3) ¿Cómo saben mis estudiantes que comprenden?. De esta forma, para poder evaluar el grado de comprensión de un tema, ya sea que implique una acción externa (visible) o requiera sólo pensamiento (no visible), se deben plantear metas a partir de los desempeños que queremos que alcancen los y las estudiantes. Una vez definidas esas metas, es central que tanto docentes como estudiantes puedan evaluar el progreso con respecto a esas metas.

La idea de **desempeño flexible** es central para PRENDER, en tanto será la medida de la comprensión de un tema por parte de los y las estudiantes, y, además, servirá como evidencia de que las propuestas didáctico-pedagógicas de nuestros cursos sirven a sus fines. Así, buscaremos que los y las estudiantes no sólo puedan resolver tareas en el marco de las herramientas y consignas de un curso determinado, sino que será necesario demostrar que esos conocimientos pueden ponerse a prueba satisfactoriamente en otros contextos o con otras herramientas. Esto no sólo redundará en hacer más robustos los aprendizajes, sino que también ayuda a la autopercepción de lo aprendido, ya que pueden ver que lo que aprendieron no sólo sirve en un contexto cerrado y acotado como un curso, sino que también puede servir para resolver problemas y aumentar su capacidad para reflexionar sobre distintas cuestiones que se escapan

directamente de lo visto, pero que, como docentes, sabemos son derivaciones de lo trabajado.

### **Aprendizaje por indagación**

Como se resume en (Bell & Binns, 2005) y (Banchi & Bell, 2008), se pueden definir distintos niveles de indagación, dispuestos en un continuo<sup>1</sup> de acuerdo a qué tienen que hacer y qué autonomía tienen las y los estudiantes:

**Nivel 0 - Indagación Confirmativa:** Confirman un principio a través de una actividad, para la cual se conocen los resultados de antemano.

**Nivel 1 - Indagación Estructurada:** Investigan a partir de una pregunta definida por la o el docente, a través de un procedimiento definido por el mismo.

**Nivel 2 - Indagación Guiada:** Investigan a partir de una pregunta definida por la o el docente, usando procedimientos definidos por ellos mismos.

**Nivel 3 - Indagación Abierta:** Investigan a partir de una pregunta de propia elección, y siguiendo los procedimientos diseñados/seleccionados por ellos mismos.

### **Indagación estructurada**

La idea detrás de esta técnica es que el aprendizaje se dé a través de la ejecución de una tarea, la falla y la solución a la misma. Opuesto a la idea tradicional de que la autoridad transmite oralmente el conocimiento a sus aprendices, se prepara el terreno para que las y los estudiantes puedan adquirir ese conocimiento por su cuenta (acompañados y guiados, desde ya) al intentar aplicarlo, justificarlo o negarlo. Es algo así como “prueba y error”.

Las actividades diseñadas en torno a esta modalidad presentan un desafío que, en principio, las y los estudiantes no saben resolver. En lugar de explicarles cómo resolverlo y luego que realicen lo que acabamos de decirles que hagan, se les deja primero experimentar. Es posible (y en muchas ocasiones esperable) que logren alcanzar la solución deseada pero siempre se debe asumir que esto no va a pasar y por lo tanto se debe dedicar una cantidad no despreciable de tiempo a realizar una puesta en común y explicación de la solución esperada.

Hay que tener cuidado al planificar una actividad de esta forma, ya que se corre el riesgo de que las y los estudiantes se frustren y consecuentemente se nieguen a seguir intentándolo. Para prevenir esto, el salto que deben realizar para alcanzar el conocimiento buscado debe ser suficientemente pequeño. No se trata de arrancar una clase sin explicar nada y esperar que aprendan todo por su cuenta. Se debe avanzar hasta el punto en el que les falta sólo un poco de

---

<sup>1</sup> Es clave que se vea estos niveles como un continuo, en cuanto a que algunas actividades podrían estar en niveles intermedios, de acuerdo a los objetivos de cada clase.

conocimiento para completar una tarea y es ese pequeño salto el que se espera que logren de forma independiente. Existe la posibilidad también de que no lo logren. Esto no debe considerarse un fracaso ya que sólo por el hecho de haberlo intentado, el haber probado cosas que no funcionan, el haber generado la intriga por la forma correcta de hacer aquello en lo que están fallando, es lo que favorece la mejor comprensión posterior cuando se revela aquello que necesitaban para completar la tarea en cuestión.

### **Dinámica de trabajo**

El desarrollo de la actividad comienza con una mínima introducción al desafío planteado. No se debe hablar sobre posibles soluciones ni dar pistas sobre cómo encararlo. Sólo se debe limitar a explicar el enunciado. En secuencias de actividades de PilasBloques incluso se puede ni siquiera hablar del enunciado; simplemente se anuncia el nombre del desafío a resolver, se les pide que lo abran y que se pongan a trabajar. La justificación para esto es evitar condicionar a alguien para que arranque apuntando a cierta solución particular. Lo deseable es que empiecen completamente desde cero y realicen todo el proceso necesario para resolver el desafío, incluyendo analizar el enunciado, interpretarlo y empezar a considerar opciones acerca de cómo encararlo.

Tras la presentación se deja una cantidad de tiempo de desarrollo. Durante este tiempo, la idea es que intenten resolver el ejercicio por su cuenta. La cantidad de tiempo depende de la dificultad del ejercicio pero siempre se deja un poco más de lo esperado. Mientras las y los estudiantes trabajan, se recorre el aula observando cómo avanzan. Esto nos permite tener una idea del avance global de la clase. Entonces, si notamos que terminaron, o alcanzaron un nivel de progreso suficiente (a veces no nos importa tanto que terminen el ejercicio si lo que queríamos que hagan ya lo hicieron) podemos dar por finalizada esta etapa. Durante este recorrido es muy importante tratar de no intervenir si no nos piden ayuda, a menos que notemos que están muy desorientados. Si vemos que están avanzando, aunque sea en un camino que sabemos que es incorrecto, es mejor dejarles continuar.

Lo ideal sería que esta etapa dure hasta que cada estudiante haya llegado a una solución más o menos viable pero sabemos que los tiempos son muy distintos y hay quienes terminan mucho antes que el resto. Buscamos un balance para darles tiempo a quienes tardan más pero sin que se aburran quienes terminaron más rápido. En ese momento pasamos a la puesta en común. En esta etapa explicamos todo el procedimiento para llegar a la solución partiendo desde cero. Y no sólo lo explicamos, también lo hacemos en vivo, como si se lo estuviéramos explicando a alguien que ni siquiera leyó de qué se trata el ejercicio.

### **Tipos de actividades en el aula**

Las actividades que se realizan durante una clase pueden clasificarse en cuatro categorías según el nivel de autonomía que las y los estudiantes tienen al realizarla. Como planteó Schwab, para enseñar ciencia primero es necesario hacer trabajar a las y los estudiantes en el laboratorio, antes de realizar una presentación formal de conceptos y principios científicos (Schwab, 1960). Así, el autor describe tres niveles de apertura y permisividad (o *nivel de investigación*, como llama Grau (1994)) para el trabajo de las y los estudiantes, de acuerdo a qué define el manual de trabajo (o, finalmente, la o el docente) y qué queda determinado por los primeros (Schwab, 1962):

**Nivel 1:** La o el docente propone un problema, y describe las formas y medios a través de los cuales la o el estudiante puede descubrir las relaciones que no conocía.

**Nivel 2:** El problema sigue siendo propuesto por la o el docente, pero los métodos y las respuestas quedan abiertas para que las defina la o el estudiante.

**Nivel 3:** El problema, los métodos y las respuestas están abiertos. Esto es, la o el estudiante debe enfrentarse al fenómeno en forma directa.

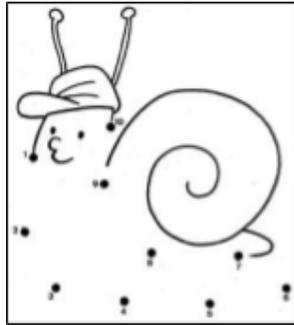
Complementariamente, Herron agregó un nivel cero, en el cual la o el docente define el problema, los métodos, los resultados esperados y las interpretaciones sobre estos resultados (Herron, 1971). Un resumen de estos niveles, junto con el tipo de indagación y el tipo de actividad asociado puede verse en el Cuadro 1.

Nivel	¿Quién define/elabora...?			Tipo de actividad	Tipo de Indagación
	¿Qué?	¿Cómo?	Respuesta		
0	Docente/Libro	Docente/Libro	Docente/Libro	Ejemplo	Confimativa
1	Docente/Libro	Docente/Libro	Estudiante	Ejercicio	Estructurada
2	Docente/Libro	Estudiante	Estudiante	Problema	Guiada
3	Estudiante	Estudiante	Estudiante	Proyecto	Abierta

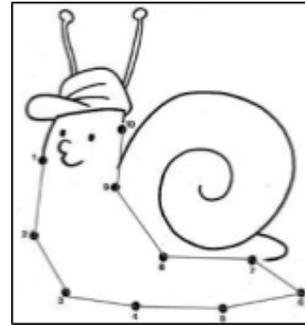
**Cuadro 1:** Tipos de actividades de acuerdo al nivel de autonomía y el tipo de indagación requerida a los estudiantes. Adaptado de (Grau, 1994).

Para ejemplificar cada uno de los niveles proponemos una analogía con distintas actividades en las que el objetivo es que las y los estudiantes realicen un dibujo. La categoría con menor nivel de

autonomía es el *ejemplo*, ya que consiste en que la o el docente realice la totalidad de la actividad mientras el alumnado se limita a observar. En este caso la analogía es una tarea de unir los puntos en la que se ve claramente, incluso antes de comenzar a resolverla, cuál es la figura que se va a dibujar (ver Figura 1).

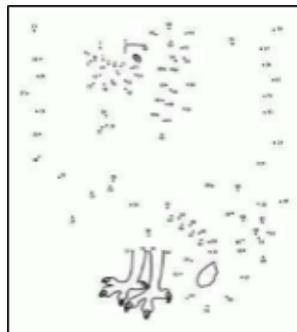


(a)

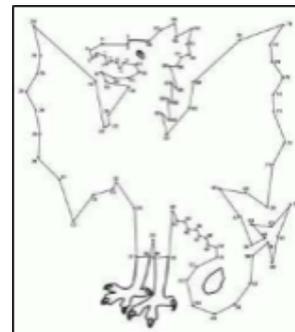


(b)

Figura 1: Analogía de un EJEMPLO de una tarea de unir con puntos. (a) Enunciado de la actividad previo a su resolución. (b) Actividad resuelta.



(a)



(b)

Figura 2: Analogía de un EJERCICIO de una tarea de unir con puntos. (a) Enunciado de la actividad previo a su resolución. (b) Actividad resuelta.

Le sigue la categoría de *ejercicio*, en la que la o el docente presenta la actividad y el alumnado encuentra la solución al desafío propuesto (es decir, lo resuelve). En la analogía, el *ejercicio* también es una tarea de unir los puntos, y alcanza con seguir las pautas del ejemplo, aunque no es tan fácil ver cuál será el resultado hasta haber comenzado a resolverlo (ver Figura 2). La característica clave de un ejercicio es que el alumnado ya sabe cómo resolverlo, dado que tiene las herramientas y la estrategia aunque no la práctica. La siguiente categoría es la de *problema*. En esta categoría, el alumnado no tiene (al menos, no trivialmente) las herramientas para encontrar la solución y, por lo tanto, requiere un proceso previo al de hallar la solución que

consiste en identificar el problema a resolver, analizarlo y evaluar distintas posibles soluciones. En la analogía seguimos teniendo puntos pero ya no están numerados así que hace falta un proceso previo de pensamiento para descubrir cómo resolverlo antes de empezar a hacerlo (ver Figura 3). Además, existe más de una solución posible y no es necesario usar todas las herramientas (los puntos) disponibles para alcanzarla.

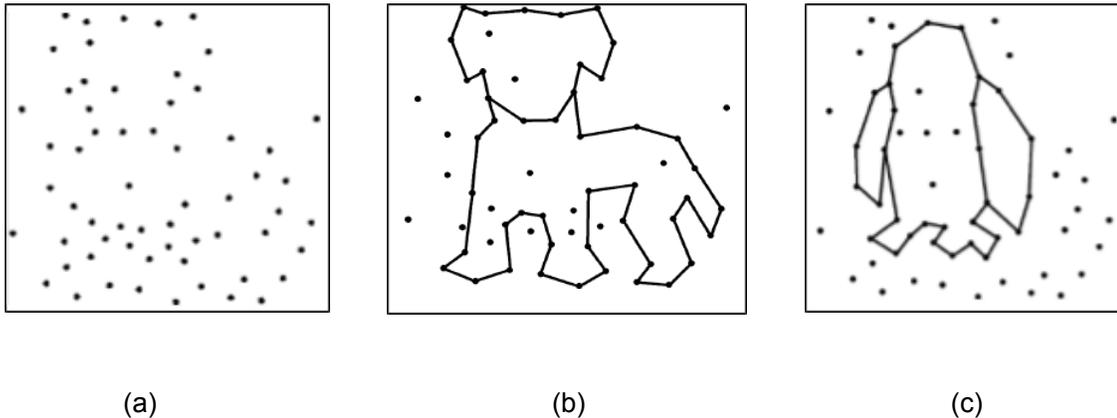


Figura 3: Analogía de un PROBLEMA de una tarea de unir con puntos. (a) Enunciado de la actividad previo a su resolución. (b) Una posible solución. (c) Otra posible solución.

Finalmente se encuentra la categoría de *proyecto* que consiste en que tanto la situación a resolver como la solución a la misma sean propuestas por el alumnado. En este caso la analogía sería una hoja en blanco.

Un mayor nivel de autonomía no implica necesariamente que la clase sea de mejor calidad o haya más aprendizaje, ya que no se está proponiendo que todas las actividades deban ser de tipo proyecto. De hecho, comenzar directamente con una actividad de tipo proyecto es muy tendiente al fracaso, ya que si no se prepara a las y los estudiantes para enfrentarse al síndrome de la hoja en blanco, lo más probable es que no sepan ni por dónde empezar. Se espera que a lo largo de un curso se trabaje con una variedad de actividades que requieren distintos grados de compromiso y autonomía de las y los estudiantes. Así, idealmente, se comenzará con actividades de bajo nivel de autonomía y se irá subiendo en los niveles de indagación de las actividades (Bell & Binns, 2005), de forma de maximizar las chances de tener éxito

### Trabajo con problemas

Según Lester (1983), un problema es una situación que un individuo o grupo quiere o necesita resolver, y para la cual no dispone de un camino rápido y directo que le lleve a la solución. En la misma línea, Varela Nieto (2002) lo plantea como una tarea que, de entrada, no tiene solución evidente, y como consecuencia exige investigación.

En particular, en nuestro enfoque optamos por utilizar los llamados problemas auténticos (Jimenez Aleixandre, 2010), que se diferencian de los problemas tradicionales en varios aspectos (como se resume en el Cuadro 2), los cuales serán centrales al momento de elegir con cuáles trabajar en nuestras clases, para no quedarnos sólo con los ejercicios, sino que podamos aplicar distintos criterios para trabajar con problemas (Pozo Municio & Postigo Angón, 1994).

### Problematización

Un inconveniente recurrente en todos los enfoques de enseñanza es la falta de motivación por parte de las y los alumnos. Así, cuando se resuelven ejercicios en clase es común que se pregunten para qué les sirve aprender a resolver tal tipo de ejercicios. Sabemos que, en general, si tenemos un contenido en nuestro programa es porque tiene una justificación y un posible aprovechamiento a futuro (no necesariamente para resolver problemas, sino también para la formación conceptual y la actitudinal), aunque a veces es difícil explicarlo y se recurre a “ya van a ver más adelante”. Sin embargo, generar motivación más allá de la utilidad real inmediata es una empresa que vale la pena. Por ejemplo, en materias como Física o Matemática suele aparecer el darle un contexto problemático a los ejercicios, más allá de que la tarea no suele requerir más que aplicar una fórmula matemática que ya conocen y usaron previamente. Por ejemplo, se utilizan consignas del tipo "A Carla se le cerró la puerta del tren de larga distancia en la cara y, como el próximo recién pasará mañana, desea alcanzarlo yendo en auto a la próxima estación que está a 5 km. Si el tren va a 50 km/h, ¿a qué velocidad debería ir el auto para llegar al menos 40 seg antes que el tren a la próxima estación?".

Características de los problemas auténticos	Características de los problemas tradicionales
No tiene respuesta obvia o inmediata.	No es necesario comprender el contenido para resolverlos.
Está contextualizado en la vida real.	Resolverlos puede reforzar actitudes superficiales y disuadir a los/as estudiantes de querer comprender el contenido.
Requiere que los/as estudiantes lleven a cabo un proceso de indagación y que diseñen ese proceso con el objetivo de llegar a una solución del problema.	Requieren un aprendizaje ritualizado y simple memorización.
Puede tener varias soluciones diferentes o varios modos de abordaje, todos igualmente válidos.	Su uso para evaluar no mide la comprensión conceptual.

Cuadro 2: Diferencias entre los problemas auténticos y tradicionales, tomado de (Perez, 2017).

Esta estrategia de contextualizar un ejercicio es lo que llamamos *disfrazar*, ya que lo que le agregamos al ejercicio es una *vestimenta* que busca ayudar a la comprensión de la tarea a resolver y hasta puede provocar cierto nivel de motivación por el mismo, pero sin cambiar el procedimiento que debe seguirse para solucionarla.

Un ejercicio *disfrazado* es un buen punto de inicio y es una mejora considerable respecto a un ejercicio donde sólo se pide usar una fórmula dados ciertos datos. Sin embargo, tratamos de ir más lejos y plantear actividades que estén más cerca de la categoría de *problema auténtico* planteada anteriormente. La problematización consiste en mucho más que disfrazar un ejercicio. El punto clave es que la solución no tiene que estar a la vista. En un ejercicio, por más difícil que sea, la estrategia para resolverlo es conocida por las y los estudiantes. Que el ejercicio esté disfrazado no cambia ese hecho. En un problema, en cambio, no hay una estrategia preestablecida que te asegure encontrar la solución al ejecutarla. De hecho, pueden considerarse múltiples soluciones y pueden ser todas correctas.

### Esquemas de trabajo

Hay muchos esquemas para organizar y aprovechar el trabajo con problemas que se encuentran en la práctica y en la bibliografía (Hmelo-Silver, 2004). Todas parten de la elección de un problema que sea interesante para las y los estudiantes. El rol de las y los docentes es el de guiar el trabajo. Algunas opciones que proponemos son las siguientes:

- a) Trabajo experimental (a veces llamado *método científico*)
  - 1) Elegir y discutir un problema auténtico, que tenga muchas conexiones posibles
  - 2) Plantear preguntas interesantes acerca del problema
  - 3) Plantear suposiciones (hipótesis) sobre cuál podría ser la respuesta para cada una
  - 4) Proponer procedimientos para poder verificar las hipótesis (por ejemplo, con experimentos)
  - 5) Ejecutar los procedimientos definidos
  - 6) Analizar los resultados, ¿Son satisfactorios?
  - 7) De ser necesario, plantear nuevas hipótesis y volver al punto 4
  
- b) Aprendizaje basado en problemas (ABP)
  - 1) Elegir y discutir un problema auténtico, que tenga muchas conexiones posibles.
  - 2) Plantear preguntas/subproblemas interesantes sobre el problema general
  - 3) Explicitar qué se sabe y no se sabe sobre el problema y las preguntas/subproblemas planteados

- 4) Determinar qué haría falta aprender para resolver las preguntas/subproblemas, y definir cómo lograrlo
- 5) Completar las actividades de aprendizaje definidas
- 6) Intentar resolver las preguntas/subproblemas
- 7) Si no se tuvo éxito, revisar lo que se sabe y no se sabe, y lo adicional que hay que aprender y volver al punto 3

### **Actividades tipo señuelo**

La idea es comenzar una jornada, taller o encuentro con una actividad para que sirva, en primer lugar para romper el hielo y en segundo lugar, pero no menos importante, para que los y las estudiantes empiecen a incorporar alguna intuición (que puede ser una analogía o metáfora) sobre alguno de los conceptos, generalmente el principal o el más complejo, de los que se van a presentar durante el resto de la clase. La actividad debe ser corta y estimulante y su principal objetivo es entrar en confianza. Es vital cuando las y los participantes no se conocen pero la experiencia demostró que también es muy efectivo cuando pertenecen a un mismo curso, o cuando hablamos de un segundo o tercer encuentro de un taller de varias clases. Decidimos llamarla "señuelo" porque cumple la doble función de, por un lado llamar la atención del público y, por otro, servir de "gancho" para acercar un tema o concepto complejo a una situación más cotidiana o descontracturada.

### **Necesidad y justificación**

Muchas veces nos encontramos frente a un grupo de gente que no se conoce entre sí. Esto sumado a que tampoco nos conocen a las y los docentes dificulta la interacción. Para estos casos es recomendable una actividad de presentación. En los casos en que el grupo ya se conoce, también está el problema de que no nos conocen a las y los docentes. Entonces también es una buena idea generar un clima de confianza antes de comenzar con los contenidos del taller. No conocer al resto de las personas en el aula no es el único impedimento para lograr una buena interacción. En talleres de varios encuentros notamos que incluso después de varias clases, a las y los estudiantes les cuesta arrancar. Es por esto que la función de estas actividades no es sólo conocer al resto sino también descontracturar y perder la timidez.

Por otro lado, nuestra posición de "profesor" puede resultar intimidante para alguna parte del alumnado. La actividad de romper el hielo tiene que permitir que no nos vean como autoridades que vienen a impartir sabiduría porque esto desalienta el tipo de interacción descontracturada que queremos. La visión que queremos que tengan es que vamos a contarles algo de forma interesante y divertida (siempre y cuando no se vaya al otro extremo y perdamos el mínimo

respeto que necesitamos para hacer fluir la clase).

### **Funciones que debe cumplir**

Cuando las y los participantes no se conocen, la principal función es presentarse. En este sentido, la actividad puede ser simplemente una ronda de presentación. Es deseable sin embargo que interactúen entre sí de alguna forma, aunque sea en pequeños grupos (por ejemplo, que se presenten y busquen alguna característica que tengan en común). Si el espacio lo permite, es muy recomendable que la actividad requiera movimiento físico a través del aula (por ejemplo, que tengan que juntarse quienes tengan alguna característica en común). Si no, también es buena idea que realicen algo en su lugar (en el sentido de que lo que tengan que hacer no sea sólo oral, puede ser que tengan que escribir o dibujar o manipular algo que se les entregue).

Una característica deseable es que la actividad tenga alguna (aunque sea muy indirecta) relación con alguno de los temas que se va a desarrollar (por ejemplo, si es un taller de robótica, la actividad puede tratar sobre caracterizar robots o describir su posible movimiento).

### **Algunos ejemplos de actividades**

Es habitual que en algunos talleres de introducción a la programación comencemos con una actividad en la que las y los participantes reciben un dibujo y deben escribir instrucciones precisas para que alguien más pueda reproducir el mismo dibujo sólo a partir de tales instrucciones. Al finalizar el tiempo que tenían asignado, se rotan las instrucciones escritas entre los grupos de forma que cada grupo reciba las instrucciones escritas por otro grupo y se les pide ahora que realicen un dibujo ejecutando las instrucciones que recibieron. En esta primera instancia hay una restricción respecto al tipo de instrucciones que pueden escribir, ya que sólo se puede hacer referencia a figuras geométricas y a sus relaciones entre ellas, pero no se puede hablar sobre aquello que las figuras representan. Es decir, se pueden escribir instrucciones como “*Dibujar un rectángulo de 3x2*” y “*Dibujar un triángulo arriba del rectángulo*” pero no “*Vamos a dibujar una casa. Primero hacer el frente de la casa*”.

Luego se repite todo (asegurando que a cada grupo le toque un dibujo diferente) quitando las restricciones a las instrucciones que se pueden escribir. Al final de la actividad se muestran los dibujos originales y las dos variantes dibujadas (una a partir de las instrucciones restringidas y la otra a partir de las instrucciones libres) para cada uno. La actividad resulta muy divertida y permite generar cierta intuición respecto al funcionamiento de un autómata y de cómo este interpreta instrucciones. Todo esto se puede retomar al empezar a programar y a trabajar con autómatas y comandos primitivos. Otra actividad que tiene mucho éxito es la que usamos como señuelo en el taller de electrónica con Arduino. En ella se le reparte a cada grupo un dispositivo con una placa Arduino, un sensor y un actuador. El objetivo es que interactúen con el sistema para descubrir

cuál es el comportamiento que tiene programado. Esta actividad permite hablar sobre comportamiento automático y sobre los componentes que debe tener cualquier sistema que lo implemente: una placa Arduino, (al menos) un sensor y (al menos) un actuador.

### **Planificación y realización de clases**

Como parte de nuestro enfoque incluimos una propuesta integral para la planificación de secuencias de actividades para el aula, el seguimiento y evaluación de la ejecución de estas planificaciones, además de que se incluyen algunas pautas para la realización de la evaluación de los aprendizajes de nuestras y nuestros estudiantes.

### **Definición de objetivos al planificar**

El primer paso para planificar una clase o un taller debe ser siempre definir el o los objetivos. Una misma clase puede tener muchos objetivos secundarios pero sólo debería tener uno (dos como mucho) principal. El desarrollo de la clase debería darse en función del objetivo principal, considerando en cada actividad cómo esta ayuda a cumplir tal objetivo. Si se detecta que una actividad no aporta al cumplimiento del objetivo principal, esta debería reevaluarse (no necesariamente eliminarse ya que podría bastar una pequeña modificación o agregado para mejorar su condición). Las actividades que se aprueban tras esta verificación pueden entonces modificarse en función de aportar a los objetivos secundarios.

Los objetivos no son cosas que queremos lograr personalmente como docentes, sino que son cosas que queremos que las y los participantes puedan alcanzar y se lleven del curso. Es por esto que siempre deben plantearse desde el punto de vista de las y los participantes. Es decir, un objetivo no puede ser “enseñar el uso de sensores y actuadores en robótica”, sino que podría plantearse como “que las y los participantes aprendan el concepto de sensor y actuador y su uso en robótica”.

Los objetivos deben ser claros y acotados, ya que, si un objetivo es demasiado general, debería poder descomponerse en varios subobjetivos más específicos, los cuales son más simples de evaluar en su cumplimiento. Además, los objetivos deberían poder ser verificables. Esto a veces se puede poner a prueba con una actividad al final de cada encuentro (por ejemplo, con un cuestionario). Un objetivo que es muy complejo o no se puede verificar de ninguna forma es de poca utilidad.

### **Manejo del tiempo**

Planificar la duración de cada actividad es clave. Estimar adecuadamente el tiempo que requiere cada actividad es algo que sólo se puede aprender a través de realizar la misma actividad

reiteradas veces y aún así la estimación puede fallar en algún contexto particular. Esto también implica controlar los tiempos durante la ejecución. Si hay más de una o un docente, quien no esté a cargo de la actividad puede estar anotando los horarios de inicio y finalización de la misma. Esto es extremadamente útil para recalcularse en la próxima ejecución del mismo taller o la misma actividad. También debería anotarse si hubo circunstancias fuera de lo común que hicieron que la duración sea mayor o menor a lo que podría haber sido.

También es una habilidad que se gana con la experiencia pero estar al tanto de los tiempos durante la ejecución sirve para recalcularse los tiempos asignados a cada parte de la actividad. Si se detecta un retraso, se puede acelerar (o directamente descartar) alguna parte de la actividad. Sin embargo, hay que tener cuidado de no apurar los tiempos de las y los estudiantes sólo para poder llegar con todo lo que se tenía planeado. Meter demasiados contenidos en menos tiempo del que requieren es contraproducente. Es preferible descartar una actividad entera que desarrollar dos actividades dedicándoles la mitad del tiempo que realmente requieren. Esto refuerza que al menos una de las actividades logre su objetivo mientras que en el otro caso es probable que no se lleven nada de ninguna de las dos.

Al estimar los tiempos de las actividades siempre hay que ser pesimistas. Si la actividad se realizó varias veces tomamos como estimación la máxima duración (siempre que no haya sido alguna situación especialmente excepcional). Si es una actividad nueva, tratar de dejar suficiente tiempo extra ante imprevistos. Ejecutarla en tiempo real es una buena forma de estimar su duración pero el tiempo requerido debe ser una cota inferior.

No es muy frecuente pero así como una actividad puede llevar más tiempo del estimado, también puede pasar que una actividad se resuelva más rápido de lo que se esperaba. Es por esto que también se debe tener un plan para estas situaciones, contando con actividades extra que sólo se realizarán si el tiempo lo permite.

### **Documentación de la planificación de una secuencias**

Cuando se planifica una clase debería quedar escrito todo lo necesario para poder desarrollarla. Contamos con una ficha de planificación de clases<sup>2</sup> para usar como esqueleto y una base de orientación<sup>3</sup> para guiar la escritura de la misma (ver anexo, donde hay una transcripción de contenido). Lo ideal, aunque inviable por el tiempo que requeriría, es que la documentación disponible permita que alguien que nunca dictó ni presenció la clase la dicte de forma muy similar al criterio de quien la diseñó. Es importante también documentar, luego de ejecutarse la clase, todo aquello que pueda servir para mejorar una próxima instancia de la misma. Esto incluye cómo

---

<sup>2</sup> Ficha de planificación de clases: [https://bit.ly/prender\\_planificacion\\_ficha](https://bit.ly/prender_planificacion_ficha)

<sup>3</sup> Base de orientación de planificación de clases: [https://bit.ly/prender\\_planificacion\\_baseorientacion](https://bit.ly/prender_planificacion_baseorientacion)

resultó, qué inconvenientes surgieron, ideas nuevas en base a algún imprevisto, etc.

### Seguimiento de la clase

Durante la realización de una clase se pueden seguir varios lineamientos para corroborar que se vaya desarrollando de acuerdo a lo planificado. Algunos de estos están resumidos en una base de orientación de seguimiento de clase<sup>4</sup>(ver anexos para más detalles).

### Observación de clases

Al momento de realizar la observación de las clases, ya sea de una o un estudiante que está presentando un tema o de un colega que está dando una clase, se decidió utilizar un protocolo de observación (Anijovich et. al, 2012) basado en la propuesta de la *escalera de realimentación* desarrollada por Perkins como parte del proyecto *Project Zero* (Perkins, 2003), sintetizada en la Figura 4.



Figura 4: Escalera de realimentación de Perkins.

**Clarificar (1)** Pedir aclaraciones sobre los contenidos, el diseño o sobre los objetivos/metas de la clase, entre otros. Por ej: *¿Qué se supone que ya vieron antes de la clase?*

**Valorar (2)** Expresar qué partes o qué ideas/enfoques de la clase son interesantes o innovadores. Por ej: *Fue interesante como, primero, se planteó hacer una actividad sin computadoras sobre repetición condicional, para mostrar los límites de lo actual, antes de programar.*

**Expresar inquietudes (3)** Indicar dudas sobre el diseño de la actividad, plantear desacuerdo en el enfoque planteado, o marcar riesgos o problemas que detectan, entre otros puntos. Por ej: *La actividad es interesante para activar a lxs alumnos, pero se corre el riesgo de generar distracción.*

**Hacer sugerencias (4)** Sugerir mejoras para las inquietudes planteadas antes, plantear extensiones o agregados e indicar sugerencias de temas que podrían ir antes o después de la clase, como para aumentar el impacto o mitigar riesgos.

<sup>4</sup> Base de orientación de seguimiento de clases: [https://bit.ly/prender\\_seguimiento\\_baseorientacion](https://bit.ly/prender_seguimiento_baseorientacion)

Según este protocolo, la realimentación (o, en inglés, *feedback*) que se le brinda a quien ejecutó la clase debe organizarse en cuatro pasos y debe realizarse en el orden dado por la escalera, asegurándose de dar todo el feedback posible de una categoría antes de pasar a la siguiente.

### **Evaluación de los aprendizajes**

Para nuestro enfoque optamos por utilizar la evaluación para el aprendizaje, como se describe en (Anijovich & Cappelletti, 2017).

Habitualmente, se entiende como una perspectiva sobre la evaluación en la que se pone a la o el estudiante en el centro del proceso. “Evaluar para aprender” se refiere a un modo de integrar la evaluación con la enseñanza y el aprendizaje de un modo más auténtico y desafiante para las y los estudiantes, según el cual ellas y ellos aprenden y muestran sus desempeños. Mientras que la evaluación formativa implica ir formando mientras se aprende, y proveer información que contribuye a que el estudiante avance, en “evaluar para aprender” se sigue ese paradigma, aunque se le agregan más cosas. Adicionalmente, existe un cambio de la “evaluación formativa” hacia el de “evaluación para el aprendizaje”, en tanto la segunda lo complejiza y amplía, porque se orienta a pensar en la evaluación como un proceso continuo, que destaca el efecto retroalimentador de la información para docentes y, en especial, para estudiantes. De esta forma, a la hora de valorar los resultados de la enseñanza se hace énfasis en el reconocimiento de los avances individuales y colectivos respecto de un punto de partida, y no solo la comparación con criterios únicos y estandarizados. Así, se debe realizar un balance entre ambas formas de evaluar el desempeño, para hacer que este proceso sea más provechoso y se incremente la capacidad de reflexión y mejora de las prácticas educativas. A continuación se resumen las principales características de un enfoque de evaluación para el aprendizaje, de acuerdo a la descripción en (Anijovich & Cappelletti, 2017):

- **Intenciones explicitadas para el aprendizaje y la formulación de criterios:** Se trata de explicitar a dónde se quiere llegar, qué se pretende que se estudie, y qué se requiere para una situación satisfactoria.
- **Relación pertinente entre la enseñanza y la evaluación, orientada hacia la búsqueda de evidencias de aprendizaje:** La evaluación deberá ser coherente con la programación de la enseñanza y utilizar instrumentos de recolección de información que sean diversos, acordes con la multiplicidad y heterogeneidad de los propósitos didácticos.
- **Diálogo y formulación de preguntas:** Se debe promover el diálogo con los estudiantes, a fin de crear oportunidades para que expresen sus ideas y visibilicen sus concepciones erróneas, y permitir reconocer mejor en qué parte del recorrido de aprendizaje están. De esta forma se contribuye a la práctica metacognitiva, que permite a los estudiantes reconocer en qué

lugar del proceso de aprendizaje se encuentran, así como identificar qué lograron y qué les falta.

- **Retroalimentación o informaciones sobre el proceso:** El equipo docente debe hacer aportes y conversar con sus estudiantes para que puedan identificar y reducir el espacio entre “lo que saben hoy” y el punto al que tienen que llegar, poniendo el énfasis en la tarea que se quiere lograr y no en la persona, en tanto la devoluciones que ponen el foco en el yo (e.g., “buen chico” o “esperaba más de vos”) no sirven para la mejora de los aprendizajes. Se trata de brindar oportunidades para que las y los estudiantes identifiquen los problemas y así desarrollen habilidades de autorregulación del aprendizaje.
- **Autoevaluación y evaluación entre pares:** La evaluación para el aprendizaje implica el tránsito hacia una cultura del aula en la que las y los estudiantes pueden cada vez más juzgar su propio trabajo y el de el resto, comprendiendo cuál es el aprendizaje que se propone que alcancen: para evaluar su propio trabajo y el de sus compañeras y compañeros deben “saber” lo que es un desempeño apropiado (“a dónde tienen que llegar”) y en qué fase están de su propio aprendizaje. Estas acciones se vinculan con procesos de autorregulación y promueven la metacognición. Esta práctica de aula fomenta relaciones horizontales e instala una cultura democrática, en la que el saber de la o el docente “para juzgar” un buen trabajo se comparte con el estudiantado. No se busca separar los procesos de autoevaluación o de evaluación entre pares de los procesos de evaluación que realiza el equipo docente, sino que, por el contrario, el equipo docente puede intervenir formulando preguntas y promoviendo la reflexión acerca de los juicios que emiten las y los estudiantes.
- **Utilización de los datos proporcionados por la evaluación:** Se busca que la evidencia acerca de los logros de las y los estudiantes sea utilizada para informar las decisiones sobre la enseñanza y el aprendizaje. La información obtenida debe crear condiciones para trabajar en la mejora de ambos procesos, en tanto la evaluación tiene que ser entendida como un proceso de diálogo y de mejora. Así, además de obtener un resultado, se desea comprender los objetivos del proceso para saber qué sucede y qué pasos seguir para mejorar.

### **Algunas ideas de cierre**

Se presentó el marco didáctico-pedagógico bautizado como PRENDER (PRopuesta de ENseñanza DuradERa), basado en las ideas de “*Enseñanza para la comprensión*” (Stone Wiske, 1999), “*Aprendizaje por indagación*” (Schwab, 1962; Herron, 1971) y “*Evaluación para el aprendizaje*” (Anijovich & Cappelletti, 2017), entre otras, integrados a partir de la experiencia del grupo en la enseñanza para estudiantes de nivel medio y universitario y el recorrido en la formación de docentes de todos los niveles en temas de didáctica de la programación.

En el contexto actual es cada vez más necesario capacitarse para enseñar contenidos de

programación y de otras áreas relacionadas a las Ciencias de la Computación. La demanda por docentes que puedan dictar estos contenidos está en constante crecimiento en todos los niveles educativos. Esta propuesta tiene un enorme potencial para acompañar el trabajo de aquellas personas que trabajan en el área de la enseñanza de las ciencias de la computación y de la programación en particular.

## Referencias

- Anderson, L.W. (Ed.), Krathwohl, D.R. (Ed.), Airasian, P.W., Cruikshank, K.A., Mayer, R.E., Pintrich, P.R., Raths, J., & Wittrock, M.C. (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's Taxonomy of Educational Objectives*. New York: Longman.
- Anijovich, R. & Cappelletti, G. (2017). *La evaluación como oportunidad*. Buenos Aires, Argentina: Paidós.
- Anijovich, R., Cappelletti, G., Mora, S., & Sabelli, M.J. (2012). *Transitar la formación pedagógica: dispositivos y estrategias*. Buenos Aires, Argentina: Paidós.
- Banchi, H. & Bell, R. (2008). "The many levels of Inquiry". En *Science and Children*, Vol. 46, No. 2, pp. 26–29.
- Bell, R., Smetana, L., & Binns, I. (2005). "Simplifying Inquiry Instruction: Assessing the inquiry level of classroom activities". En *The Science Teacher*, Vol. 72, No. 7, pp. 30–33. Recuperado de <http://www.jstor.org/stable/24138115>
- Bloom, B.S. (Ed.), Engelhart, M.D., Furst, E.J., Hill, W.H., & Krathwohl, D.R. (1956). *Taxonomy of educational objectives: The classification of educational goals. Handbook 1: Cognitive domain*. New York: David McKay
- Grau, R. (1994). "¿Qué es lo que hace difícil una investigación?". En *Revista Alambique. Didáctica de las Ciencias Experimentales*, No. 2, pp. 27–35, Editorial Graó.
- Herron, M.D. (1971). "The nature of scientific inquiry". En *School Review*, Vol. 79, No. 2, pp. 171–212. Hmelo-Silver, C.E. (2004). "Problem-Based Learning: What and How Do Students Learn?". En *Educational Psychology Review* 16, 235–266. doi: 10.1023/B:EDPR.0000034022.16470.f3
- Jimenez Aleixandre, M.P. (2010). *10 Ideas Clave. Competencias en argumentación y uso de pruebas*. Editorial Graó. ISBN/EAN 9788478278978.
- Krathwohl, D. R. (2002). A Revision of Bloom's Taxonomy: An Overview. *Theory Into Practice*, 41(4), 212–218. doi:10.1207/s15430421tip4104\_2.
- Lester, F.K. (1983). "Trends and issues in mathematical problem solving research". En R. Lesh y M. Landau (Eds.), *Acquisition of mathematical concepts and processes*. Nueva York: Academic Press.
- Perez, G. (2017). *Consideraciones didácticas para el armado de talleres en el marco de la Semana de la Enseñanza de las Ciencias*. Informe de uso interno. FCEN, UBA.

- Perkins, D. (2003). *King Arthur's round table: How collaborative conversations create smart organizations*. Hoboken, NJ: Wiley.
- Pozo Municio, J. I. & Postigo Angón, Y. (1994). "La solución de problemas como contenido procedimental en la educación obligatoria". En Pozo Municio et al (1994). *La solución de problemas*. Editorial Santillana.
- Ritchhart, R., Church, M. & Morrison, K. (2014). *Hacer visible el pensamiento*. Buenos Aires: Paidós.
- Schwab, J. (1960). "What Do Scientists Do?". En *Behavioral Science*, Vol. 5, No. 1, pp. 1–27.
- Schwab, J.J. (1962). "The teaching of science as enquiry", en J.J. Schwab and P. F. Brandwein (Eds.). En *The teaching of science*, Cambridge, Harvard University Press, pp. 1–103.
- Stone Wiske, M. et al. (1999). *La enseñanza para la comprensión*. Buenos Aires: Paidós.
- Varela Nieto, P. (2002). *La resolución de problemas en la enseñanza de las ciencias: aspectos didácticos y cognitivos*. Tesis de Doctorado. Universidad Complutense de Madrid: Madrid.

## Anexos

### Anexo 1: Ficha de Diseño de Actividades de Clase

Se requiere armar una secuencia de actividades para desarrollar en clase, que tenga al menos un problema asociado, y para el cual se deberá determinar su solución, y evaluar soluciones alternativas (e incorrectas).

La secuencia tiene que basarse en didáctica por indagación estructurada y deber trabajar sobre alguno de los temas ya vistos. Puede ser totalmente nueva, o una variante significativa sobre algunos de los ejercicios que ya hicieron en el curso. La secuencia podría realizarse en forma presencial o virtual, además de requerir o no el uso de computadoras. Pueden utilizarse cualquiera de las herramientas vistas en clase, o alguna otra de su preferencia, siempre que puedan justificar por qué lo hacen.

En caso de requerir un software a medida, alcanza con proponer el diseño y la solución de la actividad (por más que no la puedan desarrollar ahora; pueden hacer de cuenta como que alguien la programará a futuro).

#### Integrantes del equipo de trabajo

Nombres:	Mails:
----------	--------

#### **Descripción general Nombre de la actividad**

Nombre corto para la actividad

#### **Objetivos primarios de la actividad**

Indicar al menos un objetivo primario (e.g., "Que lxs estudiantes comprendan el concepto de repetición simple"). Los objetivos deben estar expresados en términos de lo que se espera que logren nustrxs estudiantes.

#### **Objetivos secundarios de la actividad**

Indicar al menos dos objetivos secundarios (e.g., "Que lxs estudiantes conozcan los valores numéricos", "Que lxs estudiantes refuercen la división en subproblemas", "Que lxs estudiantes incorporen el uso de un procedimiento principal para toda la solución"). Los objetivos deben estar expresados en términos de lo que se espera que logren nustrxs estudiantes.

### Conceptos en juego

Seleccionar los conceptos principales que se ponen en juego en la actividad. No poner todos los que aparecen en la actividad, sino aquellos por los que podrían catalogar a esta secuencia.

Comandos primitivos	Parametrización
Estrategia de solución	Funciones
Legibilidad y elección de nombres	Expresiones aritméticas (suma, multiplicación, etc.)
Subtareas y procedimientos	Expresiones lógicas (y, o, mayor, menor, etc.)
Repetición Simple	Eventos
Alternativa condicional	Variables
Repetición Condicional, Recorridos	Otros:
Sensores numéricos	

### Público objetivo

#### Nivel educativo de la actividad

Nivel educativo (inicial, primario, secundario, etc.) y grado-año-edad de la población objetivo de esta actividad

#### Características de lxs estudiantes

Incluir las características generales de lxs estudiantes que participarán de la clase, y algunos datos relevantes que pueden ayudar a determinar la orientación de la clase (e.g., Son estudiantes de la orientación en Ciencias Naturales. Se encuentran sumamente apasionadxs con temas relacionados a los dinosaurios)

#### Prerrequisitos para realizar la actividad

Qué temas/conceptos deben haber sido vistos antes del desarrollo de la actividad (e.g., Se deberá conocer el uso de repeticiones simples)

### Descripción General

#### Modalidad de la actividad

¿En qué modalidad se desarrollará la actividad? Podría ser en formato presencial, a distancia/virtual, o de manera híbrida. Si fuera a distancia/virtual o de manera híbrida, indicar qué

plataformas se utilizarían para mantener la comunicación con lxs estudiantes, y cómo se darían las indicaciones.

**¿Requiere uso de Computadora o celular/tablet? Si es así, indicar qué herramientas se utilizarán**

Si la actividad requiriera computadora, celular o tablet, indicar qué plataformas (e.g., Android, iOS) y herramientas se utilizarán (e.g., Pilas Bloques, Gobstones, App Inventor, Robótica en la Escuela, u otros).

**Listado de materiales necesarios**

Indicar una lista de materiales necesarios para desarrollar la actividad por parte del equipo docente, tanto aquellos que deberán construirse (e.g., bloques de gomaeva para indicar el comienzo de los procedimientos y las repeticiones), o aquellos que no requieren elaboración (e.g., al menos 10 colores distintos de marcadores al agua). Incluir acá, también, los materiales que se les pedirán a lxs estudiantes.

**Duración completa de la actividad (en minutos)**

**Descripción de la actividad (x ej. Escenario/s inicial/es, Primitivas)**

Incluir detalles de la dinámica de la actividad, de qué partes constará la misma, y cómo será la participación de los alumnos y las intervenciones de los docentes, entre otras cosas. Debe quedar claro qué parte de la actividad se trabaja en forma individual, grupal y/o colectiva. Recordar dejar espacio para la exploración por parte de los alumnos, así como los diferentes niveles de trabajo mencionados anteriormente.

**Problema y soluciones posibles Enunciado del problema**

Descripción de la situación problemática a intentar resolver por medio de la actividad.

Incluir todos los detalles posibles de contexto. Es importante que el ejercicio a resolver esté debidamente problematizado (i.e., no alcanza con que se pida resolver algo, sino que debe estar planteado de forma de que se presente en un contexto adecuado, y que motive a la resolución). Se recomienda dar espacio para la exploración y la indagación.

### **Soluciones posibles**

Se deberá incluir aquí una o más posibles soluciones de la actividad. También puede hacerse mención a otras soluciones que se podrían esperar, aunque no sean del todo correctas, pero que pueden servir para utilizarse en la clase para aclarar conceptos.

### **Evaluación**

#### **Criterio de evaluación de las soluciones**

Qué cosas evaluarían en las soluciones al problema presentadas por los alumnxs (e.g., el uso adecuado de la división en subtareas para alguna parte concreta del ejercicio). Qué les interesaría remarcar acerca de las posibles soluciones, tanto en los aspectos necesarios/requeridos para una solución correcta, como en las falencias típicas/posibles de aquellas que son erróneas (e.g., usar una múltiples veces el mismo comando primitivo para resolver un problema que sería más legible utilizando repetición simple).

#### **Otras actividades de evaluación**

Qué actividades de evaluación se planea desarrollar a lo largo de la planificación. Indicar claramente en qué momento se realizaría cada evaluación, qué tipo de práctica se realiza, y con qué fin se hace. Además se deberá explicitar quién realizará la evaluación (estudiante o docente) y quién será la persona evaluada.

### **Seguimiento**

Qué actividades se van a realizar de forma de verificar que se está cumpliendo o se cumplió con lo planificado mientras se realizan las actividades o luego de que ellas finalizaron.

### **Más información Observaciones**

Se puede dejar cualquier aclaración sobre el diseño elegido para la secuencia de actividades de clase

### **Links a archivos adicionales**

Pueden incluir aquí hipervínculos a archivos con información extra sobre las actividades propuestas, las soluciones a discutir, o cualquier otra información que crean relevante.

Sugerencia: hacer links a archivos en Google Drive, Dropbox u otros servicios de acceso público. Pueden acortarse usando las páginas cutt.ly ó bit.ly .

## **Anexo 2: Base de Orientación para la planificación y diseño de actividades de clase**

### **Público objetivo**

1. Definir claramente cuál será el público objetivo (aprendices) de la actividad.
2. Describir brevemente la variedad de perfiles de lxs estudiantes del curso (gustos/dificultades con ciertos temas, algunas trayectorias particulares, etc).
3. Definir los conocimientos y habilidades mínimas que deberán tener quienes realicen la actividad.

### **Objetivos**

4. Plantear los objetivos de la actividad, y la meta general que se tiene (si la hubiera).
5. Definir una cantidad reducida de objetivos para la actividad (e.g., 1 principal, y 2 secundarios).
6. Definir los objetivos en función de los cambios que se esperan lograr en lxs estudiantes (e.g., "Que lxs estudiantes aprendan el uso de parámetros para resolver escenarios cambiantes").

7. Verificar que los objetivos principales no sean demasiado generales.
8. Proponer objetivos acotados y alcanzables en el tiempo de la actividad.
9. Verificar que el objetivo definido no sea demasiado ambicioso con respecto a la planificación general (e.g., querer que en una sola clase de 2hs se aprendan los conceptos de repetición simple, alternativa condicional y variables).
10. Definir qué actividades concretas se realizarán en pos de lograr los objetivos definidos.
11. Definir de qué manera se evaluará, a lo largo de la clase, el grado de cumplimiento de los objetivos definidos.

### **Definición de la actividad**

12. Orientar la actividad para satisfacer los objetivos definidos anteriormente.
13. Definir explícitamente los conceptos que se pondrán en juego en la actividad.
14. Definir la actividad para que se focalice en los conceptos elegidos.
15. Definir una primera parte de la actividad (a modo de actividad de 'romper el hielo' y 'anzuelo'), donde se pueda activar a lxs estudiantes tanto física como mentalmente, en la cual se pueda tener un conocimiento preliminar de los conceptos a ver o se puedan llevar una analogía, se pueda motivar al público a seguir atentamente la clase que vendrá, se puedan recabar concepciones previas y alternativas, etc.
16. Relevar implícita o explícitamente las *concepciones previas* que tengan sobre el tema de la clase (ya sean correctas o erróneas).
17. Contemplar en la planificación una o más instancias donde puedan surgir las concepciones previas de lxs estudiantes.
18. Definir la actividad para que dé espacio a lxs estudiantes para la exploración e indagación sobre el problema presentado. Evitar la secuencia de clase expositiva y luego su aplicación. En su lugar, permitir que primero exploren, se choquen con el problema y descubran que les falta algo que aún no tienen (i.e., se haga *indagación estructurada*). Recién entonces se puede presentar el concepto que se espera que aprendan, o ponerle nombre a lo aprendido
19. Planificar una puesta en común para analizar y comparar soluciones alternativas.
20. Verificar la concordancia entre los objetivos y la actividad propuesta.
21. Verificar que lxs estudiantes tengan los conocimientos previos necesarios para la actividad

22. Estimar los tiempos necesarios para cada parte de la actividad, y documentarlos adecuadamente, de forma de poder revisarlos durante la ejecución de la clase.
23. Planificar la actividad para que tenga su mayor asignación de tiempo para aquellas tareas relacionadas con los objetivos primarios y, luego, con los secundarios.
24. Verificar que la actividad sea realizable en tiempo y forma de acuerdo a la planificación (e.g., realizar previamente una o varias soluciones para el problema planteado).
25. Plantear alternativas de la actividad, donde se pongan en juego los mismos conceptos, en caso de contingencias. Por ejemplo, plantear una actividad sin computadoras, para un tema que inicialmente era con computadoras, por si no hubiera disponibilidad de laboratorio.
26. Al hacer la puesta en común del ejercicio, nunca mostrar algo terminado (à lo *Art Attack*). Siempre construir la solución en vivo, al menos en su parte principal. Esto da lugar a que lxs estudiantes tengan tiempo de comprender el proceso de resolución que hay detrás de la solución presentada.
27. Incluir una vinculación de la actividad con la actividad previa y la posterior, y con el contexto general del curso/materia.

### **Problema elegido**

28. Definir el problema elegido de forma que, siempre que sea posible, permita múltiples soluciones.
29. Asegurar que la actividad esté problematizada adecuadamente, de forma de motivar a lxs estudiantes a trabajar con ella.
30. Tener en cuenta y definir varias de las múltiples soluciones que pueden haber en un problema.
31. Pensar variantes para cada ejercicio/problema, de forma de contemplar a quienes puedan encontrar dificultades con las consignas planteadas, así como para 'mantener en la clase' a quienes van más rápido, o para quienes deseen profundizar en el tema visto.
32. Verificar que las actividades hayan sido diseñadas acordemente a los recursos disponibles y restricciones del curso (e.g., disponibilidad de PCs, software necesario, espacio físico requerido, cantidad de docentes, cantidad de alumnos).

### **Evaluación de las soluciones**

33. Definir explícitamente los criterios de evaluación de las soluciones propuestas por lxs

estudiantes, contemplando que puede haber más de una solución válida.

34. Asegurar que se incluya la evaluación de la aplicación de la secuencia 'objetivo' -> 'estrategia de solución' -> 'división en partes' -> 'implementación', así como legibilidad de nombres, entre otros preceptos básicos del curso.

35. Documentar los criterios por algún medio que pueda ser compartido con lxs estudiantes y todo el equipo docente, por ejemplo, por medio de rúbricas.

### **Cierre**

36. Planificar un cierre de clase, que incluya:

a. Los temas que se trabajaron en la(s) clases(s).

b. Las cosas nuevas que se aprendieron.

c. Aplicaciones posibles de los temas vistos.

d. Dificultades que aparecieron en la clase.

e. Limitaciones de las herramientas vistas al momento, cuando aplique (por ejemplo, si en la siguiente clase se verán nuevas herramientas que permitan resolver problemas que ahora no se pudieron resolver totalmente).

f. Vinculación con las próximas clases (o próximos cursos/materias).

### **Otros aspectos relevantes al diseñar una actividad**

37. Tener en cuenta la aparición de estereotipos de género, origen social y cultural, entre otros, en los enunciados de las actividades o problemas usados (e.g., el príncipe que rescata a la princesa), de forma de evitarlos.

### **Anexo 3: Base de Orientación para el seguimiento y evaluación de clases**

En este documento se registran algunas acciones que se pueden seguir al momento de realizar un seguimiento de la ejecución de una clase, en cuanto se quiere medir la

efectividad de la planificación realizada, así como determinar posibles ajustes a próximas clases así como otras instancias de estas mismas clases.

Este documento no busca ser exhaustivo, sino servir de guía práctica para resumir qué tareas se puede realizar para hacer el seguimiento de una clase, y, finalmente, para

determinar el impacto de la misma en lxs estudiantes y en el propio cuerpo docente.

**Para realizar durante la clase**

1. Pedir que un/a par observe la clase, y registre todo lo posible sobre el desarrollo de la misma.
2. Tomar el cronograma detallado de la clase (realizado previamente), y registrar el tiempo real requerido para cada parte de la misma.
3. Evaluar el tipo y volumen de las preguntas planteadas por lxs estudiantes.
4. Registrar notas acerca del impacto de cada actividad de la clase. Por ejemplo, el nivel de motivación e interés logrados en lxs estudiantes, grado de comprensión de la/s tarea/s propuestas, el nivel de desempeño individual, grupal y colectivo, entre otros aspectos.
5. Realizar una puesta en común de todas las actividades que se hagan, donde haya espacio para que lxs estudiantes planteen sus ideas y propuestas de solución a los problemas planteados, y se trate de llegar a un acuerdo común del curso en cuál/es es/son buena/s solución/ones. En esta actividad podrían surgir concepciones erróneas o confusiones sobre los temas vistos, además de otras pistas que indiquen el grado de comprensión de los mismos, por lo que es importante realizarla antes de seguir adelante.
6. Registrar el grado de participación de lxs estudiantes, no sólo en la respuesta de preguntas, sino en la colaboración intra e intergrupos.
7. Registrar el tipo y volumen de las intervenciones de lxs estudiantes (por ejemplo, ver si relacionan el tema de la clase con otros vistos anteriormente).
8. Registrar aquellas situaciones emergentes de la clase, que facilitaron/dificultaron la realización de la clase.
9. Registrar el grado de comodidad y conformidad del equipo docente con el tipo de actividades y el enfoque elegido para la clase.
10. Presentar una actividad para que lxs estudiantes resuelvan a lo largo de la clase, y que la entreguen al finalizar la misma, para poder ser analizada posteriormente.
11. Escuchar los diálogos entre lxs estudiantes durante el trabajo en grupo, y determinar el tipo y profundidad en las aportaciones de cada estudiante (por ej., al momento de argumentar por qué se debería tomar su propuesta de solución y no alguna otra de las en discusión).
12. Plantear preguntas a los grupos de trabajo, o también al curso completo o en forma individual, donde se les pida reflexionar sobre el trabajo que hicieron o están haciendo. Esto

puede servir para conocer el grado de comprensión del problema y sobre los métodos vistos (e.g., "¿Por qué optaron por usar esta herramienta del lenguaje, en vez de esta otra?", "¿Hay otras soluciones aparte de la que hicieron?")

13. Mantener registro de aquellos grupos/estudiantes que van más despacio o más rápido, de forma de presentarles variantes de las actividades, de acuerdo a lo previsto en la planificación.

14. Registrar si ocurre que lxs estudiantes se quieren quedar en clase al momento del recreo (si lo hay) o, quizá, directamente se olvidan que hay un corte.

15. Realizar un registro de los grupos de estudiantes trabajando durante la clase (idealmente, en una o más fotografías o videos), para analizar posteriormente.

16. Determinar breves ejercicios/preguntas de control (*checkpoints*), en distintas partes de la clase, donde se pueda verificar el nivel de comprensión de lxs estudiantes sobre los temas vistos, y se pueda determinar distintos cursos de acción para la clase, a partir de los resultados obtenidos. Por ejemplo, se podría realizar un cuestionario corto luego de la primera actividad y a la mitad de la clase, por medio de alguna herramienta online (e.g., Kahoot).

17. Tratar de conocer cómo fue la percepción subjetiva del tiempo para lxs estudiantes, por medio de alguna actividad breve (por ejemplo, preguntarles "¿cuánto tiempo creen que pasamos haciendo X tarea?" o "¿la clase les pareció corta o larga?").

18. Realizar una actividad de cierre, donde se puedan poner en práctica/discusión los temas vistos en la clase.

19. Plantear un ejercicio pequeño que quede de tarea, para que lxs estudiantes lo resuelvan para la clase siguiente.

### **Para realizar luego de la clase**

20. Realizar una reunión de feedback con el/la colega que observó la clase, para que comparta sus percepciones sobre el desarrollo de la misma, y pueda dar feedback.

21. Analizar los registros del desarrollo de la clase (notas, fotos y/o videos), de forma de ver el tipo de interacciones de lxs estudiantes, y si estas se corresponden con las esperadas.

22. Analizar si las actividades de la clase tuvieron los resultados esperados en lxs estudiantes (e.g., nivel de complejidad percibida, efecto en la activación de lxs estudiantes, grado de compenetración logrado, etc), además de en los docentes (e.g., comodidad al trabajar con las actividades, conformidad con el desarrollo de la actividades y de la clase en general, etc).

23. Analizar los productos resultantes de la clase (soluciones a los ejercicios entregadas) para determinar el nivel de comprensión de lxs estudiantes.
24. Analizar el grado de cumplimiento de cada uno de los objetivos planteados en la etapa de planificación, y, finalmente, evaluar si debe modificarse la lista de objetivos.
25. Analizar el tiempo efectivo requerido por cada parte de la clase, en comparación con el tiempo planificado, e identificar posibles causas de la diferencia encontrada (por ej., “el ejercicio planteado era muy simple”, “supuse que ya habían visto X tema antes, pero no era cierto”).
26. Realizar un análisis cuantitativo de las calificaciones obtenidas en los temas vistos en la/s clase/s en comparación con los otros del curso, así como en anteriores períodos u oportunidades en que se dio el curso.
27. Pedir a lxs estudiantes que cuenten qué fue lo que se hizo y se aprendió en la clase anterior, al comienzo de la próxima clase que se tenga.
28. Realizar preguntas de repaso al comienzo de la clase siguiente, donde no se pongan en juego sólo las definiciones de los temas vistos, sino que se requiera poner a prueba la comprensión de lxs estudiantes (por ejemplo, en vez de preguntar “¿Qué es un procedimiento?”, se podría preguntar “¿En qué casos usamos procedimientos, y por qué?”).
29. Registrar las actividades autónomas realizadas por lxs estudiantes luego de la clase, por encima de lo requerido en el curso (por ejemplo, si intentaron/pudieron resolver problemas más avanzados con las herramientas vistas).
30. Registrar si lxs estudiantes piden saber más sobre alguno de los temas vistos en clase, o si piden más información o recursos para aprender más.
31. Realizar una o más encuestas a lxs estudiantes durante el curso (e.g., al terminar un tema y/o al final del bimestre/semestre/año), de forma de tener más información sobre cómo les resultaron las clases, en general, y pedir feedback sobre cada uno de los temas vistos.
32. Consultar con docentes de cursos subsiguientes acerca del nivel de manejo de los temas vistos en la/s clase/s.

## Anexo 4: Prácticas de Evaluación y su caracterización

Práctica de evaluación	Nivel educativo de aplicación	¿Cuándo se realiza?	¿Qué tipo de práctica es?	¿Con qué fin?	¿Quién realiza la evaluación?	¿Quién es la parte evaluada?	¿Quién aprovecha directamente los resultados de la evaluación?
Participación en clase	Todos	Todas las clases	Reflexión oral	Registrar-verificar	Docente	Estudiante	Docente
Realización de TPs	Secundario+	Al final del bimestre/cuatrimstre (se puede trabajar durante todo el curso)	Ejercicios; problemas	Ofrecer devoluciones	Docente	Estudiante	Docente
Solicitar producciones transversales sobre el tema (e.g., producción artística)	Todos	Al final de un tema/unidad/bimestre	Producción de material; Desempeño flexible	Registrar-verificar; Ofrecer devoluciones	Docente	Estudiante	Docente
Explicar con sus palabras el tema	Todos	Todas las clases	Reflexión oral escrita	Registrar-verificar; Ofrecer devoluciones	Docente	Estudiante	Docente
Reporte de actividades (e.g., llena una planilla donde se indica qué cosas hicieron)	Primario (2do ciclo) y secundario+	Al final de un tema/unidad/bimestre		Registrar-verificar	Estudiante	Estudiante (e mismo que evalúa)	Docente
Relacionar el tema visto con algo de la vida cotidiana	Todos	Todas las clases	Reflexión oral escrita	Registrar-verificar; Ofrecer devoluciones	Docente	Estudiante	Docente
Análisis de las preguntas que realizan	Todos	Todas las clases	Reflexión oral escrita	Registrar-verificar	Docente	Estudiante	Docente
Hacer un proyecto grupal	Primario y secundario+	Al final de cada unidad/bimestre	Proyecto	Registrar-verificar; Diagnosticar-predecir; Seleccionar, clasificar y jerarquizar; Certificar-promover	Docente	Estudiante	Docente

Práctica de evaluación	Nivel educativo de aplicación	¿Cuándo se realiza?	¿Qué tipo de práctica es?	¿Con qué fin?	¿Quién realiza la evaluación?	¿Quién es la parte evaluada?	¿Quién aprovecha directamente los resultados de la evaluación?
Hacer un proyecto de integración de a pares (integrando varias)	Primario y (preponderante en) secundario+	Al final del año	Proyecto	Certificar-promover; Ofrecer devoluciones	Docente	Estudiante	Institución / Distrito / Provincial
Evaluación escrita a desarrollar	Primario y (preponderante en) secundario+	Al final de cada unidad/bimestre	Preguntas abiertas	Registrar-verificar; Diagnosticar-predecir; Seleccionar, clasificar y jerarquizar; Certificar-promover	Docente	Estudiante	Docente
Evaluación multiple-choice (con o sin justificación)	Primario y (preponderante en) secundario+	Al final de cada tema/unidad/bimestre	Preguntas cerradas	Registrar-verificar; Diagnosticar-predecir; Seleccionar, clasificar y jerarquizar; Certificar-promover	Docente	Estudiante	Docente
Asignar tarea en forma individual	Primario y (preponderante en) secundario+	Cuando fuera necesario	Todas	Registrar-verificar	Docente	Estudiante	Docente
Evaluación diagnóstica (antes de dar un tema)	Todos	Antes de comenzar con un tema/unidad	Preguntas abiertas/cerradas; reflexión oral o escrita; ejercicios	Diagnosticar-predecir	Docente	Estudiante	Docente
Hacer un seguimiento desde una evaluación diagnóstica y evaluaciones posteriores	Todos	A lo largo de todo un curso	Todas	Registrar-verificar; Diagnosticar-predecir; Ofrecer devoluciones; Seleccionar, clasificar y jerarquizar; Certificar-promover	Docente	Estudiante	Docente

Práctica de evaluación	Nivel educativo de aplicación	¿Cuándo se realiza?	¿Qué tipo de práctica es?	¿Con qué fin?	¿Quién realiza la evaluación?	¿Quién es la parte evaluada?	¿Quién aprovecha directamente los resultados de la evaluación?
Corrección de tareas de a pares (co-evaluación)	Secundario+	A lo largo de todo un curso	Tutoría, docencia	Ofrecer devoluciones	Estudiante (o grupo de estudiantes)	Estudiante (o grupo de estudiantes) distinto	Docente
Cuestionarios	Primario y secundario+	Al final de cada clase/tema/unidad	Preguntas cerradas	Registrar-verificar; Diagnosticar-predicir; Seleccionar, clasificar y jerarquizar; Certificar-promover	Docente	Estudiante	Docente; Estudiante
Hacer apunte colaborativo de la clase	Secundario+	Al final de cada tema/unidad/bimestre	Producción escrita	Registrar-verificar; Ofrecer devoluciones	Docente; Estudiante	Estudiante	Docente; Estudiante
Debate de cierre en la clase	Secundario+	Al final de la clase	Reflexión oral	Registrar-verificar; Ofrecer devoluciones	Docente; Estudiante	Docente; Estudiante	Docente; Estudiante
Armar mapas conceptuales al final de cada clase	Secundario+	Al final de la clase/tema/unidad	Reflexión escrita (esquemas)	Registrar-verificar; Ofrecer devoluciones	Docente	Estudiante	Docente; Estudiante
Juego donde se pongan a prueba los conceptos vistos en la clase	Todos	Al final de la clase/tema/unidad	Desempeño flexible	Registrar-verificar; Ofrecer devoluciones	Docente	Estudiante	Docente
Resolución de ejercicio (con/sin trabajo individual)	Todos	En todo momento	Ejercicio	Registrar-verificar	Docente	Estudiante	Docente
Realizar preguntas al curso	Todos	En todo momento	Reflexión oral	Registrar-verificar; Ofrecer devoluciones	Docente	Estudiante	Docente
Reflexión sobre las tareas realizadas	Primario y secundario+	Al final de la clase/tema/unidad	Reflexión oral escrita	Ofrecer devoluciones	Estudiante	Estudiante (el mismo que evalúa)	Docente; Estudiante
Armar el enunciado a partir de un solución	Primario y secundario+	Al final de la clase/tema/unidad	Desempeño flexible	Registrar-verificar; Ofrecer devoluciones	Docente	Estudiante	Docente

## Medición del Pensamiento Computacional

Fabio Fernando González  
fabio\_gon@hotmail.com

Escuela Provincial de Artes Visuales “Prof. Juan Mantovani”

### Resumen

El presente artículo explora el concepto de Pensamiento Computacional y muestra los resultados de una experiencia en la que se pretende realizar una medición del mismo, buscando evidencias de la influencia de una experiencia de programación en el aula.

El trabajo busca evidencias medibles del desarrollo del PC, luego de realizar una experiencia de desarrollo de videojuegos programados por los estudiantes en el entorno Scratch. La experiencia fue realizada en dos cursos (1er. y 2do. año) del nivel secundario de una escuela en la ciudad de Santa Fe durante el año 2022, sobre una muestra de 29 casos, siguiendo la metodología pre-test / post-test.

**Palabras claves:** programación, medición, pensamiento computacional, Scratch.

Antes de desarrollar la experiencia, aclararemos algunos conceptos que suelen mencionarse frecuentemente en publicaciones especializadas, pero muchas veces con interpretaciones muy variadas.

### Codigoalfabetización

El término “codigoalfabetización” (del inglés “code-literacy”) ha sido acuñado recientemente para referirse al proceso de enseñanza-aprendizaje de la lectoescritura con los lenguajes informáticos de programación (Román, 2016).

Una persona codigoalfabetizada es capaz de leer y escribir en el lenguaje de las computadoras y otros dispositivos digitales, y por ende, de pensar computacionalmente.

El propósito del proceso de codigoalfabetización en la escuela, no radica en un fin meramente utilitarista, de “formar programadores”. Si bien quizás eso también sea una consecuencia, no es el propósito más importante. Estableciendo ciertas analogías, cuando enseñamos a sumar y restar no pensamos que lo hacemos para que haya más científicos matemáticos, ni enseñamos a leer y escribir para que haya más periodistas, novelistas o escritores profesionales.

Cuando un niño aprende a leer y escribir, domina un modo de acceder a muchos otros saberes. Se les abren innumerables oportunidades para aprender otras cosas. Cuando se aprende a leer se puede, a partir de ahí, seguir aprendiendo. Lo mismo sucede con la codificación: si aprende a codificar, puede aprender codificando (Resnik, 2012). Podríamos pensar en este tipo de saberes, como habilidades instrumentales: herramientas que nos permitirán luego, acceder a nuevos saberes. Un saber que funciona como una llave, que nos abre puertas de nuevos conocimientos.

El proceso de codigoalfabetización resulta valioso en tanto nos permite activar otras potencialidades. Se trata de que veamos a las computadoras como un medio expresivo, que despliegue la creatividad de los sujetos en un nuevo lenguaje. En este paradigma, las ciencias de la computación estarían más cerca del teatro, del dibujo o de la música, que de las matemáticas y las ciencias exactas.

### **Pensamiento Computacional**

Cuando hablamos de Pensamiento Computacional (de ahora en más abreviado PC) nos referimos a un concepto muy abstracto, que llega a considerarse intangible. No existe consenso sobre una definición formal. Ha sido un tema de constante debate dentro de la comunidad científica y no se llega a acordar qué debe incluir una definición general y completa del mismo.

Desde las primeras aproximaciones de Seymour Papert hasta las perspectivas más actuales y tan en boga en diversos ámbitos educativos, ha sido objeto de estudio de numerosos investigadores. En sus estudios, Papert describió un tipo de pensamiento que denominó “pensamiento procedural”. Sostenía que era una poderosa herramienta intelectual que debería ser adquirida por todos los niños. El fundamento era que más allá de las capacidades para manejar una computadora, y desarrollar ideas computacionales, era una habilidad que trascendía el mero ámbito informático: dotaba a los niños de estructuras aplicables a otros espacios de conocimiento (Papert, 1980).

A principios de este siglo, el concepto es retomado por Jeannette Marie Wing, profesora del Departamento de Computación de la Universidad de Carnegie Mellon, quien utilizó el término PC en un artículo queriendo describir cómo piensa un científico de computadoras.

Jeannette Marie Wing entiende el PC como el abordaje de problemas haciendo uso de los conceptos fundamentales de la informática, es decir, pensar como un científico de la computación cuando uno se enfrenta a un problema (Wing, 2006).

“El pensamiento computacional implica resolver los problemas, diseñar sistemas y entender el comportamiento humano, aprovechando los conceptos fundamentales para las ciencias informáticas. El pensamiento computacional incluye una gama de herramientas mentales que

reflejan la amplitud del campo de la informática.” (Wing, 2006, p. 33).

Esta definición, es considerada la primera definición formal del “Pensamiento Computacional” o “Computational Thinking” y en el año 2006 es la primera vez que aparece este concepto dentro de la comunidad científica en computación (Polanco Padrón, N., Ferrer Planchart, S., y Fernández Reina, M., 2021).

En 2008, Wing agregó, en la misma línea, que “el Pensamiento Computacional incluye los procesos de pensamiento implicados en la formulación de problemas y de sus soluciones, de tal modo que éstos sean representados de una manera que pueda ser abordada efectivamente por un agente-procesador de información” (Wing, 2008 p. 3718), como mencionó (Román, 2016, p. 145).

En 2012 Alfred Aho define al PC como el proceso de pensamiento involucrado en la formulación de problemas de tal manera que sus soluciones puedan ser representadas como pasos computacionales discretos y algoritmos (Aho, 2012).

En 2016, Román define el PC como una habilidad cognitiva; un constructo psicológico que se puede definir como la capacidad de formular y solucionar problemas aplicando conceptos computacionales (descomposición, secuencias, bucles, condicionales, funciones, variables), y siguiendo una lógica algorítmica (Román, 2016). Entendemos a las habilidades cognitivas como capacidades o aptitudes del ser humano relacionados con el procesamiento de la información, es decir, los que implican el uso de la memoria, la atención, la percepción, la creatividad y el pensamiento abstracto o analógico.

En el año 2021, las investigadoras venezolanas Norailith Polanco Padrón, Sonia Ferrer Planchart y Mariana Fernández Reina, trabajando en la Universidad del Zulia (Venezuela) realizan una larga y meticulosa recopilación de definiciones del PC disponibles en toda la bibliografía especializada. Ahí se pueden analizar y detectar numerosos y distintos enfoques. El PC puede ser visto como: “herramienta mental”, “actividad mental”, “procesos de pensamiento”, “proceso de resolución de problemas”, “habilidades de pensamiento”, “visión del mundo”, “metodología”, “método”, “capacidad para resolver problemas”, “proceso mental”, “manera de pensar”, “paradigma de trabajo”, “estrategia”, “actividades y habilidades”, “forma de pensar” y “tipo de pensamiento”. La lista destaca el enfoque de cada autor acerca del pensamiento computacional a partir del elemento en el cual pone el énfasis de su definición (Polanco Padrón y otros, 2021).

Si no se ha logrado aún un consenso en la definición del PC, es entendible que otros aspectos tampoco estén todavía resueltos con certeza. No hay una sola metodología aceptada y estandarizada para la medición del desarrollo de PC. Por ende será muy difícil que se gane su

lugar en los diseños curriculares.

Del mismo modo, existen discrepancias sobre cómo debería ser integrado en los currículos educativos. Si bien la “programación” está siendo incluida en los diseños curriculares de todo el mundo para desarrollar el PC, no existe consenso sobre en qué niveles se debe trabajar, bajo qué modalidades, con qué herramientas, etc.

### **Pensamiento Computacional “desenchufado”**

Muchas de estas aproximaciones al concepto de PC, lo relacionan de manera directa con los dispositivos digitales (ordenadores); tanto como origen (“hacer del modo en que lo hace la computadora”) como destino (“hacer para que lo entienda la computadora”).

Pero para el desarrollo de las habilidades relacionadas al PC, existe una gran variedad de alternativas “desenchufadas” (o desconectadas), que prescinden de tecnologías digitales. Otras, en cambio, hacen uso de tecnología (tanto software como hardware), entre las que se encuentran la gamificación, la programación y la robótica educativa.

Pero para lograr un mayor nivel de generalización, es menester independizar al PC de la presencia de un dispositivo digital. Volviendo a la idea de Papert (1980) y Wing (2006), que desdoblaron el PC en dos líneas: en una línea rescata las capacidades que se pueden aplicar intrínsecamente al trabajo en sistemas informáticos, y otra línea que se centra en habilidades que se pueden aplicar en un ámbito más general, que podríamos resumir como habilidades para la búsqueda de soluciones de problemas (extrainformáticos), aplicando metodologías cercanas al campo de las computadoras.

Entonces, en el presente trabajo, entendemos al PC como el **proceso intelectual llevado a cabo por un ser humano, que permite buscar la solución de una amplia variedad de problemas de distinta índole, haciendo uso de ciertas habilidades, cercanas pero no exclusivas de la computación.**

### **Principios del Pensamiento Computacional**

Existen cinco conceptos que se consideran los principios del PC:

- **Descomposición:** dado un problema de gran complejidad, se lo subdivide en una serie de problemas de menor envergadura, cada uno más sencillo y abordable.
- **Reconocimiento de patrones:** algunos de los pequeños problemas que surgen de la descomposición, pueden resultar similares entre sí; entonces se tratará de encontrar una solución general que sirva para todos los problemas que se parecen.

- **Abstracción:** se omitirá toda información que resulte irrelevante al problema propuesto.
- **Algoritmos:** estrategia plasmada en una secuencia ordenada de operaciones que permite hallar la solución del problema.
- **Evaluación:** capacidad de poner a prueba las soluciones creadas en busca de errores (debugging) o de soluciones más eficientes.

La palabra computacional lleva a pensar la condición sine qua non de la utilización de una computadora o de un dispositivo tecnológico. Pero si bien la ejecución de un algoritmo será realizada más eficientemente por un dispositivo digital, se prioriza la perspectiva de entenderlo como un modo de pensar soluciones a problemas.

Entonces;

- No es un proceso (necesariamente) llevado a cabo por computadoras, ya que hacemos referencia a un proceso mental. Somos los humanos, y no las computadoras, los que podemos realizar PC, o utilizarlo para guiar nuestros procesos.
- No (necesariamente) son problemas informáticos, ya que el concepto de problema puede abarcar una gran cantidad de tópicos: desde problemas matemáticos, a otros problemas del mundo real que pueden abordarse, aunque parezcan muy diferentes, usando los mismos principios.
- Hablar de PC es entender que la programación no tiene que ser sólo para programadores.

### **Medición del Pensamiento Computacional**

Al no existir consenso entre la comunidad educativa sobre la definición exacta de PC, por consiguiente, existe un vacío respecto al desarrollo de metodologías y herramientas para su evaluación (Román, 2016). Esto conlleva un problema a la hora de evaluar una propuesta pedagógica, ya que “para poder juzgar la efectividad de cualquier currículo que incorpore el pensamiento computacional, es necesario validar previamente instrumentos de medida que permitan a los educadores evaluar en qué grado los estudiantes han desarrollado esa habilidad de pensamiento” (Grover y Pea, 2013, pp.13).

Para nuestra experiencia, luego de analizar las distintas alternativas, se decidió implementar una adaptación del Test de PC de Marcos Román González (2016). Este test fue desarrollado en el

marco de la tesis doctoral “Codigoalfabetización y PC en Educación Primaria y Secundaria: validación de un instrumento y evaluación de programas”.

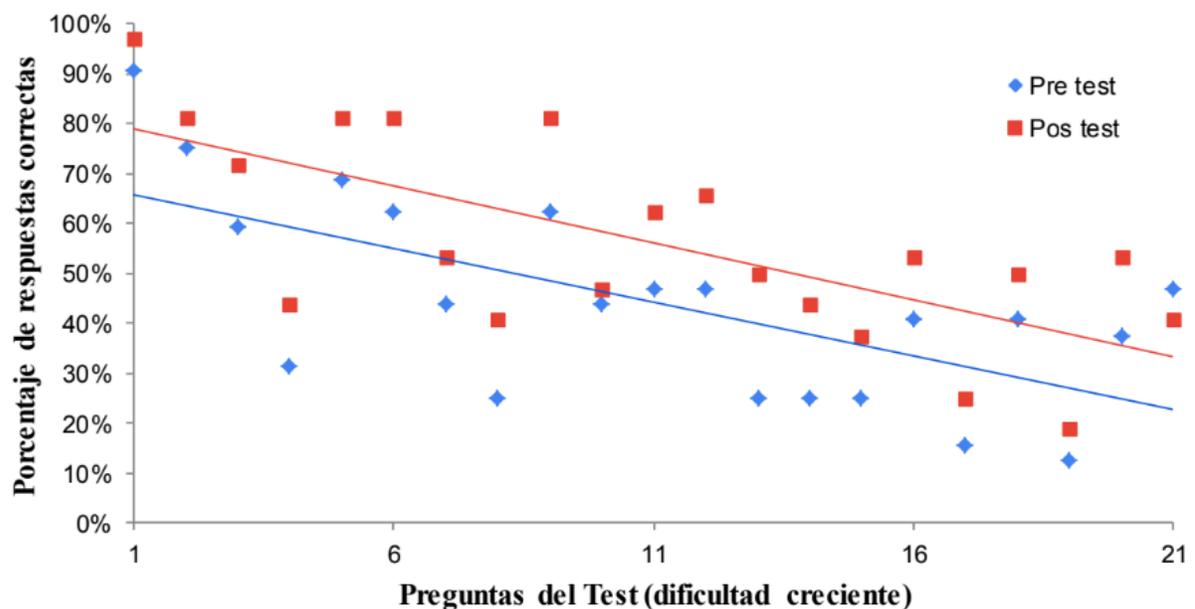
En su versión original el test es una prueba objetiva de elección múltiple con 4 opciones de respuesta (sólo 1 correcta) de 28 ítems de longitud. Las preguntas se dividen entre las del tipo “laberinto” y las del tipo “bloques”. Están organizadas con un nivel de dificultad creciente. Se puede acceder al mismo en <http://goo.gl/IYEKMB>

### La investigación

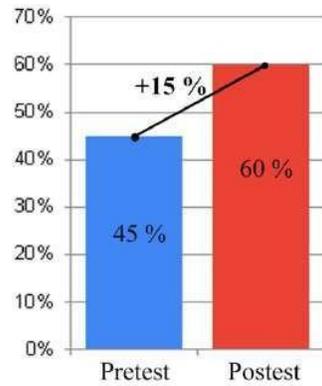
La investigación consistió en verificar si se produce una incidencia medible del desarrollo del pensamiento computacional luego de realizar una experiencia de desarrollo de videojuegos programados por los estudiantes en el entorno Scratch.

La experiencia fue realizada en dos cursos (1er. y 2do. año) del nivel secundario de una escuela en la ciudad de Santa Fe durante el año 2022, sobre una muestra de 29 casos.

El test se diseñó en Google Forms, y se implementó en dos oportunidades. La primera (pretest), antes de realizar la experiencia de desarrollo de videojuegos con Scratch. La segunda (postest), la semana siguiente de finalizado el desarrollo de los contenidos.

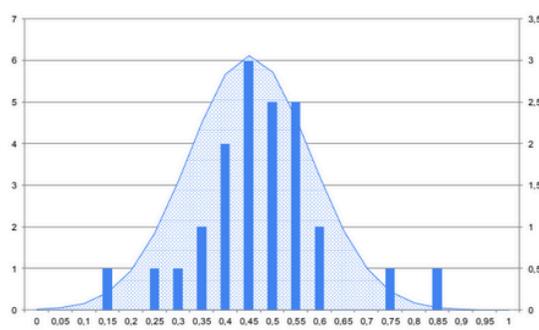


El resultado más significativo fue la mejora del 15% en los porcentajes de acierto entre los resultados del pretest (test realizado antes del taller) y los del postest (test realizado luego del taller).

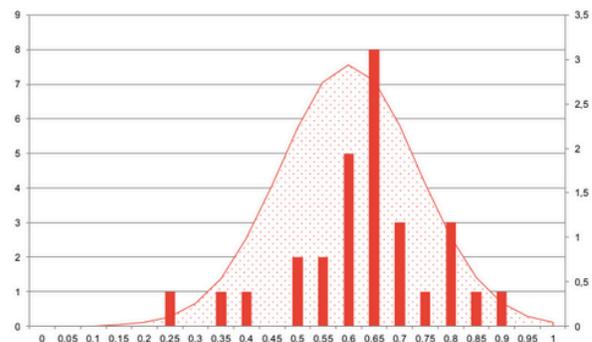


**Distribución de las puntuaciones totales en el pretest**

La distribución de las puntuaciones totales en el test se ajusta a la curva normal, también llamada distribución de Gauss, o gaussiana, con la clásica forma acampanada. Es previsible, ya que es un test que mide una aptitud del tipo cognitivo.



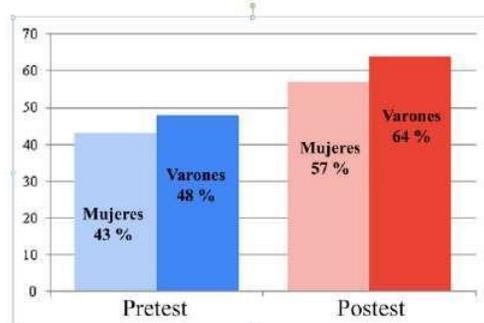
Pretest



Posttest

En la comparación de la distribución de las puntuaciones de, pretest y el posttest, vemos un desplazamiento de la curva hacia la derecha, evidenciando una mejora en el rendimiento de los estudiantes.

**Evolución del PC entre el pretest y el posttest segmentado por género**



Se observó una diferencia de rendimiento en función del género. Tanto en el pretest como en el postest, los varones obtuvieron mejor rendimiento que las mujeres.

Comparando el desempeño en el pretest, discriminado por género, el porcentaje de aciertos de las mujeres (43 %) fue menor al de los varones (48 %).

Del mismo modo, el desempeño en el postest, discriminado por género, muestra que el porcentaje de aciertos de las mujeres (57 %) fue menor al de los varones (64 %). Sin embargo, la evolución del PC fue positiva (mejora) en ambos géneros, en magnitudes similares.

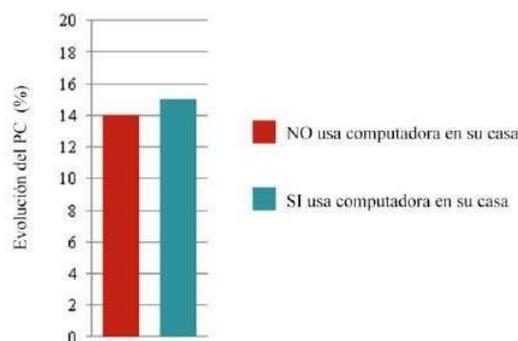
En general se pudo observar en de la mayoría de las chicas, una actitud refractaria y poco entusiasta, manifestando que no sentían que fuera una propuesta en la que ellas se sintieran involucradas. Esta autopercepción de bajo rendimiento en tareas relacionadas a la informática y cuestiones tecnológicas en general, desalienta a las chicas a relacionarse con estos temas, pudiendo influir en elecciones académicas que han de reforzar este sesgo de género, arrojándonos a un indeseable círculo vicioso (Soe & Yakura, 2008).

Posteriores investigaciones podrán indagar y determinar en qué medida estas diferencias provienen de una habilidad cognitiva con divergencias entre hombres y mujeres, y en cuánto responden a factores sociales y culturales que condicionan y determinan comportamientos esperables en chicas y chicos.

### **Evolución del PC en relación al uso de computadora en su casa**

Se le preguntó a los estudiantes en el postest: “¿Usás computadora en tu casa?”, pudiendo responder “Sí” o “No”. De ahí surge la variable “Uso de computadora en casa”.

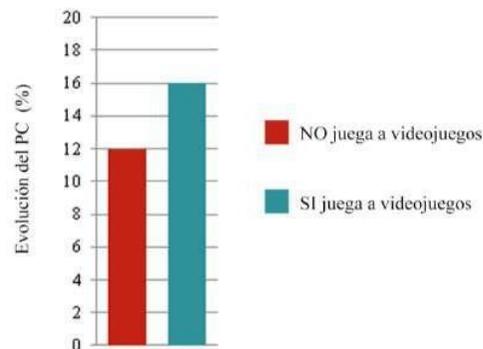
Los estudiantes que usan computadoras en su casa tuvieron una Evolución del PC levemente superior (+ 15 %) con respecto a los que no usan computadoras en su casa (+ 14 %).



### **Evolución del PC en relación al hábito de jugar con videojuegos**

Se le preguntó a los estudiantes en el postest: “¿Jugás a videojuegos?”, pudiendo responder “Si” o “No”. De ahí surge la variable “Juego a videojuegos”.

Los estudiantes que juegan con videojuegos tuvieron una Evolución del PC levemente superior (+ 15 %) con respecto a los que no juegan a videojuegos (+ 12 %).



Estas variables sirvieron para demostrar dos suposiciones previas: que el uso de una computadora en la casa y el hábito de jugar videojuegos son dos condiciones que mejorarían los resultados de la experiencia de programación con Scratch. Ambas suposiciones previas pudieron ser ratificadas, siendo la influencia del hábito de jugar con videojuegos levemente superior que la influencia del uso de computadoras en la casa.

### **Conclusiones**

Esta experiencia a pequeña escala puede servir de prueba piloto, y partir de aquí proponer ampliar la muestra, diversificar los entornos y metodologías. Obtendremos un valioso feedback que nos permitirá redireccionar nuestras prácticas, y obtener evidencias medibles del impacto de la implementación de contenidos relacionados a la programación y las ciencias de la computación en general en todas las aulas.

Creemos que la programación es una oportunidad para el desarrollo de habilidades que permiten a los estudiantes encontrar soluciones innovadoras, de manera autónoma y colaborativa. Por eso estamos convencidos de la necesidad de que la escuela secundaria orientada implemente un espacio curricular propio. El mismo debería tener un diseño curricular acorde a la complejidad de la disciplina y estar a cargo de docentes especializados (profesores de informática). Además, se debería contar con la infraestructura y los recursos materiales necesarios, y un plan de

actualización continuo para que los docentes estén siempre al día, en una disciplina en constante evolución.

Pero también debemos ser cautos. La impronta innovadora que trae del uso de tecnologías digitales en el aula, no debe hacer que todas estas propuestas sean aceptadas sin más análisis, seducidos por el halo de novedad. Debemos darle un sentido y una razón de ser. Y debemos propiciar la receptividad de lo nuevo, no sólo por nuevo, y la no-negación de lo viejo, sólo por viejo (Freire, 1965).

Luego de este recorrido, no nos cabe duda de la posición que debemos tomar ante la disyuntiva “programar o ser programados” (Papert, 1981): en propiciar un rol activo, en donde el dominio de la computadora se dé por tener la capacidad de utilizar la computadora como un medio expresivo (Resnick, 2017).

Sería interesante abrir una nueva línea de investigación que pueda profundizar en los interrogantes que surgieron luego de realizar esta experiencia. Podría resultar interesante implementar una experiencia a mayor escala, realizando intervenciones similares a las realizadas, pero con diferentes secuencias didácticas, todas apuntando al desarrollo del PC. Podríamos así comparar la eficacia de distintas estrategias, para determinar la más adecuada.

Resultaría de particular interés estudiar en profundidad, analizar, y descubrir las causas que producen desigualdad entre mujeres y varones en este campo. Sin dudas ameritaría otra investigación, con otros objetivos específicos, y seguramente implementando otra metodología, relacionando otras variables, etc. Sería inconducente pretender que se desprendan más conclusiones sobre la brecha de género del presente trabajo, cuyos objetivos fueron otros.

Una continuidad de esta línea de investigación podría ampliar los horizontes de los “cómo”, “quiénes”, “cuándo” y “por qué” se debe trabajar contenidos relacionados al PC en la educación formal, para todos los estudiantes.

## **Bibliografía**

- Aho, A. V. (2012). Computation and Computational Thinking. *The Computer Journal* 55 (7), 832–835. <https://doi.org/10.1093/comjnl/bxs074>
- Freire, P. (1965). *La Educación como práctica de la libertad*. México. Siglo XXI.
- Grover, S. y Pea, R. (2013). Computational thinking in k-12. A review of the state of the field. *Educational Researcher*, 42(1), pp.38-43, doi: 10.3102/0013189X12463051
- Papert, S. (1999)- *Logo Philosophy and Implementation*.
- Polanco Padrón, N., Ferrer Planchart, S., & Fernández Reina, M. (2021). Aproximación a

una definición de pensamiento computacional. RIED (Revista Iberoamericana De Educación a Distancia), 24(1), 55-76. <https://doi.org/10.5944/ried.24.1.27419>

Resnick, M. (2012). Reviving Papert's Dream. *Educational Technology*, 52(4), 42-46.

Resnick, M. (2017). *Lifelong Kindergarten: Cultivando la creatividad a través de proyectos, pasión, pares y juego*. MIT Media Lab, MIT Press

Román, M. (2016). *Codigoalfabetización y pensamiento computacional en Educación Primaria y Secundaria: validación de un instrumento y evaluación de programas*. [Tesis Doctoral], Universidad Nacional de Educación a Distancia.

Wing, J. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, 366(1881), 3717-3725. doi: <http://dx.doi.org/10.1098/rsta.2008.0118>

Soe, L., & Yakura, E. K. (2008). What's wrong with the pipeline? Assumptions about gender and culture in IT work. *Women's Studies*, <https://doi.org/10.1080/00497870801917028>

## Los Objetos de Aprendizaje: Insumos para la Tutoría en Programación

Marcela F. López, Eduardo F. Fernández, Paola del Olmo  
{marcelaflopez, eduardo.fernandez.unsa, paoladelolmo2}@gmail.com

Universidad Nacional de Salta

### Resumen

El presente trabajo se contextualiza en la cátedra de Programación Básica, formada por las asignaturas Elementos de Programación y Programación de las carreras Licenciatura en Análisis de Sistemas y Tecnicatura Universitaria en Programación, de la Universidad Nacional de Salta. En el mismo se presenta el modelo que se decidió seguir para diseñar e implementar Objetos de Aprendizaje (OA). Este modelo procura disminuir la carga asociada a las dimensiones tecnológicas y de interacción, priorizando el aspecto pedagógico, a fin de facilitar a los docentes la generación de materiales para ser utilizados en espacios virtuales, en contexto de clases invertidas, en modalidad de enseñanza mixta. Los OA son el soporte de un modelo de Tutoría, que se pretende automatizar a fin de colaborar con el docente en el seguimiento sostenido del proceso de aprendizaje y brindar la posibilidad de una enseñanza personalizada, atendiendo a estilos estratégicos de aprendizaje de cada estudiante, en consideración a la masividad de la cátedra, altamente numerosa, conformada por materias iniciales de las carreras.

**Palabras claves:** Objetos de Aprendizaje, Estilos de aprendizaje, Seguimiento, Tutoría.

### Introducción

El contexto en el que se basa el presente trabajo está situado en la cátedra de Programación Básica, formada por las asignaturas Elementos de Programación y Programación de las carreras Licenciatura en Análisis de Sistemas y Tecnicatura Universitaria en Programación, de la Universidad Nacional de Salta. Una característica notoria de los últimos años es el incremento en la cantidad de alumnos que se inscriben para cursar estas asignaturas, siendo en el año 2023 de 1450 estudiantes, la cual se confronta con la baja tasa de regularización que históricamente se registra al finalizar el primer cuatrimestre, de alrededor del 20%. Esta situación lleva a analizar alguno de los factores que inciden en este comportamiento. Uno de ellos es la necesidad de desarrollar pensamiento abstracto en los estudiantes de esta disciplina.

Basados en las expresiones de Carrillo y Valencia [2], en su cita sobre Dijkstra afirma; “La computación es un disciplina que tiene como objetivo modelar y representar en un computador

sistemas de conceptos u objetos con el fin de resolver problemas, realizar predicciones mediante la simulación de sus interacciones o efectuar cálculos y raciocinios con ellos; por consiguiente, el proceso de aprender a programar es complejo, la computación es una disciplina que obliga al estudiante a desarrollar jerarquías conceptuales que son mucho más profundas que en otras áreas del saber” Dijkstra,1989 (citado en Carrillo, 2003).

Entendemos que el aprendizaje de la Programación es un proceso iterativo e incremental que demanda de un alto nivel abstracción, requiere la aplicación de técnicas y heurísticas y la construcción de un estilo de programación fundado en buenas prácticas y en criterios de eficiencia algorítmica. Como docentes, observamos diferentes problemáticas en el aprendizaje de los alumnos iniciales de Programación, que derivan en el bajo rendimiento evaluativo. Frente a esta situación, se diseñó e implementó, entre otros dispositivos, el Taller de Tutoría. Este Taller plantea un seguimiento sostenido del proceso de aprendizaje que cada estudiante realiza. A su vez en el mismo, los tutores desarrollan trayectos específicos para que cada estudiante alcance los objetivos propuestos. A partir de este trabajo de tutoría se pudieron identificar y clasificar los temas de mayor dificultad para los estudiantes, respecto a su abordaje y comprensión. Estos temas son los que orientan la construcción de Objetos de Aprendizaje (OA), que, en el contexto de la cátedra, se propone diseñar e implementar. El tutor, con estos OA, podrá diseñar trayectos remediales atendiendo a los estilos de aprendizaje de cada estudiante. En este sentido, se diseñan los OA sobre la base de las fortalezas y debilidades del proceso de aprendizaje, evidenciadas por los estudiantes.

En este artículo se presenta la forma de trabajo que diseñó el equipo, para definir cada OA, disminuyendo la carga asociada a las dimensiones tecnológicas y de interacción y priorizando el aspecto pedagógico, con la finalidad de que permita a los docentes generar de forma sencilla materiales para ser utilizados en espacios virtuales.

### **Definición de OA y sus características**

Existen distintas definiciones de OA, pero en general son poco precisas.

La IEEE [6] define un OA como “Cualquier entidad digital o no digital, que pueda ser utilizada o referenciada durante el aprendizaje basado en el ordenador”. Mientras que Wiley lo define como: “Cualquier recurso digital que puede ser utilizado para soportar el aprendizaje”.

El equipo de investigación entiende y define que un OA es un bloque conceptual y autocontenido, que puede combinarse de diferentes formas para construir colecciones de OA. Tanto los OA como las colecciones pueden ser implementados utilizando diferentes recursos como lecciones, cursos, módulos, entre otros.

Por lo tanto, se puede decir que un OA es un recurso didáctico interactivo, en formato digital, desarrollado con el propósito de ser utilizado en diversos contextos educativos y responde a una necesidad instruccional, siendo ésta su principal característica. El objetivo primordial de un OA es el de habilitar y orientar el aprendizaje.

Entendiendo que los OA presentan cualidades que promueven su independencia respecto a los recursos tecnológicos sobre los que se implementan y que la cátedra sostiene la enseñanza en una configuración donde conviven la modalidad presencial y los recursos de aprendizaje remoto, a partir del uso de la plataforma educativa Moodle, es que estos OA son implementados atendiendo a las siguientes características:

- Reutilización: se debe poder dividir el contenido en pequeñas unidades de instrucción, para poder ser utilizadas en diferentes cursos.
- Interoperabilidad: las unidades de instrucción deben poder integrarse independientemente de su desarrollador o plataforma para la que se diseñó.
- Durabilidad: las unidades de instrucción deben seguir siendo reutilizables, aunque cambien las tecnologías para su presentación y distribución.
- Accesibilidad: las unidades de instrucción deben estar disponibles en cualquier parte y en todo momento.

Para que un OA pueda cumplir con su función de forma aislada e independiente de la plataforma educativa en la que sea implementada, se necesita que cumpla con ciertas características básicas, como lo son:

- Autocontenido: a fin de ser independiente debe tener una estructura adecuada como: Presentación; Objetivos; El contenido formativo; Evaluación y Conclusión.
- Breve: debe ser breve dada la limitación del medio on-line, por el tiempo medio de la sesión de un usuario.
- Independiente del contexto: no debe precisar de otro contexto que él mismo, no debe hacer referencia a otros OA.
- Etiquetado: para facilitar su identificación y búsqueda debe tener claramente definidas sus etiquetas descriptivas.
- Interoperabilidad: se deben diseñar de acuerdo a un estándar que permita la independencia de la plataforma.

Desde otro punto de vista, se debe tener presente que, para propiciar el aprendizaje, se requiere considerar aspectos muy particulares en la concepción de un OA, como lo son las dimensiones *pedagógicas, tecnológicas y la interacción*.

- Pedagógicas: un OA tiene una intención educativa precisa. Permite establecer secuencias lógicas para la efectividad del proceso de enseñanza, que se espera favorezca el aprendizaje. También promueve la construcción y difusión del conocimiento. Las características más relevantes a trabajar en su concepción son:

- Que deben estar orientados a una diversidad de estilos de aprendizaje;
- Que los contenidos deben ser relevantes;
- Que debe promover el cumplimiento de los objetivos de aprendizaje;
- Que se debe determinar secuenciación de presentación de contenidos, actividades y evaluación, acorde a los objetivos.

- Tecnológicas: los OA son recursos digitales, que abarcan diferentes aspectos tecnológicos que son tratados en general desde el área de la Ingeniería de Software. Cuando se utilizan desde una LMS se deben adecuar estándares que permitan su implementación en la misma.

- Interacción: es importante que los OA generen motivación e interés a los aprendices, para propiciar el trabajo con los mismos y así favorecer el proceso de aprendizaje.

Como se puede apreciar, el diseño e implementación de un OA requiere que los mismos cumplan con determinadas características y que además se ajusten a algún estándar para que pueda ser soportados en una plataforma educativa.

### **Diseño de un OA**

Actualmente el equipo se encuentra trabajando en el diseño de las unidades instruccionales, que en adelante se mencionarán como **Contenido**. Este Contenido de aprendizaje se organiza mediante una estructura desde la que se presentan conceptos, ejemplos, ejercicios, evaluaciones, etc. que tiene por objetivo fomentar el aprendizaje.

Para el diseño de este Contenido se tiene en cuenta las dimensiones pedagógicas, tecnológicas y la interacción. En este caso se debe considerar que los contenidos van a ser implementados en una plataforma Moodle. Esta contextualización permite al equipo generar un OA sin atender todas las características de la dimensión tecnológica y de la interacción, centrando la atención en el diseño e implementación de los contenidos seleccionados, es decir,

en la dimensión pedagógica. Posteriormente se tiene previsto que los OA alcancen todas las características necesarias, una vez que se hayan probado desde la plataforma de aprendizaje.

El desarrollo de OA es un trabajo interdisciplinario en donde intervienen expertos en el campo de conocimiento, en didáctica como así también en tecnología.

Para la producción de un OA se distinguen dos etapas principales:

- Pedagógica: en la que se definen: objetivos educativos, contenidos, actividades de aprendizaje y/o evaluación.
- Tecnológica: en la cual se adecuan los contenidos, actividades e interacción, para su presentación en la WEB y se generan los metadatos para almacenarlo en un repositorio y publicarlo en una plataforma de aprendizaje. En el caso particular de la asignatura Programación, se tomó la decisión de no definir los metadatos, ya que, en esta etapa, sólo se implementa en el Aula Virtual de la cátedra
- Desde un punto de vista pedagógico, un OA constituye un elemento que sostiene la transposición didáctica y se diseña considerando los estilos de aprendizaje de los estudiantes, con el objetivo de facilitar sostenidamente el proceso de aprendizaje.

Como recurso pedagógico, un OA debe atender a distintos tipos de usuarios y considerar las características individuales de cada uno de ellos, flexibilizando las estrategias apropiadas a los estilos de aprendizaje. Desde esta perspectiva, el diseño de un OA resulta un desafío para el docente, quien debe elegir el contenido, diseñar el trayecto de enseñanza, establecer la forma en que se presenta y se interactúa con el contenido, fundamentándose, como ya se mencionó, en las características de los usuarios o destinatarios.

Desde el punto de vista de la tecnología, el OA constituye el soporte para que los contenidos, las actividades y las evaluaciones presenten características de usabilidad, accesibilidad, interoperabilidad, entre otras. Es decir, esta etapa se relaciona directamente con la construcción del OA que será implementado atendiendo a los estándares establecidos para un objeto de aprendizaje que será publicado en WEB. En general, esta etapa es compleja ya que requiere de especialistas tecnológicos para su implementación, quienes suele ser un recurso humano escaso en muchas instituciones educativas.

### **La creación de OA en Programación**

A partir de la implementación del Taller de Tutoría aplicado a la asignatura Programación, se desarrolla una tabla que codifica las dificultades frecuentemente observadas en los estudiantes y que representan los principales obstáculos para el aprendizaje. Luego de indagar y codificar

los obstáculos, se diseñan las actividades que deben desarrollar los estudiantes. Estas actividades generan diferentes trayectos de aprendizaje. Los trayectos son analizados al interior de la cátedra para determinar cuáles favorecen al proceso de aprendizaje y cuáles deben ser rediseñados.

La tabla de códigos de errores generada a partir de los problemas detectados, son para este proyecto, un insumo fundamental para diseñar e implementar los OA que requiere la cátedra, a fin de dar mayor dinamismo a la tutoría.

Para diseñar e implementar los OA con los que van a interactuar los estudiantes, el tutor, según Pozo [9], debe realizar “Una selección y secuenciación del currículo para los estudiantes, que requiere del uso de mecanismos de planificación bastante sofisticados, que deben tener en cuenta la teoría de tutorización, de acuerdo a las necesidades del aprendiz”. En este sentido, Ausubel [1] y otros afirman que “(se) Debe fomentar el objetivo de que el estudiante, le encuentre significado a sus aprendizajes, y que supere sus dificultades, incorporando lo nuevo de un modo significativo y permanente”.

En el trabajo de Merrill [8], se plantea la posibilidad de transmisión del conocimiento a través de micro elementos de instrucción basados en una única idea y para su tratamiento, divide la información en cuatro tipos: conceptos, hechos, procedimientos y principios y su forma de presentación en: reglas, ejemplos, recordatorio y actividades.

Tomando como referencia estos autores y atendiendo a la dimensión pedagógica, los Contenidos de Programación son organizados, en un primer momento, en contenidos teóricos y contenidos prácticos. Posteriormente, el contenido teórico se define con la siguiente secuencia: Introducción, Desarrollo teórico, Ejemplo, Evaluación y Cierre. El contenido para la práctica se organiza en: Introducción, Presentación de un caso, Explicación del caso, resolución, Evaluación y Cierre.

Los OA que se diseñan en el contexto de la cátedra, se basan también en el modelo de estilos cognitivos de aprendizaje. García [5], indica que los estilos de aprendizaje son los rasgos cognitivos, afectivos, fisiológicos, de preferencias por el uso de los sentidos, ambiente, cultura, comportamiento, comodidad, desarrollo y personalidad que sirven como indicadores relativamente estables, de cómo las personas perciben, interrelacionan y responden a sus ambientes de aprendizaje y a sus propios métodos o estrategias en su forma de aprender. Los procesos cognitivos básicos mencionados por De Sánchez [3], son los siguientes: observación, comparación, relación, clasificación simple, ordenamiento, clasificación jerárquica, análisis, síntesis y evaluación. Estos procesos psicológicos están estrechamente relacionados al objetivo de aprendizaje de los estudiantes. Por su parte, Felder y Silverman [3] proponen una

clasificación de los estilos de aprendizaje, de la siguiente forma: Visuales – Verbales, Inductivos – Deductivos, Secuenciales – Globales, Activos – Reflexivos.

Bajo la mirada de la dimensión tecnológica, los Contenidos presentan el desafío de establecer cómo se van a representar digitalmente. Atendiendo a los estudios realizado por Edgar Dale [7] investigador educacional, respecto a la efectividad de los distintos medios en el proceso de aprendizaje, se considera a continuación, el Cono de aprendizaje, resultado de su estudio. Este Cono refleja la relación entre el nivel de utilización de los medios y las tasas de retención de lo aprendido. El mismo se muestra en la figura 1.



Figura 1. relación entre el nivel de utilización de los medios con las tasas de retención de lo aprendido

Considerando que la implementación de los Contenidos se realiza en una plataforma educativa, se establece, para su presentación, la utilización de formatos digitales, Textos y Multimediales.

En función a lo expuesto anteriormente, el equipo formaliza el proceso para generar un OA, el cual se concreta a partir del diseño de dos fichas, las que constituyen el Modelo de los OA. Las mismas se muestran en la Figura 2.

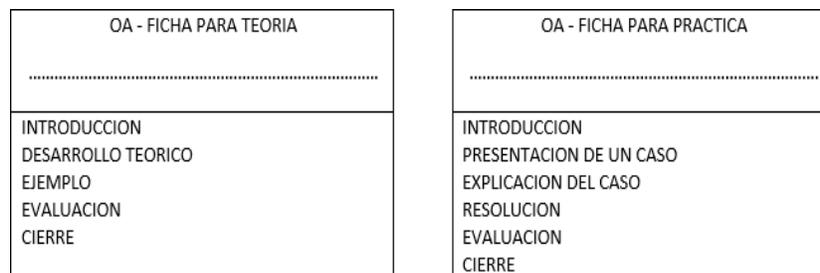


Figura 2. Modelo de las Fichas para OA

El Modelo también contiene una tabla que permite realizar un guion sobre cómo se debe implementar el contenido a desarrollar. Se presenta en la Figura 3.

Pantalla	Tema-Subtema	Texto de pantalla	Indicaciones multimediales				Observaciones
			Sonidos-locuciones	Imágenes - foto	Animaciones	Links	

Figura 3. Modelo para el guión un OA

Se espera que este Modelo permita la planificación efectiva y la construcción más adecuada para los temas seleccionados, al igual que su utilización, reutilización y actualización, apoyando y facilitando la implementación de los mismos.

Igualmente, estas fichas permiten seleccionar los contenidos por su coherencia con los objetivos de aprendizaje. De esta manera, el docente cuenta con recursos para ejercer un mejor control sobre la información disponible.

El docente que construye sus propios OA también se beneficia, ya que, una vez construidos estos OA pueden ser mejorados, planificando su modificación a partir de la documentación existen (fichas y tablas), potenciando así el trabajo previamente realizado.

Una vez diseñado los OA, son implementados utilizando recursos de la plataforma Moodle. En una primera instancia se utiliza lecciones y cuestionarios para el desarrollo del Contenido, considerando que algunas actividades pueden ser evaluadas mediante el recurso tarea. La decisión de utilizar estos recursos para la implementación, se fundamentan en lograr una construcción más sencilla y rápida que se ajusta mejor a la tecnología que se usa en el contexto de la cátedra.

### Implementación de OAs

Considerado el Modelo presentado en el apartado anterior y a modo de ejemplo, se muestra, en primer orden, la ficha del OA de teoría de Búsqueda Binaria y posteriormente el guion multimedial para su elaboración.

<b>OA - FICHA PARA TEORÍA Método de Búsqueda Binaria</b>
<p><b>INTRODUCCIÓN:</b></p> <p>Una de las tareas más importante cuando trabajamos con una lista de elementos ordenados, es establecer si un elemento determinado se encuentra en la lista. Por ejemplo, si tenemos una lista de libros de una librería, podríamos determinar si un libro dado se encuentra disponible. Como resultado de la búsqueda se obtendrá una indicación, “Si está” o “No está” en la lista. En caso de estar, también se podría retornar su ubicación y datos que estén asociados al libro, como su precio, cantidad de ejemplares, etc.</p> <p>La idea es determinar el punto medio de la lista. Esta acción partirá la lista, en dos sublistas, la</p>

sublista a la izquierda del punto medio y la sublista a la derecha. El proceso continúa determinando si el elemento buscado es igual al que se encuentra en el punto medio, si es así, se termina la búsqueda porque encontramos el elemento y su posición en la lista. Caso contrario, debemos determinar si el elemento buscado puede encontrarse en la sublista de la izquierda o en la sublista de la derecha. Esta decisión permite dejar de inspeccionar la sublista en la que sabemos que no tiene chance de encontrarse el elemento, ya que la lista está ordenada. Esta sublista es ahora nuestra lista en la que trabajamos.

A continuación, volvemos a determinar el punto medio de esta nueva lista y repetimos el mismo proceso que se aplicó en la lista original.

Este proceso continúa hasta que encontramos el elemento buscado o hasta que la lista es una lista vacía.

### **DESARROLLO TEÓRICO:**

Primero debemos definir el elemento a buscar, al cual llamaremos BUS.

A continuación, debemos inicializar los tres índices, con los cuales realizaremos el recorrido por el vector A.

- Para marcar el primer elemento del vector usaremos INI,
- Para marcar el último elemento del vector usaremos FIN
- Para marcar el elemento medio del vector usaremos MED, cuyo valor es el resultado de la división entera de  $(INI + FIN) / 2$ .

.....

Atendiendo estos requerimientos el módulo finalmente implementado en C, se define de la siguiente forma:

.....

### **EJEMPLO**

Si se tiene el siguiente vector A (2, 3, 5, 9, 16), la cantidad de elementos del vector es  $N=5$  y el elemento buscado es 9. Se desea determinar el elemento buscado está en el vector y en qué posición o si no se encuentra en el vector A, en cuyo caso se pide informar que no se encuentra.

.....

### **CIERRE**

Recordemos este método tiene tres secciones:

.....

Observemos que la condición  $A[MED]=BUS$  no es adecuada, porque cuando no se encuentra el elemento, se cumple la condición  $INI>FIN$  y el valor de MED puede no estar en el rango  $[1, N]$ , dando error la evaluación  $A[MED]=BUS$ .

También debemos recordar que este método necesita que el vector esté ordenado.

### **EVALUACION**

Se tiene una lista A de N números naturales, que se ingresa ordenada de menor a mayor y un número natural llamado BUS. Se pide determinar si el elemento se encuentra en la lista A.

Caso1: Si  $N=7$ ,  $A= (1, 3, 4, 7, 9, 16, 23)$  y  $Bus= 23$ . Se pide realizar a las siguientes consignas:

a) Indica en cual iteración de la prueba de escritorio, MED toma el valor 7.

1°		2°		3°		Nunca	
----	--	----	--	----	--	-------	--

Tabla 1: Ficha ilustrativa de OA - Búsqueda Binaria

Pantalla	Tema-Subtema	Texto de pantalla	Indicaciones multimediales				Observaciones
			Sonidos locución	Imágenes - foto	Animaciones	Links	
INT-01	Introducción	Texto 01		Img INT01 –Libros en una librería, persona buscando. Img INT02 –Persona encontrando libro	Los textos aparecerán gradualmente, al igual las imágenes		
INT-02	Introducción	Texto 02		Img INT03 – Un vector ordenado de Img INT04 – El vector se ve dividido en dos partes. Img INT05 – El elemento buscado, un número que este en el vector.			Al presentar el texto que este se resalte en el momento que se presenta la imagen, revisar el tiempo que lleva la lectura para activar la imagen.

**Introducción:**

Texto 01 ¿Por qué buscar? (título)

Una de las tareas más importante cuando trabajamos con una lista de elementos ordenados, es encontrar si un elemento determinado está en la lista. Por ejemplo, si tenemos una lista de libros de una librería, podríamos determinar si un libro dado se encuentra disponible, la devolución solamente podría ser “Si está” o podríamos devolver su ubicación y datos que están asociados al libro, como su precio, cantidad de ejemplares, etc. ....

Texto 02 ¿Cómo buscar? (Título)

La idea es determinar el punto medio de la lista. Esta acción partirá la lista, en dos sublistas, la sublista a la izquierda del punto medio y la sublista a la derecha. El proceso continúa determinando si el elemento buscado es igual al que se encuentra en el punto medio, si es así, se termina la búsqueda porque encontramos el elemento y su posición en la lista. Caso contrario, debemos determinar si el elemento buscado puede encontrarse en la sublista de la izquierda o en la sublista de la derecha. Esta decisión permite dejar de inspeccionar la sublista en la que sabemos que no tiene chance de encontrarse el elemento, ya que la lista esta ordenada. Esta sublista es ahora nuestra lista en la que trabajamos.

Tabla 2: Para el guionado de OA - Búsqueda Binaria

### Algunos resultados

Los OA facilitan el modelo de clase invertida ya que los estudiantes interactúan con ellos a fin de apropiarse de conceptos y aplicaciones que luego son reafirmados en clases presenciales. Como ejemplo de esta situación citamos la experiencia de haber utilizado estos OA en el espacio del Taller de Programación, donde los estudiantes, luego de interactuar con el OA de búsqueda binaria, debieron desarrollar un ejercicio de aplicación, que presentaron utilizando el recurso “Tarea”, disponible en el aula virtual. Esta actividad permitió al docente evaluar las resoluciones y clasificar las dificultades, aplicando para ello, codificación de los errores

Alumno	Actividad con OA	NIVEL 4			
		CodSeg1	CodSeg2	CodSeg3	CodSeg4
Alumno 1	5	1	4	18	20
Alumno 2	5	1	4	18	20
Alumno 3	10	38			
Alumno 4	0	38			
Alumno 5	5	1	4	18	20
Alumno 6	5	1	4	18	20
Alumno 7	0	1	4	18	20
Alumno 8	5	38			
Alumno 9	5	1	4	18	20
Alumno 10	5	38			
Alumno 11	0	38			
Alumno 12	10	1	4	18	20
Alumno 13	5	1	4	18	20
Alumno 14	3	1	4	18	20
Alumno 15	5	38			
Alumno 16	5	1	4	18	20
Alumno 17	5	1	4	18	20

Tabla 3: Tabla de seguimiento de alumnos del taller

A partir del análisis realizado, se diseñaron actividades remediales, atendiendo a la siguiente planificación:

Código	Descripción del problema	Acción Remedial	Comunicación
1	Interpretar la consigna - Grado malo	Para resolver este problema, se solicita en cada nivel un informe donde se analiza el problema y se debe determinar los datos de E/S (dato, identificador y TD en C, Tareas a realizar par resolver el problema. Se trabaja con los 3 últimos ejercicios del TP de cada nivel, asignando uno de ellos a cada estudiante	La comunicación con los estudiantes se realizó a través del Foro de consultas de cada Nivel, durante las clases por videoconferencia, chat privado de Telegram y encuentros presenciales.
4	Determinar las tareas que solucionan el problema – Grado regular		
5	Determinar las tareas que solucionan el problema – Grado malo		
6	Determinar las tareas que solucionan el problema - Grado muy malo		
18	Diseñar módulos funcionales	Se solicita revisar los conceptos asociados a la definición e implementación de módulos, usando los OA que se encuentran en el AV a través de las Librerías del Nivel 2, 3 y 4, leer también los apuntes para ampliar, para luego realizar las correcciones adecuadas. Esto se trabaja sobre situaciones problemáticas de cada nivel.	
19	Implementar Correctamente módulos		
20	Determinar Parámetros (cantidad y tipos de datos asociados)		
38	Cumplir con Actividades propuestas	Se solicita al alumno, cumplir con las tareas remediales propuestas en tiempo y forma.	

Tabla 4: Tabla de planificación de actividades remediales

A partir de estas tareas, se diseñaron nuevos OA con situaciones problemáticas a resolver que, según los errores detectados, realizaron los estudiantes discrecionalmente y que presentaron en un foro de discusión. Esta actividad permitió revisar conceptos y estrategias de aplicación en un espacio presencial.

### En conclusión

La incorporación de Objetos de Aprendizaje para la conformación de trayectos educativos favorece el proceso de enseñanza en tanto que permite contrarrestar los efectos de la masividad del alumnado, al facilitar recursos acordes a los estilos de aprendizaje de cada estudiante y diseñar trayectos personalizados en función del proceso de tutorización reali-

zados. Siguiendo esta línea, también se plantean nuevos desafíos hacia el interior de la cátedra ya que, el diseño y la implementación de estos OA hacen rever, de manera permanente, la práctica docente.

Estos OA constituyen una base para el aprendizaje que está en constante crecimiento y ajuste. Nuevos OA son requeridos para diseñar trayectos cada vez más pertinentes al perfil de cada estudiante y retroalimentan el diseño de trayectos más específicos en atención a los errores o imprecisiones que manifiestan los estudiantes.

Las líneas de acción en las que se deben trabajar son:

- aumentar la cantidad de OA en función de los requerimientos que se registran, lo que propiciará el armado de los trayectos de aprendizaje.
- mejorar el proceso de análisis de las producciones de los estudiantes, mediante la aplicación de los códigos de errores elaborados, procurando su automatización, en la medida de lo posible.
- evolucionar los OA buscando su independización de la plataforma de aprendizaje para que adquieran características más generales de accesibilidad, interoperabilidad y reutilización, entre otras.

## Referencias

1. Ausubel, D.; Novak, J. y Hanessian, H. (1983) Psicología educativa: un punto de vista cognitivo. México: Trillas.
2. Carrillo, M.; Valencia, H. (2003). Las Nuevas Tecnologías de Información y Comunicación en los procesos de enseñanza y aprendizaje: representación del conocimiento y educación virtual. TED: Tecné, Episteme y Didaxis. 10.17227/ted.num13- 5583.
3. de Sánchez, M. (2008). Manual para el curso de Factores de Desarrollo Intelectual, ITESM, México.
4. Felder, R. y Silverman, L. (1988). Learning and Teaching Styles in Engineering Education. *Engr. Education*, 78(7), pp 674-681.  
Link: <http://www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/LS-1988.pdf>.
5. García, J. (2010). Estilos de Aprendizaje. Web de José Luis García. Link: <http://www.jlgcue.es>.
6. Learning Technology Standards Committee. IEEE. (2002). Standard for Learning Object

Metadata. IEEE Standard 1484.12.1, Institute of Electrical and Electronics Engineers, New York.  
link: [www.ieeeltsc.org](http://www.ieeeltsc.org).

7. Madrigal Alfaro, Z. (2010). Propuesta de arquitectura para incluir a las simulaciones en los sistemas e-learning: una extensión al modelo SCORM. Tesis Doctoral. Universidad Politécnica de Madrid. Link: <http://www.oa.upm.es/3321>.

8. Merrill, M. D. (1983). Component Display Theory. In C. Reigeluth (ed.), *Instructional Design Theories and Models*. Hillsdale, NJ: Erlbaum Associates.

9. Pozo Municio, I. (1999). *Aprendices y Maestros*. Alianza.

10. Wiley D (2000). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy". *The Instructional Use of Learning Objects*.

Link: <http://reusability.org/read/chapters/wiley.doc>, accedido en 2007-03. (en inglés)

# JADiCC

Jornadas Argentinas de Didáctica  
de las Ciencias de la Computación

1 y 2 de diciembre de 2023  
NEUQUÉN  
PATAGONIA - ARGENTINA

---

*Relatos de experiencias en el aula  
y formación docente*

**Vamos a la Escuela Primaria**  
**Contactos tempranos con las Ciencias de la Computación**

Rodrigo Piersigilli, Ian Franco Acosta, Susana Parra, Jorge Rodríguez  
{rodrigo.piersigilli, ian.acosta}@est.fi.uncoma.edu.ar,  
{susana.parra, j.rodrig}@fi.uncoma.edu.ar

Facultad de Informática  
Universidad Nacional del Comahue

**Resumen:**

Considerando la era digital en la que vivimos, el conocimiento y la comprensión de los fundamentos de la informática se vuelven cada vez más necesarios en todos los aspectos de la vida. La enseñanza de las ciencias de la computación desde la escuela primaria es fundamental, fomenta el pensamiento computacional, promueve la alfabetización digital, la creatividad y la resolución de problemas.

En Argentina se desarrollan políticas públicas e iniciativas desplegadas por las Universidades Nacionales y Program.ar que promueven la incorporación de las ciencias de la computación en la educación primaria. Sin embargo, la presencia consistente y rigurosa de contenidos computacionales en las propuestas de enseñanza para la escuela primaria es un proceso incipiente en la mayoría de las jurisdicciones.

Vamos a la Primaria es una iniciativa desplegada en colaboración entre la Facultad de Informática de la Universidad Nacional del Comahue y un conjunto de escuelas primarias de Río Negro y Neuquén que busca establecer contactos tempranos entre las infancias y el conocimiento computacional. Son parte de Vamos a la Primaria cinco escuelas de la región, que incluyen a más de 150 niños y niñas.

En el marco de la iniciativa, un equipo de facilitación de la universidad visita a escuelas primarias de la región acercando experiencias educativas. Niños y niñas se contactan con conocimientos fundamentales sobre Ingeniería de Software, Algoritmos y Programación y Software Libre. Desarrollan aplicaciones móviles, construyen piezas de software para controlar robots y programan drones en un ambiente de juego en el que confluyen el entretenimiento y el aprendizaje.

**Palabras clave:** Ciencias de la Computación, Aprendizaje temprano, Escuela primaria, Universidad, Grupos.

**Introducción**

En un mundo cada vez más orientado hacia la tecnología y la información, la adquisición en

edades tempranas de habilidades y construir conocimientos en el área de las Ciencias de la Computación permite comprender mejor el mundo y mejorar las posibilidades de participación ciudadana (K-12 Computer Science Framework Steering Committee, 2016).

Los esfuerzos en Argentina para introducir las ciencias de la computación en la educación primaria responden a la creciente importancia de la tecnología en la sociedad actual (Llambi, 2023). Sin embargo, a pesar de los esfuerzos y el equipamiento tecnológico proporcionado a las escuelas, la presencia de la ciencia de la computación en la educación primaria en Argentina sigue siendo insuficiente (Quintana, 2020).

Es esencial abordar esta cuestión para garantizar que los estudiantes estén preparados para el futuro en un mundo cada vez más tecnológico. La brecha entre la disponibilidad de recursos tecnológicos y su integración efectiva en el currículo escolar es una cuestión que requiere atención y acción continua (Quintana, 2020).

La universidad pública, a través de proyectos de extensión e investigación, renueva su compromiso con la promoción de la educación en Ciencias de la Computación, diseñando actividades y proporcionando acceso a recursos tecnológicos y materiales educativos actualizados, lo que complementa los esfuerzos a nivel de políticas públicas .

En este trabajo se presenta un modelo didáctico disciplinar que busca ofrecer un primer contacto con conceptos fundamentales sobre Ciencias de la Computación. El modelo propuesto se puso en práctica en el territorio de la escuela primaria con el objetivo de recuperar aprendizajes que permitan realizar ajustes progresivos a partir de la consideración de comentarios, sugerencias y revisiones realizadas sobre el trabajo de campo.

El modelo se compone de experiencias educativas breves que fomentan la participación activa de los estudiantes a través del trabajo colaborativo, brindándoles la oportunidad de aprender y crear juntos. En estos talleres, mediante el aprendizaje experiencial los estudiantes establecen contactos tempranos con principios fundamentales de la Ingeniería de Software, prácticas y conceptos del área Algoritmos y Programación y se hace hincapié en la importancia del Software Libre en términos de soberanía tecnológica.

Por otra parte, la población estudiantil de las escuelas primarias puede no estar suficientemente familiarizada con la universidad como parte de la definición de sus proyectos de vida. En este sentido, la constitución de redes de colaboración entre la Universidad Pública y las Escuelas Primarias ayuda a destacar la relevancia y el potencial de la educación superior para contribuir al desarrollo social.

Esta propuesta educativa, expresa la necesidad de que la Universidad Pública participe

activamente en los esfuerzos destinados a fomentar el aprendizaje temprano de Ciencias de la Computación. La Universidad Nacional del Comahue, como institución de conocimiento y formación académica, despliega iniciativas en el contexto de la investigación y la extensión que juega un papel importante en la creación de bases consistentes para promover la enseñanza de las Ciencias de la Computación desde edades tempranas. Las actividades diseñadas buscan ofrecer una primera experiencia positiva con las Ciencias de la Computación, brindando una introducción al mundo de la programación y el desarrollo del pensamiento computacional.

### **Modelo didáctico disciplinar propuesto**

En esta sección se presenta un modelo didáctico disciplinar destinado a ofrecer contactos altamente significativos entre las infancias y las Ciencias de la Computación. La propuesta se estructura metodológicamente a partir de la convergencia de los conceptos de aprendizaje colaborativo como organizador de la interacción social, el abordaje por proyectos como estructurante de los procesos de aprendizaje y del juego como dispositivo de enseñanza (Barkley, 2014; Clerici 2021; Lee, 2014).

En esta instancia el modelo propuesto se compone de tres experiencias breves que asumen el formato de taller. En este ámbito las infancias, organizadas en pequeños grupos juegan a desarrollar pequeñas piezas de software que los exponen a manipular con conceptos abstractos sobre Ciencias de la Computación para programar robots, construir aplicaciones móviles, volar drones.

Se busca que cada grupo participe de dos o tres experiencias como mecanismo que favorece la transposición de los aprendizajes construidos en un escenario educativo a otro contexto. De esta manera los estudiantes mejoran las posibilidades de construir generalizaciones y avanzar sobre la elaboración de conceptos abstractos susceptibles a ser instalados en contextos diversos (Morris, 2020; Kolb, 2005).

En la instancia presentada en este trabajo el modelo se compone de tres talleres que involucran el trabajo sobre:

- dominios diversos: desarrollo de aplicaciones, programación de drones y programación de robots.
- múltiples modos de programación: basada en texto y basada en bloques.

En este contexto los talleres ofrecen experiencias concretas sobre los que se despliegan procesos de reflexión que colaboran en la formulación de conceptos abstractos para luego aplicar esas generalizaciones en otro dominio. En esta dirección se busca que las infancias participen de un proceso de aprendizaje y desarrollo de habilidades a través de experiencias

concretas (Morris, 2020; Kolb, 2005).

### **Taller de desarrollo de aplicaciones móviles**

El grupo general trabaja de manera colaborativa y creativa para desarrollar una aplicación inspirada en el popular juego Preguntados. El proceso comienza con la especificación de requerimientos, donde los participantes aportan ideas y plantean las funcionalidades deseadas. A medida que avanzamos, nos enfocamos en el desarrollo visual de la aplicación, creando una interfaz atractiva y amigable para el usuario (Patton, 2019).

### **Taller dron lab**

Se comienza por familiarizar a los participantes con las funcionalidades del dron y el entorno de programación a utilizar. El grupo general se organiza en subgrupos de cuatro integrantes y se les asignan estaciones de trabajo separadas por unos metros. Cada subgrupo debe seguir instrucciones secuenciales para avanzar el dron hasta la estación siguiente y luego pasan el control al próximo grupo. El objetivo es completar el circuito con la menor cantidad de intentos posibles, promoviendo el trabajo en equipo y la coordinación (Bai, 2020).

### **Taller programando a Frankestito**

Se exponen las funcionalidades del robot denominado Frankestito y se trabaja en el desarrollo de un producto de software que permita al robot cumplir objetivos como sortear obstáculos, recorrer una pista y patear penales. El trabajo colaborativo es fundamental en este taller, ya que los y las estudiantes ejecutan las instrucciones desarrolladas en conjunto, utilizando el lenguaje de programación Python. De esta manera, se promueve el trabajo en equipo y el aprendizaje conjunto para alcanzar los objetivos propuestos (Grosclaude, 2014; Jung, 2018).

### **Experiencia**

El modelo didáctico propuesto fue llevado a las escuelas primarias, con la intención de ajustarlo progresivamente. Se realizó la experiencia junto a estudiantes y docentes de cuarto y quinto grado de cinco escuelas primarias de la región, la población destinataria se compone de más de 150 estudiantes en total.

La duración de la actividad es de tres horas y treinta minutos con formato de rotación. Las actividades de introducción y de cierre tienen una duración de quince minutos, los talleres duran una hora cada uno.

Los talleres de desarrollo de aplicaciones móviles y de robótica educativa se realizan en aulas del establecimiento, para el primero se utiliza una computadora portátil, un proyector,

dispositivos móviles para instalar la aplicación, conexión a internet, pizarra y fibrón. Los dispositivos pueden ser los personales de los estudiantes, ya que así se amplifica la posibilidad de apropiación de saberes y artefactos. Para el segundo se requiere una computadora portátil y los robots educativos. El taller Dron Lab se realiza en lugares amplios, como SUM o patio cerrado.

Se reúne a dos o tres grados para trabajar con un rango de entre 20 y 45 estudiantes. Cada taller se realiza con grupos de hasta 15 integrantes. Cada grupo explora dos talleres.

Por medio del trabajo colaborativo y una experiencia lúdica, los estudiantes podrán desarrollar aplicaciones móviles, construir piezas de software para controlar robots y programar drones.

### **Actividades y talleres:**

**Momento cero:** Con el grupo completo de estudiantes, los talleristas se presentan y explican usos de la programación e importancia de aprender sobre Ciencias de la Computación a edades tempranas. Luego se divide el grupo general en tres para realizar los talleres.

**Taller de desarrollo de aplicaciones móviles:** Los estudiantes planifican, diseñan, implementan, testean y modifican proyectos de software.

**Primer Momento:** Se detalla el programa a desarrollar, el entorno de programación y el lenguaje de programación. Así mismo se explica qué es el proceso de desarrollo de software y que modelo de desarrollo se va a usar.

**Segundo Momento:** Mediante el uso de la pizarra se empieza con la definición de los requerimientos de la aplicación a desarrollar, en este caso una versión del preguntados.

**Tercer Momento:** Definidos los requerimientos, se procede a definir la disposición de los elementos en la pantalla.

**Cuarto Momento:** Usando el entorno de programación de App Inventor los estudiantes proceden a programar en la computadora, cada uno realiza una intervención guiados por los demás asistentes al taller.

**Quinto Momento:** Se procede a instalar el software en los dispositivos móviles (pueden ser los propios de los alumnos), donde se testea si se cumplieron los requerimientos, en caso negativo se corrige el software y se testea nuevamente, en caso afirmativo finaliza el taller.

### **Taller dron lab**

Los estudiantes crean un programa que permite un vuelo controlado de un dron. A dicho

programa podrán usarlo, analizarlo, mejorarlo y compartirlo y así comprender las ventajas del software libre. El algoritmo usado tiene composición secuencial bien definida, comienza con la instrucción de despegue y finaliza con la instrucción de aterrizaje, siguiendo el orden en el que fueron colocados los bloques.

**Primer Momento:** Se muestran los dispositivos a usar, se explican las funcionalidades y conceptos básicos de programación que son de interés para la realización del taller, como instrucciones, parámetros. Se trabaja con los bloques de programación de despegue, navegación y aterrizaje.

**Segundo momento:** El tallerista realiza un programa de vuelo del dron, con cinco instrucciones, las explica y lo ejecuta. De las instrucciones explica el color del bloque, datos ingresados por parámetro y que realiza el dron con cada una.

**Tercer momento:** Se forman tres grupos, a cada uno se le asigna una estación que tiene características particulares dadas por las diferencias en el color y tamaño de las bases, colocadas a distancias diferentes. Se presenta la situación problema que podrá tener múltiples soluciones, en este caso, “avanzar a la siguiente estación”.

**Cuarto momento:** Los estudiantes desarrollan estrategias variadas para medir distancias entre estaciones, ya que no cuentan con herramientas de medición. Cuentan pasos, baldosas y utilizan objetos cercanos como puntos de referencia. Después de obtener estas cantidades, los estudiantes deben comparar las longitudes medidas y convertirlas al equivalente en centímetros.

**Quinto momento:** Vuelcan los datos a los bloques, realizan el programa y se procede al vuelo del dron. Se permite realizar dos intentos, ya que en caso de no lograr aterrizar en la estación siguiente, podrán analizar y mejorar el programa.

Una vez concluido cada tramo, el grupo cede el dispositivo móvil, comparte y explica el programa al siguiente, para que estos puedan estudiar cómo funciona el programa heredado antes de realizar el propio. Cada programa realizado es guardado y si es mejorado se hace público a los demás, para que todo el grupo se beneficie.

El taller finaliza cuando logran que el dron supere las tres estaciones.

### **Taller Programando a Frankestito**

Los estudiantes crearán un producto de software que resuelve un problema planteado para el robot: “Esquivar obstáculos”. El comportamiento del robot lo programan usando Python, un lenguaje de programación simple, actual y potente.

**Primer Momento:** Se muestran los dispositivos a usar, se explican las funcionalidades y conceptos básicos de programación que son de interés para la realización del taller, como comandos y parámetros.

**Segundo momento:** Se trabaja con el grupo completo, se presenta la situación problema que podrá tener múltiples soluciones, en este caso, “esquivar obstáculos” y entre todos buscan estrategias para cumplir el objetivo.

**Tercer momento:** Cada estudiante coloca una instrucción y se ejecuta . Una vez que el robot cumplió el objetivo ejecutando instrucciones una en una, se construye un producto de software que sea capaz de esquivar los obstáculos en una ejecución.

**Momento de cierre de la actividad:** Luego de haber completado los tres talleres de programación, es fundamental que los estudiantes reflexionen sobre la experiencia en su conjunto y sobre cómo las habilidades y conocimientos adquiridos en estos talleres pueden aplicarse en situaciones futuras.

¿Cómo creen que las habilidades y conocimientos adquiridos en este taller podrían aplicarse en situaciones futuras? ¿Cómo esta experiencia ha influido en su percepción sobre la programación y el desarrollo de software? ¿Les gustaría seguir aprendiendo y trabajando en proyectos similares en el futuro?

### **Discusión y conclusiones**

La experiencia de llevar a cabo estos talleres y actividades de programación ha sido enriquecedora tanto para los estudiantes como para los facilitadores. A lo largo de este proceso, se han alcanzado varios aspectos importantes que merecen ser discutidos y destacados.

Los participantes han tenido la oportunidad de adquirir habilidades en ciencias de la computación y reflexionar sobre la importancia de aprender estos conceptos en edades tempranas.

En los momentos de reflexión muchos alumnos relacionaron lo aprendido en estos talleres con la automatización de procesos. Esto es significativo, ya que no se encontraba explícito en los objetivos propuestos y añade un valor agregado a su aprendizaje.

La experiencia en su conjunto ha influido en la percepción de los estudiantes sobre la programación y el desarrollo de software, dejando de percibirse como actividades exclusivas para expertos, despertando interés en campos relacionados con la tecnología, ya que expresaron sus ganas de seguir aprendiendo y trabajando en proyectos similares en el futuro,

demostrando el impacto positivo de esta iniciativa.

Es relevante señalar que después de la realización de los talleres, el contacto continuó con las escuelas participantes, permitiendo enriquecer los talleres.

En última instancia, la colaboración entre la universidad y las escuelas ayuda a achicar la brecha entre la tecnología disponible y su integración efectiva, preparando a los estudiantes para un futuro cada vez más tecnológico y automatizado.

## Referencias

- Bai, O., & Chu, H. (2021). Drones in education: A critical review. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(11), 1722-1727.
- Barkley, E. F., Cross, K. P., & Major, C. H. (2014). *Collaborative learning techniques: A handbook for college faculty*. John Wiley & Sons.
- Clerici, C., Naef, E., Eckerdt, M. (2021). El juego como dispositivo didáctico. Marcos teóricos e ideas prácticas para el aula. Espacio editorial de la Universidad de Concepción del Uruguay
- Grosclaude, E., Zurita, R., Castillo, R. D., Lechner, M., & Riquelme, J. (2014). Designing a myro-compatible robot for education as copyleft hardware. In *XX Congreso Argentino de Ciencias de la Computación (Buenos Aires, 2014)*.
- Jung, S. E., & Won, E. S. (2018). Systematic review of research trends in robotics education for young children. *Sustainability*, 10(4), 905.
- Lee, J. S., Blackwell, S., Drake, J., & Moran, K. A. (2014). Taking a leap of faith: Redefining teaching and learning in higher education through project-based learning. *Interdisciplinary Journal of Problem-Based Learning*, 8(2), 2.
- Llambi, C., Borchardt, M., Klinkovich, V., Locca, N., Martínez, C., & Scasso, M. (2023). Aprendizajes y desafíos para la enseñanza de las Ciencias de la Computación en las escuelas.
- Morris, T. H. (2020). Experiential learning—a systematic review and revision of Kolb's model. *Interactive learning environments*, 28(8), 1064-1077.
- Kolb, A. Y., & Kolb, D. A. (2005). Learning styles and learning spaces: Enhancing experiential learning in higher education. *Academy of management learning & education*, 4(2), 193-212.
- K-12 Computer Science Framework Steering Committee. (2016). *K-12 computer science framework*. ACM.
- Patton, E. W., Tissenbaum, M., & Harunani, F. (2019). MIT app inventor: Objectives, design, and development. *Computational thinking education*, 31-49.
- Quintana, R., & Mamani, G. R. (2020). Entre el Plan Nacional de Inclusión Digital Educativa

(PNIDE) y el Plan Aprender Conectados (PAC): las voces de los actores territoriales y escolares.

Clerici, C., Naef, E., Eckerdt, M. (2021). El juego como dispositivo didáctico. Marcos teóricos e ideas prácticas para el aula.

**Enseñanza de fundamentos conceptuales de programación usando Arduino****Gonzalo Pablo Fernández<sup>1,2</sup>, Christian Cossio-Mercado<sup>1,3</sup>**

{gpfernandez, ccossio}@dc.uba.ar

<sup>1</sup> Facultad de Ciencias Exactas y Naturales. Departamento de Computación.  
Universidad de Buenos Aires.

<sup>2</sup> Universidad Nacional de Quilmes

<sup>3</sup> Instituto de Ciencias de la Computación  
CONICET- Universidad de Buenos Aires

**Resumen**

Se diseñó y se dictó un curso corto de inicio en la programación usando exclusivamente placas Arduino y herramientas de electrónica aplicada, enfocado a estudiantes de últimos años del nivel secundario. A diferencia de otros trabajos y experiencias previas, el trabajo con Arduino no estaba pensado para realizar transferencia de conocimientos previamente adquiridos, por medio de otros lenguajes y entornos, sino que se eligió a esta plataforma para la presentación inicial de conceptos esenciales de la programación. Este trabajo tiene como diferencial que resume el diseño y realización de un curso propuesto fue la enseñanza de conceptos de programación, donde se usaron placas Arduino como un medio para alcanzar la comprensión de los temas, pero no como un fin en sí mismo. El curso recibió buenas evaluaciones de parte de las alumnas y los alumnos participantes, quienes pudieron poner a prueba sus conocimientos por medio de un proyecto final realizado como parte del curso. Se pudo ver que la herramienta usada fue suficientemente versátil como para permitir el trabajo de conceptos diversos como división en subtareas, funciones y procedimientos, entre otros. Más allá de que la duración acotada del curso no lo permitió, el diseño elegido abre la puerta para seguir trabajando con otros conceptos como eventos y variables, entre otros. Se espera que este diseño de curso pueda ser útil a otras personas que buscan nuevas formas de introducir a sus estudiantes al mundo de la programación.

**Palabras clave:** Arduino, Programación y Robótica

**Introducción**

En los últimos años cobró gran relevancia la enseñanza de Programación y Robótica en la educación denivel primario y secundario. Así, actualmente la abstracción, la resolución de problemas y el pensamiento creativo son capacidades que se asocian con las Ciencias de la Computación, además de con la Matemática, como ocurría anteriormente. Si bien estas habilidades son comunes a ambas ciencias, dado que la forma tradicional de abordar la Matemática en la escuela es mediante problemas de abstracción numérica, las Ciencias de la Computación (que incluyen a la Robótica, la Inteligencia Artificial, entre otras áreas) otorgan un

enfoque diferencial que posibilita llevar la abstracción y resolución de problemas a un plano más general.

El denominado Pensamiento Computacional incluye capacidades como el reconocimiento de patrones, la descomposición de problemas complejos, la comprensión y el diseño de sistemas, la búsqueda de heurísticas para encontrar una solución, el diseño de algoritmos, entre otros (Nardelli, 2019; Wing, 2006). Todas estas habilidades resultan importantes en la educación media para la comprensión de fenómenos y procesos complejos estudiados en ciencias y para comprender el mundo en el cual viven, además de que son de utilidad para la continuación de estudios en el nivel superior.

Por todas estas razones la enseñanza de las Ciencias de la Computación se promueve en distintos niveles escolares para que forme parte de la currícula general y no solamente como conocimiento específico relacionado a la informática. Su incorporación desde los niveles básicos de la escolarización está tomando gran importancia en todo el mundo, por lo que hay un interés generalizado en llevarlo a cabo de manera efectiva en el largo plazo. En particular, en Argentina hace varios años que se trabaja en cómo introducir a la programación y el Pensamiento Computacional como parte de la enseñanza obligatoria. La resolución N° 263/15 del Consejo Federal de Educación remarca que es de “importancia estratégica para el sistema educativo argentino la enseñanza y el aprendizaje de la programación durante la escolaridad obligatoria, para fortalecer el desarrollo económico-social de la Nación”. En este sentido, la resolución N° 343/18 del Consejo Federal de Educación aprobó los Núcleos de Aprendizaje Prioritarios (NAP) para Educación Digital, Programación y Robótica, estableciendo la obligatoriedad de la enseñanza de estos saberes en todos los niveles educativos obligatorios. En esa misma resolución se estableció un plazo de dos años para la adecuación de la currícula con el objetivo de incorporar dichos contenidos, aunque en la actualidad la implementación no es completa en las escuelas de las distintas jurisdicciones del país.

A pesar de la intención y su promoción institucional, aún no existen programas educativos, recursos preestablecidos ni metodologías estándares para llevar a las Ciencias de la Computación a todas las escuelas del país, ya que la implementación de la resolución del CFE citada es dispar, con resultados concretos recién en el nivel medio, y con baja cobertura en inicial y primario. Por otro lado, el campo de la Didáctica de la Computación se encuentra en una etapa de desarrollo inicial, donde queda mucho por investigar para conocer qué tipo de recursos son propicios a emplear de acuerdo a cada contexto educativo, lo que dificulta el cumplimiento de los objetivos planteados, aunque se avizora un gran desarrollo del área en los próximos años.

La Fundación Sadosky, por medio de la iniciativa Program.ar lleva diez años desarrollando

material y propuestas para impulsar la implementación de contenidos de programación y de Ciencias de la Computación en general. Entre los recursos que desarrollaron se destaca la herramienta PilasBloques (Sanzo et al. 2017), que está enfocada para el aprendizaje de conceptos de programación desde el 1er ciclo de primaria, siguiendo un enfoque de exploración y aprendizaje por indagación.

Uno de los campos más populares y elegidos para abordar la enseñanza de la computación es la robótica. Esta elección radica en su naturaleza tangible la cual facilita la comprensión y el dimensionamiento del comportamiento lógico en estudio en un sentido físico, haciendo que su aprendizaje sea más atractivo e interactivo (Horn et al. 2009).

Existen varios inconvenientes al utilizar a la robótica sin un contexto didáctico pedagógico pensado para el público objetivo (Umaschi Bers, 2022). Además, mal implementado resulta contraproducente ya que, debido a su naturaleza concreta, se contrapone con el objetivo de generar abstracción en los estudiantes, que es uno de los cuatro principales componentes del pensamiento computacional, según Wing (2006). Además, muchos de los entornos más populares son específicos para unos pocos diseños particulares de hardware y están restringidos a funcionar únicamente con tales modelos, como los kits y el entorno de programación de Lego (Fortunati et al. 2022).

Se busca un aprendizaje de la programación que sea significativo, robusto y duradero, y permita que los participantes puedan resolver problemas en una variedad de situaciones, por lo que no es conveniente ceñirse al uso de una herramienta en particular o trabajar con ciertos tipos de problemas. Sólo se puede mostrar una comprensión cabal de un tema si los aprendizajes pueden transferirse a otros entornos y situaciones (i.e., el desempeño flexible que mencionan (Stone Wiske et al. 1999) y otros autores), para lo cual es necesario trabajar sobre los conceptos de la programación y no hacer foco en utilizar una herramienta particular. En otras palabras, se busca enseñar a programar y no a escribir código en un lenguaje particular.

Existen propuestas que utilizan robótica como herramienta para enseñar a programar suelen fallar en mantener el foco en la enseñanza de la programación y terminan dirigidos a enseñar las particularidades de la herramienta usada. En algunos casos, el foco está sólo en llamar la atención de los y las estudiantes (y sus familias), y no se evidencia una comprensión clara del aporte de esta área a la enseñanza-aprendizaje de conceptos esenciales. Existen otros enfoques en donde la robótica aparece sólo para hacer transferencia de conocimiento, luego de haber utilizado otras herramientas como PilasBloques (Sanzo et al. 2017) o Scratch (Resnick et al. 2009), o, inclusive, luego de haber trabajado con otros lenguajes para la enseñanza de los conceptos fundamentales de la programación.

Por otro lado, muchas de las propuestas que proponen usar robótica se enfocan más en trabajar con conceptos específicos del área más que en aquellos de la programación en especial (Filippov et al. 2017; Scaradozzi et al. 2015). En estos trabajos se suele destacar como aspecto positivo de la enseñanza la motivación que consiguen en sus estudiantes pero pocos hacen hincapié en que hayan aprendido algo. Otras propuestas sí apuntan a enseñar programación usando la robótica como una mera herramienta. En (Major et al. 2012) se hace un relevamiento de varios trabajos que utilizan tal enfoque. Si bien la mayoría de los artículos relevados concluye que hubo aprendizaje, el mismo artículo menciona que los estudios no fueron rigurosos.

En este trabajo presentamos el diseño de un curso corto de programación inicial y los resultados preliminares de su primera puesta en práctica. El curso está orientado a personas sin conocimientos previos y tiene como objetivo la introducción a la programación, con foco en las bases conceptuales del área. Para ello se eligió trabajar con placas Arduino, de forma de introducir a los conceptos de la programación, pero manteniendo una vinculación con el mundo físico, con el fin de motivar la atención y fomentar la participación de los y las cursantes.

En el curso se decidió utilizar el entorno de programación por bloques *Arduino en la Escuela* (Fernández et al. 2021). Si bien existen muchos otros entornos de programación por bloques para trabajar con Arduino, la mayoría está orientado a servir de primer acercamiento a la programación de Arduino, sólo como un paso previo para pasar la entorno (IDE) en texto, además de que están enfocados en el aprendizaje de las particularidades de Arduino. De hecho, en general varios de estos entornos por bloques sólo traducen en forma literal los bloques a código en texto esperado por el IDE de Arduino, debido a que no fueron diseñados para la enseñanza y el aprendizaje de los conceptos esenciales de programación, que es nuestro foco en esta propuesta. Por ejemplo, muchas de estas herramientas presentan dificultades o están imposibilitados para incorporar parámetros a los procedimientos y las funciones, en general debido a que no disponen de algoritmos de inferencia y control de tipos (ver resumen del relevamiento incluido en Fernández et al. 2021).

## **Materiales y métodos**

### **Contexto, participantes y metodología**

El curso se dictó en el marco del Programa Talleres de Ciencia organizado por la Dirección de Orientación Vocacional<sup>1</sup> de FCEN-UBA. De la primera puesta en práctica participaron 44 estudiantes de entre 16 y 18 años de escuelas medias del AMBA. Aunque sólo 5 (11%) personas cursan sus estudios en instituciones con orientación en informática, el 88% aseguró

---

<sup>1</sup> <https://exactas.uba.ar/dov>

haber tenido formación en contenidos relacionados con programación (e.g., computación, informática y tics). Más específicamente el 30% del total había usado Scratch en la escuela, mientras que un 5% había utilizado PilasBloques como parte de las clases que tomó, y el 13% había usado placas Arduino antes.

El curso se organizó en cinco encuentros semanales de tres horas cada uno. En el primer encuentro, antes de comenzar con las actividades, se realizó una encuesta inicial por medio de un formulario electrónico en línea, de forma de relevar los conocimientos previos en programación. Por otro lado, el último encuentro se cerró con una encuesta final para medir el progreso durante el curso, así como para evaluar las satisfacciones de las expectativas de los y las participantes.

### **Diseño general del curso**

Para el diseño del curso se siguió un marco teórico-conceptual propio que se utiliza desde experiencias anteriores, definido bajo el nombre PRENDER (Cossio-Mercado y Fernández, 2023), el cual integra en una sola propuesta a la Enseñanza para la Comprensión (Stone Wiske, 1999), la Evaluación para el Aprendizaje (Anijovich & Cappelletti, 2017), Aprendizaje por Indagación (Schwab, 1960 y 1962), el trabajo con problemas auténticos (Pozo Municio & Postigo Angón, 1994), además de los lineamientos para la enseñanza de las bases conceptuales de la programación en (Martínez López, 2013).

El curso consistió en cinco encuentros de tres horas cada uno, donde el último fue dedicado especialmente al trabajo con proyectos en grupo: el *primer encuentro* fue introductorio y en él se abordaron aspectos básicos sobre la electrónica necesarios para empezar a interactuar con placas Arduino (como los conceptos de sensores y actuadores) y los elementos fundamentales de la programación (comandos, expresiones y programas). El *segundo encuentro* se enfocó en la división en subtareas como mecanismo para mejorar la legibilidad de los programas, así como para reutilizar porciones de código ya definidas, y se presentaron los procedimientos y los parámetros. En el *tercer encuentro* se siguió trabajando con división en subtareas y se presentaron las funciones y algunos comandos y expresiones compuestas (la alternativa condicional y operadores aritméticos y booleanos). En el *cuarto encuentro* se plantearon actividades que requerían utilizar todas las herramientas vistas en los primeros 3 encuentros pero que además requerían experimentar de alguna forma con los componentes para terminar de entender cuál debía ser el comportamiento a programar. En varios de los encuentros, también, se incluyeron discusiones conceptuales respecto a qué es la computación y qué es programar (como, por ejemplo, hablar del origen de las computadoras o sobre la representación de información). Finalmente, *el quinto y último encuentro* se destinó por entero al trabajo en grupo con proyectos, los cuales se podían elegir de una lista de propuestas o el equipo podía

optar por trabajar con un proyecto propio con ayuda del equipo docente.

El curso se diseñó con un fuerte componente práctico, pero intercalando espacios de discusión para asentar los conocimientos. En promedio, cada encuentro consistió en dos horas de práctica, durante las cuales se trabaja en grupos de 2 o 3 personas en alguna actividad propuesta, y una hora de espacio de discusión, que incluyó puestas en común de las actividades, formalización de los nuevos conceptos que se presentaron en la actividad y discusión sobre posibles soluciones alternativas. En líneas generales, se buscó mantener la cohesión de cada clase individualmente, así como la conexión con el resto de ellas, por lo que se decidió que cada clase comenzará con un repaso de lo visto en la clase anterior y finalizará con un resumen de los nuevos contenidos vistos.

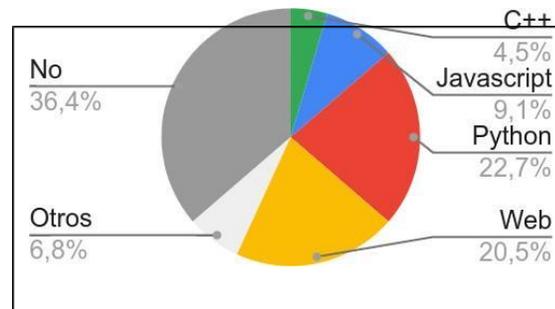
### **Tipos de actividades realizadas**

Al principio del primer encuentro se realizó una actividad señuelo, siguiendo en anteriores, con el fin de tener una referencia conceptual del funcionamiento de un sistema y sus partes. Estas ideas serían retomadas durante todo el curso en casi todas las actividades. La mayor parte del tiempo de los primeros cuatro encuentros se dedicó a actividades prácticas que consistieron en presentar un enunciado y dejar a los grupos que trabajen por su cuenta. Durante estas actividades se aplicó el enfoque de aprendizaje por indagación estructurada (Banchi y Bell, 2008) ya que la solución esperada requería utilizar herramientas que no habían sido presentadas aún o que sí habían sido presentadas, pero debían usarse de una forma distinta a como habían sido usadas hasta el momento. Al estar trabajando con una herramienta de programación por bloques, este tipo de actividades fueron muy productivas permitiendo a una gran proporción de los grupos encontrar alguna de las soluciones esperadas y, a su vez, que algunos grupos encontraran soluciones alternativas que no estaban contempladas.

No todas las actividades prácticas consistieron en programar un comportamiento en el Arduino. Para formalizar los conceptos de parámetros y de división en subtareas también se propusieron actividades desconectadas. Otro tipo de actividades consistió en una consigna abierta pero acotada en la que se esperaba que sean los y las estudiantes quienes decidieran qué hacer, dentro de un espectro acotado de posibilidades. El objetivo de estas actividades fue promover progresivamente la autonomía de cara al proyecto del último encuentro, aplicando una variedad de actividades a lo largo del curso (Tamir, 1989; Grau, 1994). Algunos ejemplos de este tipo de actividades fueron armar una imagen a elección para dibujar en la matriz de leds y componer una melodía a elección para reproducir en el zumbador (*buzzer*). Un último tipo de actividades consistió en experimentar para deducir un comportamiento a partir de lo visto. En el anexo se puede ver el detalle de las actividades realizadas por cada encuentro del curso.

## Resultados

Los 44 participantes respondieron la encuesta inicial realizada al principio del primer encuentro. Allí se les preguntó si sabían programar y, de ser así, en qué lenguajes. En la Figura 1 se muestran las respuestas a esa pregunta tabuladas en base a los lenguajes que afirmaron conocer.



**Figura 1:** Respuestas tabuladas a la pregunta *¿Sabés programar? ¿En qué lenguajes?* La etiqueta *Web* corresponde a respuestas relacionadas a html y css.

De los 44 participantes, sólo 32 completaron la encuesta final. En la encuesta final se les preguntó si sabían programar antes de tomar el curso y si sabían programar ahora que había finalizado el mismo. En ambos casos las respuestas se tabularon con las etiquetas *Sí*, *Poco* y *No*. En el caso de la segunda pregunta también aparecieron respuestas del estilo “un poco más que antes” dado que a la primera habían respondido que sí. En la Figura 2 se muestran los pares de respuestas a esas dos preguntas.

En la Figura 2 se puede ver cómo fue el cambio de autopercepción de los y las participantes con respecto a su conocimiento sobre programación, lo que fue consultado en la encuesta de cierre. En cada caso se muestra autopercepción antes de comenzar el taller y luego de finalizar el mismo. Con tonos de verde se muestran las respuestas que parecen indicar que el conocimiento aumentó tras haber cursado. Con tonos de azul se muestran las respuestas que parecen indicar que el curso no influyó en el conocimiento, ya que respondieron lo mismo. Con tonos de rojo se muestran las respuestas que parecen indicar que el curso tuvo un impacto negativo en el conocimiento sobre programación.

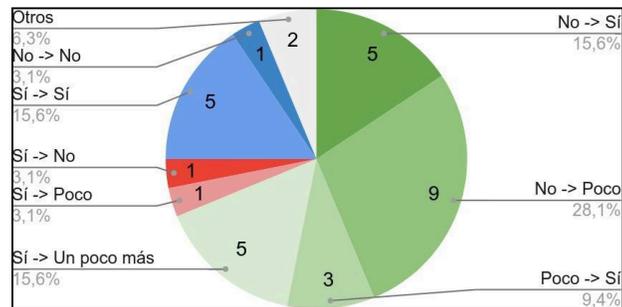


Figura 2: ¿Sabías programar antes de tomar el curso? y ¿Sabés programar ahora?

Se destaca que de las 8 personas que en la encuesta inicial afirmaron no saber programar a pesar de haber usado entornos de programación por bloques, 4 cambiaron de opinión en la encuesta final y aseguraron que sí sabían programar antes de tomar el curso. A su vez, de las 9 personas que en la encuesta inicial aseguraron saber programar, 4 negaron en la encuesta final que supieran programar antes de tomar el curso.

En el Cuadro 1 se sintetiza el promedio de respuestas a cada una de las preguntas de satisfacción tanto respecto al taller como al uso de la herramienta Arduino en la Escuela. Todas las preguntas se respondieron con una escala likert de 1 a 5.

A pesar de que el objetivo del curso fue enseñar a programar de forma general, haciendo foco en los conceptos y no en un lenguaje específico, varias de las respuestas obtenidas en la encuesta final dejaban ver que algunos participantes no llegaron a comprenderlo. A la pregunta “¿Qué te gustaría haber aprendido en el taller que no aprendiste? ¿Por qué te interesa?” se recibieron respuestas como “Me hubiese interesado trabajar más con el código que con bloques porque me parece que me va a ser de mayor utilidad para el futuro y tiene mayor complejidad” o “capaz algo sobre algoritmos” además de que mucha gente mostró interés en aprender un lenguaje particular como python. También se apreció esta deficiencia en respuestas a la pregunta “¿Sabés programar ahora?” con respuestas como “Si, pero no en su totalidad. Esto debido a que no hay únicamente un lenguaje” o “ahora si se programar pero solo en el idioma de arduino en la escuela”.

## Conclusiones

A lo largo de las cinco clases del curso se buscó enseñar los conceptos esenciales de la programación, de forma independiente de la herramienta usada. De acuerdo al contacto personal con los y las participantes, se pudo apreciar una buena recepción del curso y de la herramienta utilizada. Específicamente, de acuerdo a los resultados de la encuesta de cierre, hubo un cambio en la autopercepción del conocimiento acerca de programación y, en su mayoría, se reporta una mejora sustancial.

Adicionalmente, los y las participantes indicaron que el curso fue de su agrado, aunque no todos no terminan de apreciar que los aprendizajes logrados pueden ir más allá del entorno Arduino y la herramienta elegida para el curso.

Finalmente, se identifica como indispensable que a lo largo del curso se intercale el trabajo con diferentes herramientas y, en particular, se realicen actividades con diferentes lenguajes de programación de uso general, como Python. Esto permitiría mejorar la autopercepción del conocimiento que se tiene de programación, además de facilitar la transferencia a otros entornos y herramientas.

Pregunta	Promedio
¿Cuánto dirías que el taller cumplió con tus expectativas?	4,10
¿Cuál te parece que fue el nivel de dificultad del taller?	2,97
¿Qué te pareció la velocidad con la que se dieron los temas y se trabajó con las actividades del taller?	3,53
¿Qué te pareció en general la herramienta Arduino en la Escuela?	4,28
Fácil de usar (no presenta complejidades al momento de hacer lo que quiero hacer)	3,59
Intuitiva (es fácil encontrar las cosas que necesito en el momento que las necesito)	3,69
Entretenida (me divierte hacer actividades y me dan ganas de hacer más actividades similares)	4,13
Potente (me permite realizar las cosas que quiero realizar)	3,69
Educativa (mientras lo uso voy entendiendo las cosas que hago y cuando algo falla me resulta fácil entender por qué falla)	3,88
Frustrante (me topo constantemente con problemas que no puedo resolver y la herramienta no me ayuda a entender cómo solucionarlos)	2,78
Compleja (me cuesta entender qué hace cada cosa y cómo hacer lo que quiero hacer)	2,59
Motivadora (usarla me dan ganas de seguir usándola, de seguir explorando y de seguir pensando qué otras cosas podría hacer)	3,78
Visualmente atractiva (la interfaz es agradable a la vista)	3,94
Aburrida (resolver ejercicios es un tedio y prefiero ponerme a hacer otra cosa)	2,09
En general, ¿cómo fue tu experiencia usando Arduino en la Escuela?	4,25

Cuadro 1: Promedios de respuestas a las preguntas de satisfacción.

### Trabajo futuro

De acuerdo a lo mencionado anteriormente, en próximas ediciones del taller se realizará un cambio en el diseño del mismo, de forma de agregar más clases (hasta llegar a los 8 encuentros), tanto para poder desarrollar más los temas ya incluidos, como para permitir

actividades donde se usen otros lenguajes de programación, tanto de bloques (e.g., PilasBloques) como de texto (e.g., Python). Así, se buscará que los y las participantes puedan apreciar mejor que adquirieron conceptos útiles de programación, que pueden utilizarse para resolver problemas de su interés, donde se usan otras herramientas y lenguajes.

En lo respectivo a la herramienta utilizada, sería deseable que la misma cuente con una versión en modo texto, pero manteniendo la legibilidad de Arduino en la Escuela (i.e., que tenga las mismas etiquetas que la versión en bloques, como, por ejemplo, `sensor.medir()`), de forma de poder usarla en una transición hacia lenguajes en texto, como paso previo a pasar al trabajo con Python y otros lenguajes.

### Referencias

- Anijovich, R. & Cappelletti, G. (2017). *La evaluación como oportunidad*. Buenos Aires, Argentina: Paidós.
- Banchi, H. & Bell, R. The many levels of Inquiry. (2008). En *Science and Children*, Vol. 46, No. 2, pp. 26–29.
- Cossio-Mercado, C. & Fernández, G. P. (2023). PRENDER: Una propuesta didáctico-pedagógica para la enseñanza de las Ciencias de la Computación. Enviado a las Jornadas Argentinas de Didáctica de las Ciencias de la Computación 2023.
- Fernández, G. P., Ticona Oquendo, M. B. & Cossio-Mercado, C. (2021). Arduino en la Escuela: una herramienta versátil para la enseñanza de programación y robótica. En *Jornadas Argentinas de Didáctica de las Ciencias de la Computación 2021*, pp. 32-44.
- Filippov, S., Ten, N., Shirokolobov, I. & Fradkov, A. (2017). Teaching Robotics in Secondary School. En *IFAC-PapersOnLine*, Vol. 50 No. 1, pp. 12155-12160. doi: 10.1016/j.ifacol.2017.08.2143
- Fortunati, L., Manganelli, A.M., & Ferrin, G. (2022). Arts and crafts robots or LEGO® MINDSTORMS robots? A comparative study in educational robotics. *International Journal of Technology and Design Education*, No. 32, pp. 287–310. doi: 10.1007/s10798-020-09609-7
- Grau, R. (1994). “¿Qué es lo que hace difícil una investigación?”. En *Revista Alambique. Didáctica de las Ciencias Experimentales*, No. 2, pp. 27–35, Editorial Graó.
- Herron, M.D. (1971). “The nature of scientific inquiry”. En *School Review*, Vol. 79, No. 2, pp. 171–212.
- Horn M.S., Solovey E.T., Crouser J.C. & Jacob R.J.K. (2009). Comparing the Use of Tangible and Graphical Programming Languages for Informal Science Education. En *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, April 2009, pp. 975-984. doi: 10.1145/1518701.1518851

Major, L., Kyriacou, T. & Brereton, O.P. (2012). Systematic literature review: teaching novices programming using robots. En IET Software, Vol. 6, No. 6, pp. 502-513, doi: 10.1049/iet-sen.2011.0125

Martínez López, P. E. (2013). Las bases conceptuales de la Programación: Una nueva forma de aprender a programar. 1ra ed. La Plata : el autor, EBook. ISBN 978-987-33-4081-9

Nardelli, E. (2019). Do we really need computational thinking?. En Communications of the ACM, Vol. 62, No. 2, pp. 32-35. doi: 10.1145/3231587

Pozo Municio, J.I. & Postigo Angón, Y. (1994). "La solución de problemas como contenido procedimental en la educación obligatoria". En Pozo Municio et al (1994). La solución de problemas. Editorial Santillana.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y.. (2009). Scratch: programming for all. En *Commun. ACM* Vol. 52, No. 11 (November 2009), pp. 60–67. doi: 10.1145/1592761.1592779

Sanzo, A., Schapachnik, F., Factorovich, P., & O'Connor, F. S. (2017). Pilas bloques: A scenario-based children learning platform. En Twelfth Latin American Conference on Learning Technologies (LACLO), La Plata, Argentina, 2017, pp. 1-6, doi: 10.1109/LACLO.2017.8120926.

Scaradozzi, D., Sorbi, L., Pedale, A., Valzano, M. & Vergine, C. (2015). Teaching Robotics at the Primary School: An Innovative Approach. En *Procedia - Social and Behavioral Sciences* Vol. 174 pp. 3838-3846. doi: 10.1016/j.sbspro.2015.01.1122

Schwab, J. (1960). "What Do Scientists Do?". En *Behavioral Science*, Vol. 5, No. 1, pp.

1–27.

Schwab, J.J. (1962). "The teaching of science as enquiry", en J.J. Schwab and P. F. Brandwein (Eds.). En *The teaching of science*, Cambridge, Harvard University Press,

pp. 1–103.

Stone Wiske, M. et al. (1999). La enseñanza para la comprensión. Buenos Aires: Paidós. Tamir,

P. (1989). "Training teachers to teach effectively in the laboratory". En *Science*

*Education*, Vol. 73, No. 1, pp. 59–69.

Umaschi Bers, M. (2022). Coding as another language: a pedagogical approach for teaching computer science in early childhood. En *Journal of Computers in Education*, No. 6, pp. 499–528. <https://doi.org/10.1007/s40692-019-00147-3>

Wing J. M. (2006). Computational thinking. En *Communications of the ACM*, Vol. 49, No.

3, March 2006, pp. 33-36. doi: 10.1145/1118178.1118215

## **Anexo**

A continuación se resumen las actividades realizadas en cada encuentro del curso.

### ***Primer encuentro***

- Explorando sensores y actuadores: Actividad de tipo señuelo en la que se le da a cada grupo un sistema compuesto por un Arduino, un sensor y un actuador y se les pide interactuar con el sistema con el objetivo de descubrir cuál es el comportamiento programado. La actividad sirve para introducir los conceptos de sensor, actuador y de comportamiento automático.
- La automatización en la vida cotidiana: Actividad de discusión en la que se propone identificar distintos escenarios en la vida cotidiana que presentan comportamiento automático (como las alarmas, los ascensores o la pava eléctrica) y sus componentes (tanto sensores como actuadores) involucrados.
- Encendiendo una luz LED: Se presenta el entorno Arduino en la Escuela y se propone comenzar programando el comportamiento más simple: encender el LED incorporado de la placa. Se reparte un kit a cada grupo y se deja indagar para encontrar una solución.
- Simulando un Flash: Se propone modificar la solución para que la luz se encienda durante un breve período y luego se apague. En esta actividad empiezan a aparecer ciertas características notables sobre el funcionamiento de Arduino entre las que se destaca el hecho de que el programa principal no tiene principio y fin como en la mayoría de los autómatas convencionales sino que este se ejecuta en ciclo indefinidamente.
- LED externo fijo: Actividad guiada en la que se explican muy brevemente los principios fundamentales de la electricidad que permiten alimentar desde la placa un componente externo ejemplificado con una luz LED. Al proveerle alimentación constante, esta se mantiene encendida.
- LED externo programado: Actividad indagatoria en la que se busca una solución similar a la de la actividad *Encendiendo una luz LED* pero ahora controlando una luz externa en lugar del LED incorporado de la placa.
- Matriz LED: Actividad de proyecto en la que se les explicó cómo conectar la matriz de

LEDs y se les propuso que realicen sobre ella algún dibujo a elección.

- Cierre del primer encuentro: Se retomaron algunos aspectos del funcionamiento de la placa Arduino y los conceptos de sensor y actuador.

### ***Segundo encuentro***

- Repaso del encuentro anterior: Se retomaron los conceptos de sensor y actuador y se repasaron las distintas soluciones a cada una de las actividades.
- Qué es una computadora: Actividad de discusión en la que se busca clasificar distintas entidades que podrían ser llamadas “computadoras”.

- **Código morse:** Actividad indagatoria en la que se propone hacer un programa que genere el patrón de código morse para algunas palabras. Luego se pide modificar el código para pasar a describir otras palabras similares. El objetivo de esta actividad es presentar la noción de procedimientos encapsulando el comportamiento de varios comandos en las acciones de describir una palabra, una letra o un símbolo (un punto o una línea) particular.
- **¿Cómo te llamás?:** Actividad de proyecto en la que se propone utilizar los procedimientos definidos en la actividad anterior, y otros nuevos, para que realicen programas que generen el código morse de sus nombres, o de las palabras u oraciones que elijan.
- **La súper merienda:** Actividad desconectada en la que se propone escribir en papel un programa para preparar la merienda. Esta actividad sirve para reforzar el concepto de procedimiento e introducir el de división en subtareas, así como para mostrar un problema que admite infinitud de soluciones válidas.
- **Melodías Divertidas:** Volviendo a Arduino, ahora se muestra cómo usar el buzzer para reproducir una nota musical. Se propone armar procedimientos para las distintas notas y usarlos para componer una melodía. Para esta actividad se contó con un listado de melodías de las cuales se esperaba que eligieran una para componer pero también se permitió que realizaran una nueva melodía desde cero. Esta actividad permite introducir parámetros ya que los procedimientos específicos de cada nota resultan ser casi idénticos.
- **Cierre del segundo encuentro:** Se formalizan los conceptos de comando y procedimiento que habían sido mencionados pero no definidos. Se habla un poco sobre parámetros pero no se profundiza demasiado en el tema.

### ***Tercer encuentro***

- **Repaso del encuentro anterior:** Se retoman los conceptos de comando y procedimiento y se formaliza el de parámetro, retomando cómo había sido su presentación y cómo habían sido usados en el encuentro anterior. También se realiza una breve actividad de experimentación para comprender un poco más el funcionamiento de los parámetros en un programa.
- **La puerta del colegio:** Actividad de discusión en la que se presenta un caso hipotético de un problema sencillo y se trabaja en pensar posibles soluciones.

- ¿Está oscuro?: Actividad indagatoria en la que se propone utilizar el sensor de luminosidad para realizar un sistema de encendido automático ante la ausencia de luz. Esta actividad permite introducir la alternativa condicional y el uso de sensores. Además, es la primera actividad en la que se desarrolla un sistema con un comportamiento claramente definido.
- Nivel de luz: Actividad indagatoria en la que se propone modificar la solución a la actividad anterior para que el umbral de oscuridad no sea determinado por el sensor sino por quien está programando el sistema. En la primera parte de esta actividad se pretende plantear la estrategia con las herramientas conocidas hasta el momento. Luego se realiza una actividad de experimentación para comprender

cómo funciona el sensor de luminosidad y cómo se interpretan los datos que mide. Finalmente se desarrolla la implementación de la solución introduciendo operadores de comparación. El cierre de esta actividad permite discutir acerca de los tipos de señales que puede interpretar el Arduino que se puede relacionar fácilmente con tipos de datos.

- Luz y movimiento: Se complejiza el sistema (y de esta forma, su comportamiento) agregando un sensor de movimiento para que la luz sólo se encienda al detectar movimiento. Esta actividad permite trabajar composición de expresiones a través de operadores booleanos y presentar funciones.
- Cierre del tercer encuentro: Se retoma el concepto de tipo de dato, se formaliza el concepto de función y se arma una tabla con las distintas herramientas vistas hasta el momento. También se realiza un repaso de los distintos tipos de conexiones que se usaron con la placa Arduino hasta el momento y se observa la relación entre estos y los tipos de las expresiones.

Elementos del lenguaje	Comandos	Expresiones
<b>Literales</b>		Números, pines, valores de verdad, notas, textos, etc 
<b>Primitivos</b>	Comandos primitivos 	Expresiones primitivas 
<b>Definiciones propias</b>	Procedimientos 	Funciones 
<b>Composición</b>	Alternativas y repeticiones 	Operadores 

### Cuarto encuentro

- Repaso del encuentro anterior: Se retoma la tabla de herramientas con la que se cerró el encuentro anterior y la última solución propuesta para la actividad de la luz que se enciende automáticamente.
- Intensidad variable: A partir de la solución del encuentro anterior se propone modificarla para que la intensidad con la que se enciende la luz sea proporcional al nivel de oscuridad

sensado. Esta actividad permite introducir los operadores aritméticos y los pines PWM.

- Estación de medición: Se propone un proyecto bastante común en Arduino que consiste en un sistema que registre mediciones a lo largo del tiempo. Se muestran algunos posibles ejemplos y se elige un sensor arbitrario para incorporar. Tras ver cómo acceder a las lecturas del sensor se agrega el lector de tarjeta SD para registrar en una tarjeta los datos medidos. Se menciona también que está la

posibilidad de agregar también un reloj para registrar la fecha y la hora de cada medición.

- **Asistente de estacionamiento:** Se propone otro proyecto que consiste en simular el asistente de estacionamiento de los autos. En una primera etapa se muestra el sensor de distancia que se va a usar. Luego se motiva a que prueben distintas estrategias para simular el comportamiento deseado. En el medio de la actividad se realiza una experimentación para observar mejor cómo funciona el dispositivo.
- **Cierre del cuarto encuentro:** Se vuelve a retomar la tabla de herramientas y se hace un repaso rápido de todos los conceptos vistos a lo largo del curso. Luego se retoma cómo fue el proceso de pensamiento para llegar a la solución en la actividad del asistente de estacionamiento y las distintas etapas que se fueron atravesando. A partir de este resumen se formaliza el concepto de programar como manera de comunicar una solución a un problema a partir de plantear el problema, analizarlo, considerar diferentes alternativas, analizarlas y finalmente implementar una de ellas de forma que sea entendible tanto para las computadoras como para las personas. Por último se presentan algunos posibles casos para elegir como proyecto final aunque se deja abierta la posibilidad a que elijan uno propio.

### ***Quinto encuentro***

- Se presenta una guía para seleccionar un proyecto que deberán desarrollar durante el encuentro. La guía permite elegir un proyecto ya definido o armar uno nuevo a partir de combinar varios o de tomar uno incompleto y que el grupo decida cómo completarlo. Durante un tiempo los grupos trabajan en un documento en el que deben describir detalladamente el proyecto (nombre, objetivos, materiales necesarios, etc.). Luego se les pide que armen un prototipo y programen, al menos, la principal funcionalidad del sistema. Para finalizar se les pide un boceto para un poster de divulgación que deberán presentar a fin de año. Durante los últimos minutos del encuentro se les pide que completen la encuesta final.

## **La Programación y Robótica en la formación docente del Nivel Inicial. Estrategias y resultados de un dispositivo de formación situada.**

Mariela I.S. Hirtz, Silvina Manzor, Susana Morales  
mhconectar.rn@gmail.com, msconectar.rn@gmail.com, smconectar.rn@gmail.com

PLANIED – PNIDECI  
Ministerio de educación y DDHH de la Provincia de Río Negro

### **Resumen**

La entrega de Aulas Digitales Móviles (ADM) para los establecimientos del Nivel Inicial por parte de programas nacionales inició en el año 2019. Desde entonces se ha desarrollado una estrategia de formación docente en la provincia que acompaña la entrega del equipamiento de programas nacionales, con la finalidad de fortalecer la inclusión de recursos digitales en la enseñanza.

En el siguiente trabajo nos centraremos en los desafíos, estrategias y resultados del dispositivo de formación docente que tuvo como eje la enseñanza de la programación y la robótica para el nivel inicial, a partir de la incorporación de la iniciación a las nociones de programación en un marco de juegos, trabajando a partir de ejemplos y problemas, enfatizando el uso creativo y no meramente instrumental de las tecnologías.

El formato taller y la modalidad presencial de la propuesta se desarrolló durante los años 2019 y 2022 y contaron con la declaración de interés educativo. Los desafíos de la propuesta que tiene como eje la programación y la robótica para el nivel inicial, radica en la reciente inserción en el ámbito educativo formal. Los NAP de educación digital se constituyen en el año 2018 a partir de la resolución 342/18 y el último diseño curricular para el nivel inicial que incluye a la educación digital como eje transversal data del año 2019. Por lo tanto, resulta una propuesta novedosa en la formación docente del nivel inicial. En segundo lugar, aparecen las expectativas y representaciones docentes sobre la robótica y la programación, por lo que fue necesario tomar estrategias de sensibilización para reflexionar sobre su dimensión educativa.

Las distintas dinámicas de trabajo a partir de juegos se constituyeron como posibilidades de trabajo a modo de introducir a la programación a las niñas y los niños. Reflexionar sobre las finalidades, estrategias y contenidos propició la definición de una propuesta de tipo "Didáctica de la Robótica para el nivel inicial" que se construyó en base a las orientaciones del equipo nacional con fuerte impronta del equipo provincial, a partir del área de la coordinación pedagógica, técnica y las docentes del nivel inicial.

La propuesta alcanzó a 555 docentes y aproximadamente 10.000 niñas y niños de la provincia. La fortaleza del proyecto se centró, además de la presencialidad, en el dispositivo taller con

actividades prácticas y la orientación de propuestas pedagógicas para contribuir a las experiencias en las salas.

### **Palabras Clave** Formación docente - Nivel Inicial - Programación – Robótica

#### **Introducción**

En el año 2019 el programa nacional “Aprender conectados” (decreto 386/2018) inició la entrega de equipamiento<sup>1</sup> a los establecimientos públicos del nivel inicial de nuestro país. En la provincia de Río Negro, ese año se entregó al 51% de los establecimientos de la provincia, que recién se completara a todos las instituciones del nivel inicial en noviembre del 2022 bajo el programa “Conectar Igualdad”. Denominaremos “primera etapa” al equipamiento entregado durante el 2019 y “segunda etapa” al entregado en noviembre del 2022.

La distinción que se desprende entre ambas etapas es el kit de robótica educativa que sólo se incluyó en la primera etapa (2019). El kit de robótica se compone de dos dispositivos robóticos denominados “Robotita”, tres alfombras (una transparente, el “Mapa del tesoro” y “Ciudad”) y tarjetas con indicaciones para realizar juegos de iniciación a la programación. Otra distinción se refiere al contenido incluido en las tablets. Durante la primera entrega las tablets incluyeron dos modos de funcionamiento, el “modo docente” y el “modo alumno”. El primer modo permite el acceso a Android, por lo tanto, es posible descargar aplicaciones y configurar aspectos generales del uso del dispositivo, en tanto el segundo modo, contiene aplicaciones y materiales precargados, entre los que se incluye el programa para el aprendizaje infantil de programación en bloques “ScratchJr”, que puede utilizarse sin necesidad de conexión a internet. En cuanto a los ADM NI de la segunda etapa (2022), no incluye el kit de robótica y las tablets no poseen el modo alumno

Los trayectos de formación docente que ofrecimos como equipo provincial constituido por un equipo de coordinación técnica y pedagógica, docentes del nivel inicial (facilitadoras pedagógicas) y facilitadores técnicos, convocaron a las docentes y equipos directivos del nivel inicial durante el año 2019 y 2022 de los jardines que recibieron equipamiento durante la primera etapa<sup>2</sup>, mientras que los establecimientos que recibieron equipamiento en la segunda etapa están participando de los trayectos 2023.

Es por eso que a fines de esta ponencia, nos referiremos a los trayectos de formación docente dictados durante el 2019 y el 2022 denominados “Trayecto de Formación situada: Recursos

---

<sup>1</sup> El equipamiento genérico denominado Aulas Digitales Móviles para el Nivel Inicial (en adelante ADM NI) se compone de un carro de carga y guarda que contiene una notebook, tablets, una pizarra digital interactiva, un proyector, un servidor portátil, un parlante y un micrófono.

<sup>2</sup> Al tratarse de una propuesta presencial, el trayecto se vio interrumpido durante el 2020 y el 2021 por los límites que imponía a la presencialidad el distanciamiento sanitario

Digitales en el Nivel Inicial” (resolución de declaración interés educativo de la provincia nro. 5416/19) y “Aula Digital Móvil: Educación Digital para el Nivel Inicial” (resolución de declaración de interés educativo nro. 1703/22) porque son los trayectos en los que el eje transversal de la propuesta se orienta a la enseñanza de la programación y la robótica para el nivel inicial. Esta impronta ha significado recuperar los sentidos de las políticas educativas dispuestos en los documentos del Ministerio de Educación de la Nación (NAP de Educación digital, guía de orientaciones, entre otros) y en el diseño curricular para el nivel inicial (2019) y las prácticas de enseñanza.

Entendemos que el enfoque de Alfabetización Crítica Digital de David Buckingham (2012) es una referencia que enmarca nuestras finalidades formativas, tanto en la formación permanente como en las prácticas docentes en la enseñanza obligatoria, ya que “las habilidades críticas y creativas en relación con los nuevos medios (...) debe constituirse en un derecho educativo básico” (p.187). En línea con esta perspectiva, Barbará, M.P (2015) recupera estos sentidos para la formación de docentes en el nivel inicial

Teniendo en cuenta la alfabetización en informática debemos pensar que para que el docente sea alfabetizador debe conocer usos y funciones de las TIC, y de esta forma poder enseñar de manera crítica a los niños el manejo de los dispositivos para que ellos puedan producir, jugar, investigar, comunicarse (p.80).

Como equipo provincial organizamos nuestras propuestas con el principal propósito de fortalecer los conocimientos y estrategias para el uso pedagógico, incluido en la planificación de los saberes de los campos de experiencias del nivel inicial, y promover el cuidado, el uso y la incorporación de los dispositivos en las prácticas de enseñanza en las salas de 4 y 5 años.

### **Metodología de trabajo:**

El trayecto formativo se organizó en dos encuentros presenciales y se realizaron en los jardines que contaban con el equipamiento (51% de los jardines de la provincia divididos en los 15 consejos escolares). En total fueron cuatro cohortes entre el 2019 y el 2022. Entre los encuentros presenciales se realizaron ajustes a las actividades propuestas de acuerdo con las necesidades de las docentes, atendiendo entre otros aspectos, a la alfabetización digital, a los emergentes técnicos y los proyectos institucionales que se realizaban en los jardines. Entendemos que nuestra propuesta desde ejemplos y actividades prácticas orientan la apropiación del uso de dispositivos a partir de actividades del nivel inicial, y que funcionen como orientadores en las propuestas de actividades, sin intenciones de convertirse en aspectos prescriptivos de las prácticas docentes.

Durante el primer encuentro para trabajar en la iniciación a la programación, orientamos

estrategias que permitan recuperar nociones del razonamiento lógico, que posteriormente puedan recuperarse en el ejercicio de la programación. Para ello, utilizamos la pizarra del nivel inicial en sus distintos modos. Trabajamos desde ejemplos, que permitían a las docentes apropiarse del uso de la pizarra a partir de ejemplos que pudieran llevar a sus salas. Por ejemplo, el modo “mouse” de la pizarra digital permite mover elementos para ordenar procedimientos a partir de imágenes. Esto implica en la sala reconstruir un proceso, y para ello recurrimos a aquellos procesos que suelen trabajarse en el nivel inicial, tal como el cepillado de dientes, el crecimiento de una planta, los momentos del día y las rutinas. De esta manera se propuso repasar nociones temporales (antes – después). El “modo pizarra” que permite intervenir las imágenes permite resolver laberintos, dibujar procesos (como el crecimiento de una planta o mariposa) y repasamos las opciones de trabajo con el modo “lienzo”. A continuación, trasladamos la atención a las nociones espaciales desde una perspectiva matemática (adelante, atrás, derecha, izquierda). Para ello, les propusimos a las docentes un desafío en grupos, en el cual dibujaron en una hoja una cuadrícula (de 4 casilleros de alto por 4 de ancho) en la que definieron un lugar de salida, un lugar de llegada y dos obstáculos. Luego, en el piso dibujamos con cinta un tablero con las mismas características y les asignamos los siguientes roles: Líder del grupo, “Mensajera” (es aquella integrante que sólo recibe las indicaciones que les mandarán las programadoras con tarjetas) y las programadoras. Guiándose por las tarjetas de dirección del kit (indicaciones arriba/avanzar, abajo/retroceder, derecha, izquierda, Iniciar, detenerse, repetir) marcaron el recorrido que ellas consideraron correcto para ir desde un punto al otro. En un segundo momento, replicaron la cuadrícula en el piso y dispusieron las tarjetas para orientar los diferentes recorridos pensados en grupos, verificando de esta manera si el recorrido pensado fue el correcto y de otras posibles soluciones. Se dispuso de un momento de planificación de experiencias a realizar con los grupos de las salas.

Durante el segundo encuentro de formación se profundizó en el trabajo con “Robotita”, los tableros y el programa Scratch Jr. (segunda y tercera etapa de la didáctica de la robótica propuesta por la orientación del equipo nacional). El trabajo con Robotita supone una transición entre el trabajo espacial y corporal realizado en un tablero en el piso, con otra escala de trabajo más pequeña: un tablero de mesa. Es por eso, que espacialidad, lateralidad, razonamiento, numeración, figuras geométricas, oralidad: Narrativas y relatos, colores primarios, etc., son posibles de incorporarse en los recorridos dispuestos en los tableros, y que se vinculen a contenidos posibles de insertarse como parte de una secuencia de enseñanza. Para ello les ofrecimos una propuesta a modo de ejemplo que integre estos sentidos.

La propuesta que sugerimos para integrar el uso de tableros en el piso y el uso de robotita para la enseñanza de un eje transversal en el diseño curricular es el de educación ambiental. La

actividad se presentó con la finalidad de propiciar en las niñas y los niños la distinción de los objetos respecto de la basura, para distinguir los elementos orgánicos e inorgánicos que lo constituyen y promover acciones de separación de la basura y más adelante, de reciclaje. Aunque adherimos a la idea de que las acciones individuales no son las únicas a considerar para comprender la complejidad de los problemas ambientales, consideramos que la oportunidad de reconocer y desnaturalizar el destino de los objetos de consumo es una buena iniciativa en el recorrido por la enseñanza del ambiente en el nivel inicial. Entre las orientaciones para las docentes definimos:

- Momento 1: El paso de una escala a otra de la programación ¿Cómo se insertan los contenidos vinculados a la programación en esta propuesta?

Los contenidos de programación en un tablero sirven como medio para conceptualizar, distinguir y aplicarlos a situaciones específicas. En este caso es el tablero que se diseñará con imágenes de un contenido específico (en este ejemplo materiales orgánicos y reutilizables) permitirá resolver distintos recorridos de acuerdo con la complejidad e intencionalidad que orientará la docente. La programación aquí constituye una actividad concreta que permite trabajar sobre un aspecto concreto de una secuencia de implementación. Por ejemplo, en este caso, ayuda a los aspectos de definición de los objetos y posibles aplicaciones. Esto no significa que esta secuencia constituya o comprenda el trabajo en su totalidad: será necesario para este contenido continuar con la labor de desarrollo en función de los propósitos pensados. Por ejemplo, para este tema se puede trabajar sobre emprendimientos locales para el reciclado de basura, cómo podemos generar menos basura, un proyecto de reciclaje escolar etc. Es decir, la actividad de programación, y todas las variantes que veremos, se constituyen como partes del proceso de construcción de aprendizajes de contenidos de área, y en paralelo, se aprenden nociones de programación y manejo de dispositivos digitales. En una etapa posterior las tablets, a través del uso de Tux Paint, de la cámara de fotos o de Scratch, puede contribuir a una elaboración sobre lo aprendido.

- Momento 2: Presentación finalidades y objetivos de la secuencia sugerida para pensar y planificar propuestas que integran la programación y la robótica a los saberes en los campos de experiencias

Propósito: Distinguir los objetos de consumo que intervienen en el ambiente para conocer posibles acciones de intervención domiciliaria.

Objetivos: Reconocer los posibles usos y reutilización de materiales sólidos y de materiales orgánicos - Fomentar la creatividad mediante la construcción de tableros y juguetes con los objetos reciclables. Distinguir texturas, materiales y posibles acciones de recuperación o destino de los objetos de consumo. Propiciar habilidades espaciales, de abstracción y de

resolución de problemas.

- Momento 3: Posible secuencia integradora. Actividades.

La propuesta no tiene carácter prescriptivo. Se constituye en una orientación de posibles recorridos en la sala, entre los que mencionamos: la elaboración de un “Cesto” con la distinción de objetos de la basura (orgánicos, inorgánicos/ reciclables no reciclables) que permiten recuperar tipo de material, origen y destino. Con las botellas de plástico se pueden elaborar carritos, autitos, personajes que sirvieran de móvil en el recorrido de un tablero en el piso o en la mesa. Los tableros pueden incorporar imágenes u objetos tales como botellas de plástico, objetos de vidrio, papeles, cartones, cáscaras y alimentos para que las niñas y los niños diseñen recorridos con los autitos construidos bajo la consigna “el autito pasa a buscar los objetos reciclables” o a la inversa. Esto requería que previamente se habilite la instancia explicativa. Luego, esta actividad se trasladaría a un tablero con imágenes para que robotita sea la que circule por el mismo mediante la programación de sus pasos. Esta etapa, a posteriori permite la incorporación de otros recursos, estrategias y contenidos sobre el tema.

Finalmente, la última parte del segundo encuentro se orientó en el uso del programa ScratchJr. a partir de proponerles desafíos que permitan conocer el entorno, los bloques y su funcionamiento. Se cierra el encuentro proyectando posibles experiencias en las salas a partir de lo desarrollado.

La asistencia, la implementación y entrega de un trabajo final fueron los requisitos necesarios para acreditar el trayecto. Respecto al trabajo final, éste debió incluir el registro de implementación de las experiencias (registro visual y/o audiovisual), los contenidos trabajados, los dispositivos utilizados, y una evaluación de este (fortalezas y cuestiones a mejorar). Hemos conocido los trabajos de las docentes cuando compartieron sus registros en formatos video, fotografías y relatos. Pudimos observar que para la planificación de una secuencia de trabajo manifestaron la vinculación con los campos de experiencias del Diseño Curricular del Nivel Inicial de Río Negro (2019).

### **Las experiencias de las docentes en sus salas a partir de lo trabajado en los encuentros.**

Como hemos mencionado, las docentes seleccionaron aquellos contenidos digitales del trayecto de formación docente posibles de llevar a sus salas, teniendo en cuenta los contenidos o proyectos que se estuvieran desarrollando en sus jardines. A partir de estas propuestas presentadas por las docentes, constatamos que nuestra propuesta de formación promovió el uso de recursos digitales de las niñas y los niños y no sólo como recurso de la docente para presentar un contenido.

La mayoría de las docentes resaltan como atractivo para sus clases el uso de los dispositivos tecnológicos. Estas experiencias son coincidentes con los resultados de investigaciones previas con docentes del nivel inicial<sup>3</sup> del Alto Valle, en la que haremos alusión al artículo de Jara, M, Muñoz, M.E; Tirachini, A (2016) de la que se desprende

Las maestras sostienen que, inmersos en un mundo de expansión tecnológica, es necesario que los niños/as aprendan, en edades tempranas, a interactuar con el medio digital desde “otros lugares”, que no sean solamente el de la “recreación o juego libre” y que para ello es importante establecer finalidades claras con relación a los contenidos que se quieren trabajar (pp.441)

En nuestra experiencia, la posibilidad de que las docentes puedan experimentar con su grupo de niñas/os secuencias de enseñanza basadas en las orientaciones ofrecidas para el uso de los recursos digitales, promovieron el uso de recursos digitales. De un total de 555 participantes recuperamos alrededor del 60% de sus experiencias.

A modo de presentar las experiencias de las docentes, agrupamos los resultados siguiendo los siguientes usos en las que se incluyeron diversas maneras de agrupamiento de las niñas y los niños: en pequeños grupos (multitarea), en grupo total, de modo individual; siempre dependiendo de la propuesta pensada y sus propósitos.

### 1) **Sobre el uso de la pizarra digital**

Un grupo de docentes utilizan la pizarra digital para abordar los contenidos de los Campos de Experiencias. Para ello, elaboran actividades interactivas en las que han utilizado las orientaciones ofrecidas en el trayecto incorporando imágenes en programas que no requieren de conexión a internet, por ejemplo, Power Point o Impress. De esta manera las/os niñas/os utilizaron la pizarra digital para seleccionar y arrastrar la correlación de cantidades de elementos con números escritos, dibujar partes que faltan en una figura de una cara o del cuerpo humano. También utilizaron el arrastre de elementos para armar rompecabezas con animales, objetos y obras de arte. Otro grupo de docentes utilizó la función dibujar y animar el dibujo para crear animaciones de canciones, cuentos, poesías. De esta manera, se incorporan algunos recursos conocidos por las docentes como las imágenes, no sólo para “enseñar a mirar” sino también da cuenta que los/as niños/as exploran mediante los recursos digitales saberes en los Campos de Experiencias. Estos acercamientos dialogan con los resultados la

---

<sup>3</sup> Artículo presentado como resultado de la investigación “Enseñar ciencias sociales en contexto de cultura digital” de la FACE UNCOma dirigido por Miguel Ángel Jara y codirigido por María Esther Muñoz.

misma investigación citada previamente, en donde el uso de imágenes por parte de las docentes del nivel inicial aparece como uno de los principales recursos para la enseñanza “porque permite: traducir símbolos verbales en visuales, transmitir sentimientos y actitudes, estudiar momentos de un proceso, recoger datos de segundo orden, que no se aprecian a primera vista, simplificar realidades complejas estudiar el pasado, establecer comparaciones, hacer visible lo alejado e invisible” (p.63)” (Menegazzo (en Santos Guerra, 1984, citado por Joselevich 2014) Hirtz, Terachini, Benavidez 2018 p.3)

### **Sobre la enseñanza de la programación y la robótica con elementos concretos**

Un grupo significativo de docentes recuperó las orientaciones ofrecidas en el trayecto sobre la iniciación a la programación con materiales concretos y el uso de robotita con alfombras. Estas tendencias de utilización han prevalecido por sobre la propuesta de uso de las tablets utilizando el programa ScratchJr. que ha sido seleccionada por un grupo menor de docentes. Esto indicaría que las docentes optaron por el uso de materiales concretos para trabajar las primeras aproximaciones a la programación y la robótica por sobre la propuesta utilizando las tablets. También se desprende de estas iniciativas la preferencia por la utilización de las tablets en una etapa posterior de trabajo. En este grupo, la vinculación con los contenidos de los Campos de Experiencias también es un aspecto relevante, porque buscan contextualizar las actividades para la iniciación a la programación de acuerdo al tema que están trabajando (las más recurrentes fueron Dinosaurios; Cuentos Tradicionales; Actividades de Rutina; Espacialidad; Número Escrito; etc.). Este grupo de docentes logran que las/os niñas/os también exploren los recursos digitales y se inicien en la Programación y Robótica.

Las docentes han hecho uso y adecuaciones para su grupo de niños/as de la sala en la propuesta de las tarjetas “adelante, atrás, girar a la derecha, girar a la izquierda” y del tablero en el piso para su grupo utilizando un material concreto extra como una tela atada en la muñeca o brazo (para diferenciar el lado izquierdo o derecho), ubicar en el espacio un punto de referencia para indicar dicha posición (una ventana, la puerta, una mesa, etc.). El acercamiento se posibilita jugando a “Simón dice”, “El semáforo”, “Estatuas”, en el que estos juegos tradicionales se constituyen como variables para el uso de las tarjetas para manifestar las acciones a seguir (en los juegos tradicionales mencionados las acciones se dan de manera oral o con movimientos del cuerpo). En otros casos los tableros en el piso se vincularon al recorrido con el cuentos tradicionales, donde los personajes que deberán transitar el tablero, como Caperucita Roja, se enfrenta a un obstáculo, que es el Lobo y la llegada será la casa de la Abuela. Otras docentes utilizaron el tablero para hacer una búsqueda de tesoro, para ello, un/a pirata fue el personaje que debía seguir el recorrido, el obstáculo fue creado por los/as niños/as y el lugar de llegada fue el tesoro. Otra docente utilizó el tablero para ubicar el recorrido que

hicieron del Jardín al Museo de Dinosaurios que visitaron, y para este caso, los obstáculos eran puntos de referencias que tuvo el trayecto. Vale aclarar, que para todos los casos, los personajes deben seguir el recorrido que “les han programado” los/as niños/as programadores, es decir, los personajes tuvieron que seguir indicaciones previamente armadas y para ello hacen uso de las tarjetas. Aquí también se ha observado el uso de los roles que fueron presentados en el trayecto (líder, programadores, ejecutor). En estas propuestas en las cuales predomina el uso del cuerpo y su movimiento son las más elegidas por las docentes para trabajar la programación, además, un par lograron compartir esta estrategia con los/as docentes de Educación Física.

El uso de tableros más pequeños para utilizar la Robotita propició el uso de la alfombra transparente, siguiendo la propuesta de trasladar lo realizado en el tablero grande al tablero más chico. Las docentes han manifestado que esta instancia con Robotita fue fundamental la estrategia del uso del dispositivo en pequeños grupos (multitarea), y que sea rotativo (ya sea en esa misma jornada o en otro día).

Respecto al uso de las tablets para programar con Scratch Jr., como se mencionó anteriormente, un grupo menor de docentes logró llegar a esta etapa. No obstante, manifestaron que fue del agrado de los/as niños/as y que se sorprenden de que ellos/as aprenden muy rápido la dinámica de programación. Aquí la limitación estuvo más que nada, por el tiempo de implementación, al ser la última etapa de las orientaciones brindadas para la “Didáctica de la Programación y Robótica”, no todas las docentes lograron desarrollarla.

## **Conclusiones**

El equipamiento y las propuestas de trabajo tuvieron recepciones muy positivas por parte de las docentes. Manifestaron interés por conocer, manipular y planificar actividades para sus grupos de sala. Reconocieron como muy positivas aquellas propuestas presenciales que permitan construir colectivamente los sentidos de la incorporación de los dispositivos digitales en sus clases, valorando que son saberes fundamentales para la formación de las niñas y los niños en el contexto de cultura digital.

Sobre la programación y la robótica el interés por conocer cómo las propuestas eran recibidas por los grupos de niñas y niños motivó la realización de experiencias cercanas a los encuentros de formación. La perspectiva situada de los trayectos, habilitó múltiples formatos de experiencias, donde es posible reconocer y desprender componentes axiológicos de las planificaciones de enseñanza. Las docentes manifestaron que en sus propuestas era posible trabajar nociones temporales, tales como las rutinas y hechos próximos, utilizando gráficos y tarjetas, y nociones espaciales a partir de la propuesta de tableros. En esos puntos es donde

manifestaron los diálogos que, mediante las propuestas de juego, las niñas y los niños pudieron ejercitar el razonamiento lógico y los aprendizajes sobre rutinas y nociones temporales y espaciales. Estos aspectos resultaron potenciales para trabajar con tableros y “Robotita”. Para estas propuestas, las docentes recrean cuentos y canciones, se disfrazan, generan interés a partir de juegos y búsquedas. Presentaron a “Robotita” incorporando nociones de cuidado y fue posible el trabajo en pequeños grupos, dividiendo el uso de los distintos dispositivos.

La diversidad de formatos de agrupamientos en el nivel inicial y la creciente incorporación de proyectos que incluyen las nuevas tecnologías, motivan profundizar el trabajo sobre otros aspectos, tales como la alfabetización crítica digital, el resguardo de documentos y la promoción de espacios de formación que habiliten compartir experiencias que fortalezcan los sentidos de la educación digital en el nivel inicial.

Más de quinientas docentes participaron entre el 2019 y el 2022 en los trayectos de formación. El universo de niños y niñas alcanzados por las experiencias que registraron las docentes movilizaron experiencias que incluyeron alrededor de diez mil niñas y niños. Entendemos que el entusiasmo y compromiso de las instituciones del nivel inicial por emprender otros recorridos en la enseñanza, motiva dar continuidad a nuevos trayectos de formación docente que fortalezcan la promoción de la formación situada, en servicio y de la promoción de las experiencias como motor de los cambios que buscamos como educadores.

### **Bibliografía**

- Barbará, M.P (2015). La alfabetización cultural en TIC (tecnologías de la información y la comunicación) En Origlio, F “La alfabetización cultural en la escuela infantil: hacia otra mirada de la educación inicial”. NOVEDUC. CAMUS Ediciones.
- Buckingham, D (2012). Más allá de la tecnología: aprendizaje infantil en la era de la cultura digital. 1era Ed. 1era reimpresión. Manantial. Buenos Aires.
- Consejo Federal de Educación. “Aprobación e implementación de “Núcleos de Aprendizaje Prioritarios para Educación Digital, Programación y Robótica” (NAP) Res. 342/18 disponible en <https://www.argentina.gob.ar/consejofederaleduccion/documentos/resoluciones/17/12/18>
- Hirtz, M; Tirachini, A; Benavidez, V (2018) “Valoraciones del profesorado del nivel inicial sobre el aprendizaje del conocimiento social con dispositivos digitales” en Libro de actas Jornadas Nacionales y VI Internacionales de Enseñanza de la Historia. II Jornadas Nacionales Red de Docentes e Investigadores en la Enseñanza de la Geografía. IV Encuentro Iberoamericano de Investigación en Didáctica de las Ciencias Sociales, San Carlos de Bariloche, Provincia de Río Negro, del 3 al 8 de octubre de 2018. Disponible en <http://rdi.uncoma.edu.ar/handle/uncomaid/15649>
- Jara, M; Muñoz M.E; Tirachini, A (2016) Ciencias sociales en la educación infantil en

entornos digitales: experiencias del profesorado. *Revista História Hoje*, v. 5, nº 9, p. 418-430 - 2016 Ministerio de educación y Deportes (2016) “Argentina Enseña y aprende” como anexo de la Resolución CFE N°285/16, en la página 5 cita a la Ley N° 26206, 2006, Art. 2 Ministerio de Educación y deportes (2019) Aprender Conectados Nivel Inicial Ministerio de Educación y deportes (2019) Navegación segura en internet Nivel Inicial

Ministerio de Educación y deportes (2019) Programación y Robótica para el Nivel Inicial

Ministerio de Educación y Deportes de la Nación, (2016) “Competencias de educación digital”, 1.a ed., Buenos Aires, 2016 Dirección de Educación Digital y Contenidos Multiplataforma, Educ.ar S. E.

Ministerio de Educación y Deportes de la Nación, (2016) “Marco nacional de integración de los aprendizajes: hacia el desarrollo de capacidades”

Ministerio de Educación y Deportes de la Nación, (2016) “Orientaciones pedagógicas” Ministerio de Educación y Deportes de la Nación. Secretaría de innovación y calidad educativa. (2016) “Indicadores de progresión de los aprendizajes”

Santisteban, A. (2011). Las finalidades de la enseñanza de las Ciencias sociales. En Santisteban, A. & Pagès, J. (2011). *Didáctica del Conocimiento del Medio Social y Cultural en la Educación Primaria* (pp. 63-83). Madrid: Síntesis

## **¿Cómo incorporan los dispositivos robóticos escolares los profesores del nivel primario y medio de la provincia de Río Negro? Resultado de un dispositivo de formación docente.**

Laura J. Peña, M. Eugenia Walker, Nicolás A. I. Ticac, Facundo Bárcena  
lpconectar.rn@gmail.com, ewconectar.rn@gmail.com, ntconectar.rn@gmail.com,  
bfconectar.rn@gmail.com

Subsecretaría de Innovación, Calidad y Planeamiento Educativo  
Ministerio de Educación y Derechos Humanos de la Provincia de Río Negro

### **Resumen**

Los trayectos de formación de programación y robótica destinados a docentes y Referentes Escolares TIC (RET) del nivel primario y secundario de la provincia, surgen a partir de políticas educativas nacionales que propiciaron la entrega de equipamiento a las escuelas en el marco de los programas “Escuelas del futuro” (2017-2019) y “Aprender Conectados” (2018-2019). Este equipo desde entonces ha desarrollado diversidad de trayectos de formación docente con el propósito de incorporar e incluir contenidos de educación digital, programación y robótica a las prácticas de enseñanza.

Los desafíos que implicó la planificación de propuestas de formación docente para que efectivamente se concreten en las aulas, se vinculan fundamentalmente a dos aspectos. En primer lugar, reconocer la enseñanza y el aprendizaje de la programación y la robótica como saberes que forman parte de la cultura y ciudadanía digital del mundo actual. Por otro lado, la generación de un espacio en la planificación docente, donde estos contenidos no queden excluidos o separados de la transversalidad que implica pensar las nuevas tecnologías en relación a los saberes de los espacios curriculares. De este sentido se desprende la necesidad de superar una mirada instrumental del uso de los programas y dispositivos, en tanto se conviertan en recursos que fortalezcan los aprendizajes de programación y robótica y promuevan el desarrollo de competencias digitales a partir de la resolución de problemas propios de las disciplinas curriculares o en un sentido más amplio, a partir de proyectos institucionales e interdisciplinarios.

En esta presentación nos ocuparemos de caracterizar las estrategias de formación docente y los resultados de las propuestas didácticas de los docentes participantes, cuando utilizan dispositivos robóticos para enseñar contenidos curriculares en diálogo permanente con los saberes de la Educación Digital. Las preguntas que orientaron estas decisiones fueron: **¿Qué contenidos requieren conocer los docentes para incorporar una mirada transversal de la programación y la robótica? ¿Qué estrategias didácticas favorecen su inclusión en la planificación de la enseñanza? ¿Cómo vincularlas a los saberes disciplinares propuestos en el diseño curricular de la escuela primaria y secundaria rionegrina?**

La experiencia de formación demostró que el profesorado muestra interés por conocer primero el funcionamiento de los dispositivos robóticos, para poder pensar cómo

lo implementarían en sus clases. En segundo lugar, cuando incorporan conocimientos de programación y robótica educativa, pueden establecer estrategias de enseñanza en sus clases y requieren de acercamiento interdisciplinario sobre el conocimiento escolar para poder pensar proyectos y secuencias aplicadas a contenidos curriculares.

**Palabras clave:** Programación y Robótica Educativa, Formación Docente Situada, Estrategias de Formación, Nivel Primario y Secundario, Programación y Robótica Vinculada a Contenidos Curriculares.

## **Introducción**

En esta era digital, la educación se ha convertido en un campo en constante evolución, donde la incorporación de saberes emergentes desempeña un papel crucial en la formación de las futuras generaciones. Promovemos una perspectiva integradora de los recursos digitales en la enseñanza, para superar el mero uso instrumental de los recursos y favorecer la elaboración de estrategias didácticas que promuevan la resolución de problemas y se integren a los proyectos institucionales.

Es por eso que nuestra experiencia se centrará en los desafíos, logros y resultados de este dispositivo de formación, relatando qué, cómo y para qué los docentes de Río Negro han incorporado los contenidos propuestos como una herramienta educativa. En este artículo nos centraremos en los trayectos de formación desarrollados durante el año 2022.

## **Una referencia técnica del equipamiento entregado por el programa nacional**

El equipamiento entregado contiene a los dispositivos Q-Scout denominado kit Matebot para las escuelas primarias y kit Emotibot para las escuelas secundarias. Son robots de la empresa Robobloq que cuenta con diferentes componentes tales como un sensor ultrasónico de distancia, luces de color programable (RGB), sensor seguidor de línea y zumbador (buzzer) incorporado a la placa. El dispositivo Emotibot cuenta también con un sensor de sonido y una matriz led para generar mensajes y figuras. Poseen además, una estructura metálica a la cual se pueden anexar piezas de expansión como vigas, placas y más sensores. Las rutinas de programación se generan a partir del entorno de programación multiplataforma MyQode, basado en Scratch y compatible con Arduino. Este utiliza la forma de programación en bloque para que se realicen dichas programaciones y se carguen a los procesadores de los dispositivos.

## **Desafíos al pensar el trayecto de formación para los y las docentes de Escuela Primaria y Escuela Secundaria**

La planificación de los trayectos de formación docente para los niveles primario y secundario

tuvieron en cuenta los Núcleos de Aprendizaje Prioritarios de educación digital, programación y robótica (NAP EDPR, 2018), cuyo documento sostiene:

El aprendizaje de la educación digital, la programación y la robótica se enmarca en un proceso de transversalidad. De este modo, se pretende habilitar de forma creativa la generación de proyectos originales y diversos, que puedan estar relacionados con las problemáticas de las comunidades educativas, las economías regionales y otros aspectos socioculturales relevantes (p.10).

Entre los NAP de ED se recuperan para nivel Primario “la creación y el uso de juegos y diversos recursos en los que se utilicen conocimientos sobre los principios básicos de la programación física y la robótica, incluyendo las dimensiones de diseño, construcción, operación y uso” y “el diseño, la construcción y la depuración de secuencias de programación y robótica para desarrollar proyectos orientados a resolver problemas en el hogar, la escuela y la comunidad, a partir del uso de estructuras simples de código que involucren la utilización de variables y distintos formatos de entrada y salida de datos. (p-17 a 20)

En el nivel medio los ejes centrales se constituyen en la resolución de problemas “a partir de su descomposición en partes pequeñas, aplicando diferentes estrategias, utilizando entornos de programación tanto textuales como icónicos, con distintos propósitos, incluyendo el control, la automatización y la simulación de sistemas físicos” y “la aplicación de sus habilidades analíticas, de resolución de problemas y de diseño para desarrollar proyectos de robótica o programación física, (...) construyendo soluciones originales a problemas de su entorno social, económico, ambiental y cultural” (p.6)

Estos núcleos representaron diferentes desafíos al momento de planificar los trayectos de formación para cada uno de los niveles, no sólo por la complejidad de los contenidos. Desde nuestra experiencia en territorio<sup>1</sup> conocemos que la mayoría de los y las docentes, en su formación profesional de base, no acceden a los saberes propios de la programación y la robótica educativa, y por ende de su abordaje pedagógico y didáctico. Además, identificamos que entre ellas y ellos circulan representaciones acerca de las nuevas tecnologías, las cuales suelen obstaculizar su incorporación en las prácticas de enseñanza. Por ejemplo, considerar que para enseñar programación y robótica el/la docente debe ser experto en dicha área. Esto definió la necesidad de considerar que la propuesta formativa debía estar orientada a favorecer el acercamiento y familiarización de los docentes con los entornos de programación, los

---

<sup>1</sup> Desde el año 2017 el equipo de formación de PLANIED RN lleva adelante instancias de acompañamiento y formación docente con la finalidad de promover la alfabetización digital, la programación y la robótica; en el marco de la entrega de diversos dispositivos digitales en los niveles educativos inicial, primario y secundario. En esas instancias se elaboran materiales originales y adecuados a cada nivel y su especificidad.

dispositivos digitales y robóticos.

Para la elaboración de los materiales, tomamos como punto de partida las “Guías didácticas” elaboradas por los equipos nacionales y que acompañaron la entrega de equipamiento en las escuelas, los materiales facilitados por la fundación Sadosky y la iniciativa “Program.ar”. Las mismas orientan sobre el uso de los dispositivos y proponen actividades para utilizarlos en el aula, con el propósito de preparar a los y las estudiantes para el uso crítico, creativo y seguro de la tecnología.

A partir de éstos documentos, el equipo provincial estructuró y realizó las adecuaciones de los contenidos de acuerdo a la realidad de las escuelas rionegrinas y en relación al tipo de equipamiento con el que contaban, tales como incorporar en las propuestas los proyectos que las escuelas primarias y secundarias generan; alimentación saludable, cuidado del ambiente, proyectos de sistematización de huertas, ciudadanía digital, entre otros.

Por su parte los diseños curriculares definen sentidos de los contenidos de ED. El diseño curricular del nivel primario sostiene: “Cuando nos planteamos utilizar las TIC como herramienta pedagógica, estamos hablando de utilizar las tecnologías para potenciar la enseñanza de las áreas curriculares.” (D. C. Nivel Primario - 2011, pág. 37). Es decir se debe promover procesos de construcción significativos que potencien el desarrollo de las habilidades lingüísticas, comunicativas, sociales, lógicas y computacionales de las y los estudiantes, respetando sus tiempos y modos de aprendizaje. La inclusión de la educación digital, la programación y la robótica educativa, se constituyen en herramienta pedagógica en tanto permiten ampliar y complejizar el campo de saberes de las y los estudiantes, acercándolos a la cultura digital de nuestra actualidad, de una manera reflexiva y de uso responsable.

En cuanto al diseño curricular de la escuela secundaria rionegrina, uno de los pilares que sostiene la propuesta educativa, define “Educar en las múltiples y emergentes alfabetizaciones” (D.C ESRN - 2018, pág.34). Esto implica que los y las estudiantes deben apropiarse de un conjunto de saberes que abarcan las diferentes áreas de conocimientos, entre ellas, los contenidos digitales, ya que son considerados temas que constituyen una respuesta a los problemas actuales y que le demandan a la Educación una atención prioritaria. Particularmente, si se analizan los diferentes campos o áreas de conocimiento, podemos encontrar que el uso de la tecnología nuevamente se vincula con los saberes curriculares a partir de la resolución de problemas reales y contextualizados. Las propuestas pedagógicas deben estar orientadas a usar la tecnología para innovar, motivar y enseñar a reconocer cómo influyen en la organización social, en la realidad socio - ambiental y en la construcción de subjetividades, permitiendo que sean utilizadas para procesar información, comunicarla y visualizarla según los contenidos a tratar.

Enfatizando sobre la relevancia de la transversalidad de las nuevas tecnologías, la convocatoria fue abierta a docentes de todas las áreas curriculares y a RET de las escuelas participantes, fomentando el trabajo interdisciplinar en las instituciones para la construcción de propuestas en el aula.

Un último desafío fue crear e impulsar una propuesta de formación para docentes que implicaba -además de enseñar a utilizar los dispositivos y sus entornos de programación- realizar el acompañamiento en la construcción de una propuesta de enseñanza vinculadas a lo abordado y su puesta en práctica en aula. Para ello gestionamos la declaración de interés educativo del proyecto, con reconocimiento de puntaje docente, considerando relevante la práctica en servicio y presencial.

En suma, para cada nivel educativo se propusieron instancias de formación diferenciadas. En el caso del nivel primario, el trayecto se denominó “Saberes emergentes en educación digital para el tercer ciclo de la escuela primaria”, el cual se organizó en cuatro ejes: “Programación educativa”, “Robótica educativa”, “Propuestas educativas para la enseñanza de sensores digitales y físicos” y “Elaboración de proyectos para la ciudadanía digital responsable”. Cada eje tuvo una duración de 4 semanas: tres clases virtuales, una clase por semana y luego un encuentro presencial. Se establecieron sedes en las cuales se agruparon localidades cercanas entre sí, cada una con su propio espacio virtual, y también sus referentes pedagógicos, tutores y facilitadores técnicos. A tal fin se seleccionaron las escuelas primarias que tenían los dispositivos digitales afines a la propuesta; favoreciendo a dos docentes de tercer ciclo y un RET de cada una<sup>2</sup>. Las aulas virtuales, presentes en la plataforma del ministerio de educación de Río Negro, dieron acceso a las y los docentes a los materiales (tutoriales, documentos, bibliografía, entre otros) y a las actividades obligatorias propuestas en los foros. Para la acreditación y aprobación de cada eje del trayecto se solicitaba haber realizado y aprobado las actividades obligatorias de por lo menos dos clases virtuales y asistir al encuentro presencial. Cada encuentro presencial se pensó y constituyó como espacio de socialización de experiencias, inquietudes y logros; conllevando a la reflexión sobre la construcción de futuras propuestas de enseñanza que vinculan los contenidos curriculares con los de programación y robótica.

Los contenidos desarrollados durante los cuatro ejes fueron: algoritmos; programación con bloques con Scratch: selección y edición de objetos y fondos, coordinación y sincronización entre objetos, enviar y recibir mensajes, movimiento, apariencia, sensores digitales, condicionales, entre otros; programación con Q-Scout para Matebot: trayectorias simples,

---

<sup>2</sup> Según la normativa Res. 233/98 la licencia Art. 25° respecto a las capacitaciones docentes, otorga autorización para su usufructo al 20% de plantel docente por institución.

encendido de luces y sonido, combinaciones y sensor ultrasónico; ciudadanía digital.

En tanto, para el nivel medio planificamos dos encuentros presenciales en diferentes localidades sede<sup>3</sup> de la provincia, de 4 horas cada uno y que requería para su acreditación la planificación y/o puesta en práctica de una actividad con estudiantes. Las instancias de acompañamiento se dieron de forma presencial en cada encuentro con los diferentes perfiles pedagógicos y técnicos. Se pudieron abordar las dificultades e inquietudes de los y las docentes, en diferentes momentos se habilitaron espacios de debate y discusión acerca de las actividades propuestas y de su incorporación en las prácticas de enseñanza. Además se generaron documentos y video tutoriales con la información ofrecida en cada encuentro, no sólo de aspectos técnicos y del uso de los diferentes elementos del Emotibot, sino también con sugerencias didácticas y de gestión de la clase.

Los contenidos y materiales de la formación para el nivel medio, se secuenciaron en una primera instancia para reconocer las partes del robot y el entorno de programación: bloques de acción, sonido, luz y condicionales. Luego avanzamos complejizando los desafíos de programación y ampliando los contenidos al uso de los sensores, matriz LED y sus respectivos bloques de programación.

En cada uno de los trayectos se propusieron actividades de programación para analizarlas como parte de una actividad áulica, es decir que se vinculaba con algunos contenidos disciplinares, con la intención de que los y las docentes puedan planificar propuestas de enseñanza que permitan innovar en sus clases, no sólo desde la incorporación instrumental del dispositivo sino también porque el trabajo con programación y robótica propone una dinámica de clase que fomenta el trabajo colectivo, cooperativo y por roles.

### **Estrategias de formación docente**

Una estrategia de formación con los entornos de programación y los dispositivos robóticos fue plantear las actividades a partir del reconocimiento de su funcionamiento y la exploración de los mismos. Estas tareas se realizaron a partir de desafíos de programación y de sus recreaciones -cada vez más complejos- que permitieron construir los primeros saberes, constituyendo a los trayectos de formación en espacios de oportunidades para la experimentación, el diálogo, la colaboración y la creación de nuevas resoluciones, habilitando situaciones significativas de aprendizaje.

En el nivel primario, una estrategia fundamental fue la secuenciación de los contenidos y las actividades a partir de instancias virtuales y presenciales, en los cuatro ejes antes mencionados.

---

<sup>3</sup> Localidades sede: Cipolletti, Cinco Saltos, El Bolsón, San Carlos de Bariloche, Villa Regina, General Roca, Viedma, Sierra Grande, Choele Choel, Catriel, Allen y Río Colorado.

Esta organización permitió realizar un abordaje paulatino y cada vez más complejo respecto a la programación y la robótica; pudiendo hacer el acompañamiento y dar las orientaciones requeridas por las y los participantes.

Además, cada clase se planteó como un modelo no prescriptivo, a partir de la resolución de actividades, las cuales requerían que cada docente primero resolviera un desafío de programación como estudiante y, posteriormente, la re pensara como docente que enseña a programar.

Por ejemplo, en el eje “Programación Educativa”, una actividad de programación en papel utilizando tableros (grilla cuadrículada) e instrucciones (flechas) para ir de un punto a otro; habilitó el análisis y reflexión sobre qué contenidos podrían abordar con ésta, de qué manera, cuáles serían las readecuaciones que harían en cada caso, cómo la presentarían, cuál sería la finalidad, etc. Luego, y para potenciar este recurso se les propuso hacerlo en el suelo, de forma tridimensional y con la corporalidad, que les haría poner en juego nuevas estrategias y desarrollar otras habilidades.

Otra estrategia de complejización fue la exploración y resolución de desafíos con el entorno de programación Minecraft ofrecido por la plataforma code.org en su “hora del código”, logrando una primera aproximación al entorno visual, su funcionamiento y organización para resolver problemas, como así también con algunos recursos del portal program.ar. Posteriormente, conocer y utilizar el entorno de programación de Scratch -muy similar a los anteriores- permitió recrear programaciones e intervenirlas como también crear y depurar programaciones propias. Para lograr ese cometido, pusimos a disposición de las y los participantes tarjetas de programación, guías y tutoriales, lo cual nos permitió focalizar en los contenidos relevantes que permitieron avanzar en cada ejercicio.

En las clases del eje “Robótica educativa” desarrollamos actividades concernientes al reconocimiento del dispositivo robótico MateBot: los componentes del kit, el armado, funcionamiento y cuidados, la exploración del software MyQode y su puesta en marcha. Cabe destacar que el entorno de programación tiene una lógica y organización similar al de Scratch, pero requirió realizar el reconocimiento de los bloques que hacen funcionar los motores. Las primeras programaciones fueron de trayectorias simples y el reconocimiento de cómo funcionan según los bloques de programación; posteriormente se avanzó con programaciones de luces led, sonido, y distintas combinaciones de cada una. En tanto, los ejercicios y desafíos de robótica consistieron en dinámicas de trabajo colaborativos y de roles, en los cuales los docentes se organizaban en grupo -generalmente por escuela- y cada docente llevó adelante una parte de la tarea de forma consensuada con sus pares. Posteriormente, hacían que el robot ejecutase la programación; luego cada grupo analizaba y socializaba cómo se había ejecutado,

si era lo que esperaban o si existía algún fallo, en ese caso revisaban esa programación y la modificaban hasta cumplir el desafío.

En el eje “Propuestas educativas para la enseñanza de sensores digitales y físicos”; en la primera clase, la actividad se centró en la reflexión sobre la enseñanza de la programación y la robótica en el tercer ciclo del nivel primario, a partir del análisis de una secuencia de actividades y a la luz de los conceptos vertidos en la investigación de Karen Brennan; respecto a las implicancias de la enseñanza de la programación con Scratch en las niñas y los niños.

En las subsiguientes clases abordamos la incorporación de sensores y condicionales en la programación con Scratch y la programación del sensor ultrasónico en MateBot. En este contexto, las y los docentes comenzaron con la construcción metodológica de propuestas de enseñanza vinculando los contenidos curriculares de sus áreas con la programación y la robótica. Algunos realizaron trabajos interdisciplinarios con colegas de la misma institución.

Por último, durante el desarrollo del eje “Elaboración de proyectos para la ciudadanía digital responsable”, las y los docentes llevaron adelante la implementación de las propuestas de enseñanza elaboradas en la instancia anterior. Se les solicitó que realicen un registro audiovisual y relatoría de la experiencia a fin de compartir y socializar en los espacios compartidos sus reflexiones, inquietudes, desafíos y progresos.

Como cierre del trayecto, se abordó la elaboración proyectos para la ciudadanía digital responsable a partir de un dossier de actividades y recursos que fueron analizados por los presentes.

Previo al primer encuentro del trayecto “Robótica para el nivel Medio I” les solicitamos a los y las participantes que asistan con el kit armado. Para ello cada escuela contaba con el manual del dispositivo y les facilitamos vía correo electrónico material guía para el armado del prototipo denominado “Captain Alloy”. La sugerencia fue que realicen esta tarea en conjunto, en la medida de las posibilidades de los y las docentes, con la intención de fomentar desde la práctica el trabajo en equipo y colaborativo, esperando que esa dinámica se replique posteriormente en el aula. Además, se iniciaba con la posibilidad de que sean los propios docentes quienes gestionen el recurso en su escuela, conociendo los protocolos de préstamos y cuidados, establecidos y convenidos entre el equipo provincial y la dirección de nivel.

En la instancia presencial, las primeras actividades consistieron en la familiarización con las partes del robot, el entorno de programación y en algunas programaciones de prueba que permitieron desde la observación comprender cómo funciona el robot y la programación. En tal sentido, se propusieron desafíos en los que entraban en juego los contenidos trabajados. En el momento de revisión, habilitamos el espacio para probar la programación, discutir sobre las

diferentes formas de resolver el desafío, detectar los errores y encontrar las soluciones. Para comenzar a pensar en la vinculación con las áreas curriculares, presentamos una actividad ejemplo de trabajo con el robot vinculada con el área de Educación Matemática a fin de analizar y diferenciar los contenidos involucrados de cada una de las disciplinas. En ese momento fue importante dialogar acerca de la organización de la clase: planificación previa junto al RET, gestión de la clase y posibilidades de expandir la secuencia didáctica. La tarea final del encuentro consistió en que cada docente pueda esbozar una planificación de un actividad para trabajar con sus estudiantes y que se vincule con un contenido curricular. En el momento del plenario expusieron sus ideas e intercambiaron sugerencias que enriquecieron las propuestas pedagógicas; también se identificaron los desafíos y las fortalezas que implica incorporar los contenidos de programación y robótica en las aulas.

El segundo trayecto “Robótica para el nivel medio II”, comenzó con un breve repaso de los contenidos trabajados en el primer módulo. El propósito de este encuentro era el de trabajar con los sensores (ultrasónico, de sonido y sigue línea) desde su conexión y programación, para luego avanzar hacia la vinculación con los diferentes espacios curriculares y ampliar algunos aspectos didácticos, como por ejemplo la dinámica de roles, como posibilidad de trabajo en el aula con el dispositivo. A tal fin, la propuesta fue resolver diferentes desafíos de programación involucrando los sensores del robot. La dinámica de resolución fue grupal, conformando equipos de trabajo e identificando los diferentes roles: programador, responsable de materiales, líder de equipo, entre otros. En la puesta en común se analizaron los resultados, los errores y las posibilidades de vinculación con contenidos curriculares. En esta instancia se logró identificar el uso de los sensores para estudiar su funcionamiento y también al comprender su operatividad, la posibilidad de modelizar situaciones de la vida cotidiana, por ejemplo, sensores de incendios, detectores de movimiento, alarmas perimetrales, circuitos, vehículos de rescate, entre otros. Por otro lado, la invitación a los y las docentes fue a continuar pensando qué puede aportar la programación y la robótica al estudio de un contenido curricular, por lo que propusimos ejemplos, como los de las figuras 1 y 2.



Figura 1. Posibles vinculaciones con áreas de conocimiento



Figura 2. Aportes desde la robótica al estudio del sonido

Finalmente, los y las docentes expusieron sus experiencias de implementación y las apreciaciones finales del trayecto de formación realizado.

### Los resultados de las propuestas didácticas

La implementación de propuestas de enseñanza de programación y robótica en cada nivel fue fundamental para inaugurar las experiencias de trabajo, abordando dificultades, inquietudes y logros a partir de la socialización de estas, con el acompañamiento de tutores y RET.

En el nivel primario, la implementación de las propuestas de enseñanza recuperaron la programación en soporte papel, la organización lógica de secuencias y textos, el seguimiento de instrucciones para hacer un barco de papel, la programación con Minecraft y Scratch y el uso de Matebot. Las y los docentes readecuaron algunas de las actividades, desarrolladas en el trayecto, a los contenidos y características de sus grupos de estudiantes, originando una vasta variedad de recursos y ejercicios. Por ejemplo, una pareja pedagógica vinculó la programación con tableros e instrucciones con la temática de alimentación saludable. El desafío consistió en que los y las estudiantes debían crear un código (instrucciones) para que otro compañero recorriera físicamente el tablero y recolectara determinados alimentos saludables. Con esa actividad se buscó promover que los estudiantes desarrollen habilidades del pensamiento lógico y algorítmico, ya que debían anticipar qué instrucciones y en qué orden serían las adecuadas para lograr superar el desafío, mientras ponía en juego los saberes disciplinares trabajados en clases anteriores, clasificando y seleccionando alimentos.

Otras docentes llevaron adelante una experiencia de programación, en este caso con Scratch. Ellas propusieron que sus estudiantes creen una mascota virtual, para lo cual utilizaron los tutoriales facilitados en el trayecto. Las y los estudiantes, en el entorno Scratch escogieron un escenario, una mascota y distintos objetos (comida, agua, etc.) para crear un juego interactivo. Cada vez que el usuario dirige con los comandos (flechas del teclado) a la mascota hacia un objeto este desaparece o cambia de apariencia; entonces si una mascota se acerca a un cuenco con alimento, ese cuenco desaparece de escena o aparece vacío. Con esta actividad, las y los estudiantes comenzaron a explorar el mundo de la programación de los videojuegos, comprender la lógica detrás de cada bloque y sus combinaciones, situándolos en el lugar de creadores. Las docentes previamente habían trabajado sobre la coherencia textual, y la importancia del orden de las oraciones y párrafos para que un texto fuera comprensible. Dieron a sus estudiantes secuencias de párrafos desordenados que debieron ordenar para descubrir el contenido de un artículo. Cuando sus estudiantes comenzaron a programar con Scratch, retomaron la importancia del orden y la coherencia en las instrucciones para que el programa ejecute correctamente la programación deseada. Esa vinculación de contenidos permite establecer relaciones cada vez más complejas entre distintos temas, asociar ideas y explorar posibilidades, a medida que se busca fortalecer habilidades lingüísticas, sociales y algorítmicas.

Otros grupos de docentes crearon programaciones interactivas (trivias de preguntas y respuestas) que funcionaron como autoevaluaciones de un contenido, y que motivaron a los y las estudiantes a que exploren el entorno Scratch a partir de la interacción propuesta. Dentro de los contenidos que trabajaron los y las docentes pueden observarse: polígonos, múltiplos y divisores, cálculos, ciclo del agua, estados de la materia, cuentos maravillosos, tipos de textos (recetas e instrucciones), entre otros.

En el caso de robótica, los y las docentes optaron por actividades exploratorias y de reconocimiento del entorno de programación para Matebot, las funciones de los bloques de programación y su combinación, para realizar programaciones de base para futuros desafíos más complejos; como por ejemplo, que el robot se desplace hacia adelante y hacia atrás, gire, encienda las luces. En esa instancia establecieron dinámicas de trabajo grupal, donde cada estudiante tenía un rol: entre todos crearon el código, uno estaba encargado de introducir el código, otro lo ejecutaba, y luego realizaban el análisis de dicha ejecución. Esos roles se alternaban, para que todos pudiesen tener la experiencia de cada momento y centrarse en esa responsabilidad.

En tanto, en el nivel medio se observó que los y las docentes incorporan el uso del dispositivo robótico a partir de diferentes instancias de apropiación. Los y las docentes se sienten seguros replicando la misma dinámica en el aula que la que transitaron en el trayecto de formación. De

ésta manera, presentan el dispositivo con una actividad sencilla de trayectoria, que permita la exploración y familiarización con el entorno de programación. Por ejemplo, avanzar de un punto A a un punto B o realizar la trayectoria que forme un cuadrado, lo que implica poner en juego aspectos básicos del uso del robot: conexiones y bloques de programación. En un segundo momento, se inicia la vinculación con el contenido curricular. Las experiencias nos muestran que el uso del robot sirve para abordar y afianzar algunos contenidos. Tomando como ejemplo actividades vinculadas con física y matemática, los y las estudiantes lograron interpretar gráficos y fórmulas que modelizan variaciones lineales en función del recorrido que realiza el robot y analizar las relaciones entre las variables (tiempo, distancia y velocidad) en diferentes marcos de representación para el estudio de un mismo objeto matemático. Para este caso el uso del dispositivo sirvió para experimentar diferentes trayectorias y sus variaciones, registrando lo observado en una tabla, y a partir de allí avanzar hacia la modelización matemática de la situación propuesta (representación gráfica).

Las propuestas de formación alcanzaron a 109 docentes del nivel medio y a 115 del nivel primario de la provincia de Río Negro. Estimamos que durante el desarrollo de los trayectos los/as docentes implementaron el uso de los dispositivos digitales en sus clases, alcanzando al menos a 600 estudiantes en el nivel medio y 2700 estudiantes en el nivel primario.

### **Conclusiones finales**

Durante los trayectos de formación, cada participante tuvo el espacio para compartir sus progresos, inquietudes y dificultades, generando un diálogo con pares docentes a partir de sus experiencias, como así también para recibir los aportes de otros, enriqueciendo el proceso de aprendizaje y apropiación de los saberes. La interacción entre grupos, el intercambio de sugerencias y observaciones, potenció la instancia de programación, poniendo en valor el “error” como parte del aprendizaje; instancia que sería relevante a la hora de trabajar con sus estudiantes en el salón de clases con dinámicas similares. Cabe destacar, que las y los docentes participantes de los trayectos, de ambos niveles educativos, y que realizaron la implementación de estas propuestas innovadoras, expresaron que sus estudiantes se mostraron interesados, atentos y motivados en cada momento, que pudieron compartir y socializar sus progresos de forma significativa, creando un ambiente ameno de aprendizaje. Los y las estudiantes en el salón de clases experimentaron con mucho entusiasmo cada desafío y actividad; su curiosidad y creatividad los animó a continuar explorando los entornos de programación. En tal sentido, se destaca que el trabajo colaborativo tuvo un rol muy importante, permitiendo expresar sus puntos de vistas, intercambiar opiniones, tomar decisiones y superar los desafíos propuestos por sus docentes.

Como equipo de formación, consideramos que las actividades que se propusieron tuvieron la

intención de situar a los y las docentes como estudiantes, con el fin de poder enfrentar los mismos obstáculos que puedan surgir en el aula y darles orientaciones y herramientas para su abordaje. Esta decisión fue valorada por los y las participantes ya que lograron realizar experiencias de implementación con sus estudiantes, superando las representaciones que tenían sobre la incorporación de los dispositivos en las clases. Entendemos que estas propuestas de abordaje de la programación y la robótica en el contexto educativo son el puntapié inicial para nuevos y más aprendizajes que deben continuar fortaleciéndose en las escuelas, no sólo en las clases disciplinares sino a partir de proyectos interdisciplinarios que se sostengan en el tiempo.

Es por ello que el compromiso como equipo provincial es continuar acompañando en territorio a los y las docentes en el uso e incorporación de los dispositivos digitales disponibles en las planificaciones curriculares. Además consideramos necesario poder realizar una investigación que nos permita abordar y analizar el impacto de la formación en los procesos de enseñanza y aprendizaje, y que comparándolas con experiencias en contextos similares nos permita reorientar y fortalecer nuestras propuestas.

### **Bibliografía**

- Areces, C., Benotti, L., Cortez Sánchez, J., Fervari, R., García, E., Gómez, M., Martínez, M., Onetti, C., Rodríguez Pesce, E. y Wolovick, N. (2018) Ciencias de la computación para el aula : 2do. Ciclo de primaria : libro para docentes. - 1a ed . - Ciudad Autónoma de Buenos Aires. Fundación Sadosky.
- Brennan, K (2021) How kids manage self-directed programming projects: Strategies and structures, *Journal of the Learning Sciences*, 30:4-5, 576-610
- Consejo Federal de Educación (2013). Res. 137/13. Diseño curricular del Ciclo Básico y Ciclo Orientado de las escuelas técnicas.
- Consejo Federal de Educación (2018). "Núcleos de Aprendizaje Prioritarios para Educación Digital, Programación y Robótica" (NAP) Res. 342/18. Disponible en <https://www.educ.ar/recursos/150123/nap-de-educacion-digital-programacion-y-robotica>
- Consejo Provincial de Educación de la Provincia de Río Negro. (2011). Resolución 2135/11: Diseño Curricular del Nivel Primario. Disponible en [https://educacion.rionegro.gov.ar/files/nivel\\_primario/disenio\\_nivel\\_primario.pdf](https://educacion.rionegro.gov.ar/files/nivel_primario/disenio_nivel_primario.pdf)
- Czemerinski, H. Leonardi, C, Klinkovich, V, Miller, I, Palermo, A, Rodríguez, L; Andújar, L, Schaab, J, Ganem, T, Bonello, M. y Schapachnik, F (2018) "Ciencias de la computación para el aula : 1er. ciclo de primaria : libro para docentes"; 1a ed . -
- Ciudad Autónoma de Buenos Aires : Fundación Sadosky. ISBN 978-987-27416-6-2  
Ministerio de Educación y Deportes (2016) "Competencias de educación digital", 1.a ed.

Buenos Aires, 2016 Dirección de Educación Digital y Contenidos Multiplataforma, Educ.ar S. E.

Ministerio de Educación y Deportes de la Nación, (2016) “Orientaciones pedagógicas”  
Ministerio de Educación y Deportes (2017) “Guía didáctica Escuelas en RED”.

Ministerio de Educación y Derechos Humanos de la provincia de Río Negro (2017).

Resolución del CPE 947/17: Diseño Curricular del Ciclo Básico y Ciclo Orientado de las ESRN.

**Robótica educativa con hardware y software libre como herramientas  
para la enseñanza de programación.  
Una experiencia en bibliotecas populares y espacios no formales de aprendizaje**

Valentín Basel

valentinbasel@gmail.com

CIECS - UNC - CONICET

## Resumen

El uso de kit de robótica educativa en las escuelas primarias y secundaria de la República Argentina se ha popularizado en los últimos cinco años, principalmente gracias a la aprobación de los **Núcleos de Aprendizaje Prioritario (NAP) de Educación Digital, Programación y Robótica** en el año 2018, dando al surgimiento de distintas propuestas de equipos principalmente comerciales, adquiridos por el estado o por entidades particulares, que tratan de dar cuenta de los requerimientos necesarios para su utilización en el ámbito escolar. Sin embargo, al tratarse dispositivos físicos atravesados por lógicas no siempre concomitantes con las necesidades escolares, y en muchos casos caros y complejos de mantener, nos enfrentamos al dilema, y la oportunidad, de pensar alternativas que den cuenta de una mirada soberana sobre las tecnologías aplicadas a la educación, poniendo en foco las necesidades del sistema educativo de Argentina.

**Palabras claves:** Software libre, hardware libre, domótica, python, micropython

## Introducción

Los sistemas electrónicos de robótica educativa tienen su génesis en el trabajo de Seymour Papert y el Lenguaje LOGO. Es interesante atender al desarrollo del lenguaje LOGO, dado que es uno (y probablemente el primero) de los intentos de pensar una herramienta de software y hardware específica para la enseñanza con niños y niñas con una perspectiva pedagógica fundamentada. Podemos ver que las primeras experiencias de LOGO se remontan al año 1967. Desarrolladas por parte de Wally Feurzeig, Cynthia Solomon y Seymour Papert. (Feurzeig et al., 1970), estas experiencias tenían la iniciativa de crear un lenguaje de programación específico para la enseñanza de matemáticas mediante la programación (Papert, 1987; Solomon & Papert, 1976). En ese entonces, el problema de LOGO radicaba precisamente en la falta de hardware barato capaz de implementar las capacidades del lenguaje.

Al ser un lenguaje orientado a permitir a los estudiantes dibujar formas geométricas parametrizadas mediante “primitivas” de LOGO, se diseñó para su uso un robot con forma de tortuga (de ahí que LOGO use la imagen de una tortuga en sus versiones computarizadas) capaz de dibujar en papeles en el suelo. Hay que tener en cuenta que el primer robot LOGO no

era radio controlado, sino que simplemente tenía sensores y dos motores con los que se movía de manera autónoma, como los modernos robots que se utilizan en enseñanza de programación en la actualidad.

Sin embargo, al ser un equipo extremadamente caro y complejo de manejar, su utilización nunca pudo ser masiva. Solo recién con el advenimiento de los microordenadores (como el ZX spectrum o el Commodore C64) fue posible implementar una versión de LOGO que pudiera ser usada en las escuelas, pero el robot tortuga fue dejado de lado por una versión representada gráficamente en un televisor.

### **La robótica como herramienta para aprender a programar**

Tomando como base una perspectiva constructivista del conocimiento, y en especial el enfoque de Seymour Papert (1980), podemos afirmar que la robótica es una herramienta (Pittí et al., 2010) que permite a los estudiantes abordar la construcción de su propio conocimiento a partir del diseño y fabricación de un robot de carácter pedagógico. En este sentido, se puede decir que la robótica utilizada en el contexto educativo tiene múltiples aplicaciones como herramienta transversal en la enseñanza/aprendizaje de conceptos básicos de los lenguajes de programación. La implementación de robots como herramientas pedagógicas para la enseñanza de lenguajes de programación implica una serie de compromisos teóricos, pero también técnicos y logísticos específicos, desde la adquisición de herramientas para la fabricación y mantenimiento de los mismos, hasta la capacitación de los docentes para trabajar en el aula.

Dado que un robot es un componente físico y, por lo tanto, tiene un coste de adquisición y mantenimiento que puede ser elevado, las decisiones de costo/beneficio son fundamentales -aunque no exclusivas- a la hora de optar por utilizar la robótica en un colegio. Se puede considerar al software libre (Adell & Bernabé, 2007) y al hardware de especificaciones libres (González et al., 2003) como alternativas para aplicar estos métodos de enseñanza en el contexto del aula. Estas alternativas presentan ventajas pragmáticas (menor o hasta incluso nulo costo por licencias) que permiten ahorrar presupuesto y, por sus valores éticos, políticos y sociales (Hart, 2003), funcionan como disparadores de discusiones sobre los principios que una institución educativa tendría que promover. El software libre en la educación trata sobre la libertad de docentes y estudiantes, por lo tanto podemos decir que la aplicación de recursos educativos abiertos (Vidal Ledo et al., 2013) como filosofía para el desarrollo de una currícula específica de enseñanza de programación, es una opción más que interesante para implementar un sistema de enseñanza de programación usando robots educativos.

Los sistemas de robótica ofrecen herramientas interesantes para pensar la enseñanza de

programación (Basel, 2020; Letzen et al., 2019), su naturaleza física obliga a tomar decisiones de control sobre el hardware que vuelven más complejo el proceso de escribir el código fuente que controlará ese dispositivo (Barrera Lombana, 2015), siendo perentorio no solo tener en cuenta los objetivos propuestos para la actividad educativa (seguir una línea, resolver un laberinto etc.) sino, además, tener que constantemente corregir las desviaciones e imperfecciones de los sistemas mecánicos e imprevistos que puedan ocurrir. El uso de repeticiones, variables y estructuras condicionales se vuelve un proceso, si se quiere, “natural” cuando se programa un pequeño dispositivo electrónico con sensores y actuadores que, además, tiene la posibilidad de afectar el mundo real de los estudiantes y no únicamente meras simulaciones (Bağcı et al., 2017).

Por otro lado, es importante hacer una distinción entre “aprendizaje DE robótica” y “aprendizaje CON robótica”, entendiendo que la robótica es una disciplina en sí misma cuyo empleo en la industria tiene un nivel de complejidad que requiere un alto grado de especialización para técnicos e ingenieros que trabajan en ese ámbito. Por eso mismo, el aprendizaje CON robots debería ser pensado como un medio para facilitar la construcción de conocimientos transversales por parte de los estudiantes (Papert, 1993), adaptando las complejidades inherentes al diseño y construcción de un robot. Este aprendizaje CON promueve en los estudiantes el desarrollo de competencias tales como: la toma de decisiones basadas en el conocimiento, la formulación de explicaciones científicas y el trabajo en equipo (Rojas & Zúñiga, 2012), y como mencionamos, el aprendizaje de programación.

### **Robótica educativa libre**

Como dijimos en la sección anterior, los kits de robótica conllevan en su formación una serie de decisiones logísticas y pedagógicas que pueden exceder las necesidades comerciales de una industria que desarrolla componentes de robots educativos. Como ejemplo, podemos ver el desarrollo de kits de robótica educativa propuestos por la empresa LEGO, que en vez de usar un tipo de conector estándar RJ12<sup>1</sup> de fácil adquisición y armado con pinzas crimpadoras<sup>2</sup> de equipos telefónicos, utilizan un conector RJ12 de desarrollo propio donde el “clip” de seguridad que sostiene la ficha, típicamente en el centro de la pieza de plástico, está corrido a un costado. Esto hace que los cables de LEGO sean completamente incompatibles con la posibilidad de repararlos mediante cables telefónicos y fichas RJ12 comunes. Si se rompe una ficha, hay que cambiar todo el cable, pagando sobre costes.

Esta práctica de protección industrial no es solo aplicable a grandes empresas como LEGO, sino también a todo tipo de proyectos de robótica educativa que ocultan los diseños de su

---

<sup>1</sup> <https://es.wikipedia.org/wiki/RJ-12>

<sup>2</sup> <https://es.wikipedia.org/wiki/Crimpadora>

hardware, muchas veces derivados directos de ARDUINO, mediante carcasas de plásticos que no permiten acceder a la placa de circuito impreso como una estrategia de protección comercial? de sus productos, generando una dependencia directa de las instituciones para el mantenimiento y readecuación del hardware entregado. Como contrapartida a estas propuestas, la emergencia de la denominada cultura *maker* (Blackley et al., 2017; Dousay, 2017; Ludeña, 2019; Schrock, 2014) nos permite dar cuenta de alternativas “hágalo usted mismo”<sup>3</sup> en la producción de equipos de robótica educativa libres, entendiendo por libre a sistemas donde el diseño de los planos de los circuitos sea compartido mediante licencias de software libre, así como también el software de control y firmware embebido sea software libre.

La comunidad de desarrolladores de software libre plantea que el software, para ser considerado libre, debe poder ser copiado, estudiado, distribuido y modificado libremente por cualquier persona o comunidad (Stallman & Lessig, 2007). En este sentido, se vuelve de vital importancia contar con el código fuente de los programas para estudiarlos y modificarlos; además de tener una licencia GNU/GPL que proteja el derecho de autor y permita que ese código pueda ser distribuido sin el peligro de que sea apropiado por alguien más.

El movimiento de hardware libre -o hardware de especificaciones abiertas-, busca llevar el concepto del software libre al diseño de componentes físicos, especificando una licencia que permite distribuir planos y código fuente de desarrollos de PCBs (Printed Circuit Board por sus siglas en inglés) y hardware electrónico. Asimismo, se considera que un diseño de circuito (esquemático, diseño de PCB y archivos GERBER) debe ser desarrollado con software libre y utilizando formatos abiertos. González y cols. (2003) plantean una clasificación del hardware en función de su diseño y del software empleado para su creación, y definen los tres tipos de archivos primarios necesarios para la fabricación de un PCB: el esquemático, el archivo de PCB y el archivo GERBER. En función de esa clasificación, se puede diferenciar al hardware en tres tipos:

- (P): Software de diseño propietario.
- (L): Software de diseño libre.
- (M): Software de diseño propietario, pero multi plataforma (funciona en sistemas operativos libres y también en propietarios).

Los tres archivos de construcción para el circuito electrónico pueden ser clasificados como

---

<sup>3</sup> La tradición anglosajona se refiere al término *do it yourself* como un movimiento ético, que rechaza la idea de tener que comprar las cosas que uno desea o necesita, intentando hacerlas por uno mismo. Sin embargo, esta práctica tiene raíces muy profundas en Argentina, donde podemos encontrar revistas especializadas en la construcción de electrónica o aeromodelismo como lo fue la revista **LUPIN**, o la revista **Mecánica Popular**, donde se compartían planos y experiencias de fabricación de lo más variopintos.

enteramente libres (tipo LLL) o completamente propietarios (PPP), y todas sus combinaciones posibles.

Pensar en diseños de kits de robótica educativa basados en software libre y hardware de especificaciones libres, permitiría independizarnos de proveedores de tecnología educativa y prácticas que pueden ser consideradas cerradas y restrictivas en la adquisición y utilización de dicho hardware como herramienta didáctica. Existen numerosos proyectos de robótica educativa que dan cuenta de estas lógicas de producción más emparejadas a pensar desde una perspectiva de soberanía tecnológica (Candón-Mena, 2012; Haché, 2014), como el caso del robot DOCTI de la **Cooperativa de Robótica Educativa Córdoba** (CREC) y el robot PICHE, diseñado por los integrantes del **proyecto AMULEN** perteneciente a la Universidad Nacional de Río Cuarto.

Por lo dicho, la robótica educativa con software y hardware libre es una opción más viable para su aplicación en el proceso de aprendizaje, porque permite a las escuelas adaptar la tecnología a las necesidades específicas de la institución, reutilizar componentes que se encuentran en el colegio (aportados por la comunidad escolar), reciclar y ahorrar costos. A su vez, por la gran expectativa que genera en las y los alumnos y docentes, permite romper barreras culturales y políticas que pueden ir en detrimento de la calidad educativa; barreras como el alto costo de adquisición de los elementos, las licencias privativas para el software de control o la falta de documentación específica para comunidades minoritarias (traducciones a idiomas que no son considerados viables comercialmente por ejemplo). La robótica educativa con software y hardware libre permite aprovechar las ventajas que ofrece la filosofía de desarrollo que promueven las comunidades de software libre, donde “mil ojos ven más que uno” (Eric S. Raymond, 1998). Así, se promueve el involucramiento de la comunidad escolar en su conjunto y la construcción colectiva de soluciones que pueden ser utilizadas en otros colegios y viceversa, generando una situación en las que todos los involucrados resultan favorecidos.

### **Hardware de Domótica educativa**

Durante el periodo comprendido entre el año 2019 y 2022, se diseñó, evaluó y probó en espacios no formales de aprendizaje, el hardware de domótica educativa **domoticaro**. Este hardware, basado la placa **wemos d1 mini**<sup>4</sup>, puede ser construido mediante herramientas sencillas como lápiz soldador de estaño (cautín) y alicates, instrumentos comunes que usan las personas que se dedican a la electrónica. En el proceso de desarrollo, se tomaron algunas decisiones orientadas a simplificar su construcción con técnicas caseras y componentes estándares y relativamente fáciles de conseguir o reciclar.

---

<sup>4</sup> [https://www.wemos.cc/en/latest/d1/d1\\_mini.html](https://www.wemos.cc/en/latest/d1/d1_mini.html)

Un circuito impreso (PCB) es una estructura compuesta por caminos, pistas o buses hechos de material conductor, que se encuentran laminados sobre una base no conductora. Su función principal es proporcionar conexiones eléctricas entre los componentes electrónicos y mantenerlos en su lugar de forma mecánica mediante la base. Esta estructura suele ser de diversos materiales como la resinas de fibra de vidrio reforzada, cerámica, plástico, teflón o polímeros como la baquelita. En Argentina se consiguen fácilmente placas de Pertinax Fenólico con una capa de cobre que cubre una de sus caras, en el caso de las placas dobles las dos caras de la placa están cubiertas por una fina capa de cobre. El proceso de producción artesanal de este tipo de placas consiste en dibujar con algún tipo de pintura no soluble en agua (como fibras indelebles) sobre la capa de cobre, para luego poner esa placa dibujada dentro de una cubeta con una solución de cloruro férrico, que procede a hacer un grabado químico sobre el PCB eliminando todo el cobre que no está protegido por la pintura. Mediante este sencillo procedimiento se obtiene una placa que tiene las pistas de cobre marcadas y listas para soldar o hacer los pequeños agujeros necesarios para soldar componentes electrónicos. Al diseñar un PCB, es necesario tener en cuenta las posibilidades técnicas de transferir el diseño realizado con una herramienta CAD<sup>5</sup>, por ejemplo si se usaran técnicas de serigrafía.

La serigrafía es una técnica de impresión empleada en el método de reproducción de documentos e imágenes sobre cualquier material, y consiste en transferir una tinta a través de una malla tensada en un marco, el paso de la tinta se bloquea en las áreas donde no habrá imagen mediante una emulsión o barniz, quedando libre la zona donde pasará la tinta. El sistema de impresión es repetitivo, esto es, que una vez que el primer modelo se ha logrado, la impresión puede ser repetida cientos y hasta miles de veces sin perder definición. La utilización de técnicas serigraficas es bastante estándar en las empresas de desarrollo de circuitos impresos, desde grandes empresas que diseñan placas madres de computadoras, hasta PyMES que diseñan componentes para la industria local, por tanto, es una elección más que evidente para la producción de un hardware pensado para ser fabricado en pequeñas escalas, o de forma casera.

La placa **domoticaro** fue diseñada para ser fabricada de manera casera, a través de PyMES locales o por estudiantes secundarios. Para ello se decidió:

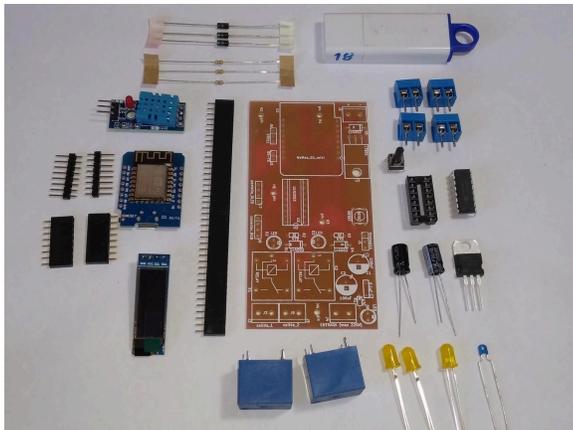
- Usar solo una capa de cobre .
- Las pistas de cobre tienen un grosor sobre dimensionado para soportar mejor la sobre temperatura, en casos de estudiantes que están aprendiendo a soldar con estaño.

---

<sup>5</sup> Como la herramienta de diseño KICAD: <https://www.kicad.org/>

- Las dimensiones de los Pads (espacio de inserción de los componentes) también están sobredimensionadas para que los estudiantes tengan más margen de exceder la cantidad de estaño sin arruinar el PCB.
- La separación entre las pistas se maximizó todo lo posible para evitar cortocircuitos por “bolitas de estaño” que se corrieran por el PCB.
- Todos los componentes electrónicos debían ser del tipo “tecnología de agujeros pasantes” ( THT- Through-Hole Technology) para facilitar su manipulación por parte de los estudiantes así como la capacidad de reparar los errores de colocación o remplazo de componentes.
- Al buscar una solución de compromiso entre la cantidad de componentes que redunden en la calidad y robustez del dispositivo y el tiempo de elaboración que los estudiantes pueden disponer en el ámbito escolar, se optó por mantener lo más bajo posible la cantidad de componentes aunque ello signifique perder seguridad eléctrica contra usos indebidos.
- Empleo de borneras estándares en vez de conectores especiales, que permitirían trabajar con cualquier tipo de cable.
- El software de diseño, los protocolos de comunicación, el firmware embebido en el microcontrolador y las librerías de control deben tener licencias libres y ser compatibles con sistemas GNU/Linux u otro tipo de sistema operativo de licencia libre.

Respecto de la utilización de componentes THT, es importante destacar que permite un mejor trabajo para manos inexpertas, así como facilita el remplazo de componentes defectuosos. Ya que obliga a tener un PCB de mayores dimensiones que lo que se podría lograr usando componentes de montaje superficial (o SMD por las siglas en inglés de surface-mount device).



La elección de los dispositivos **Wemos d1 mini** permitió tener un hardware basado en el

microchip **ESP8266** y unificar los lenguajes de programación necesarios para trabajar en el microcontrolador así como en las computadoras, siendo MicroPython la elección para usar en los ESP8266, y Python para usar en las computadoras de los estudiantes, simplificando el diseño curricular.

### **Una experiencia en bibliotecas populares de la provincia de Córdoba**

Para poner a prueba la propuesta se seleccionaron tres espacios no formales de aprendizaje para trabajar con estudiantes en un curso de domótica educativa. Para esta ponencia, nos focalizaremos en la experiencia desarrollada en uno de los espacios donde trabajamos específicamente la fabricación de hardware.

Pensar una actividad de investigación, implica tomar una postura epistemológica sobre como “vemos el mundo”, la forma en que accedemos e interpretamos los datos que recolectamos y la vigilancia epistemológica que debemos tener para poder considerar que nuestra pesquisa es lo suficientemente plausible para ser analizada y validada por nuestros colegas (Becker, 2019). En ese sentido, trabajar con bibliotecas populares implicó una serie de decisiones, tanto metodológicas como logísticas, en función de la heterogeneidad general que suelen tener las mismas, dado que al ser organizaciones más o menos formales, las herramientas, infraestructura y esquemas organizacionales son construidos y sostenidos por las personas que integran la biblioteca en un momento particular.

Así, podemos ver que para trabajar con tecnologías digitales las tres bibliotecas en las que se participó se encontraban en situaciones muy disímiles en función de sus capacidades.-Solo tenían (además del espacio físico que prestaban para los talleres), luz e internet para compartir con los estudiantes. Se seleccionaron dos bibliotecas para dar cursos de programación con el hardware previamente soldado; en la tercera biblioteca se trabajó directamente con una propuesta curricular orientada a la fabricación de las placas. Esta decisión tuvo que ver con que, al momento de realizar el taller, la biblioteca Juana Manuela Gorriti estaba en un proceso de mudanza desde su lugar original a una nueva instalación y, por eso, su infraestructura y su alcance territorial eran más limitados.

En general, podemos decir que en las tres bibliotecas solo contamos con espacio físico, electricidad e internet a través de wi-fi, dado que los talleres técnicos son muy poco habituales en estos espacios y no cuentan con infraestructura ni herramientas más específicas. Basta hacer una pesquisa en los anuarios que genera la Comisión Nacional de Bibliotecas Populares (CONABIP)<sup>6</sup> y contar los trabajos efectuados por las distintas bibliotecas populares a lo largo

---

<sup>6</sup> [https://www.conabip.gob.ar//experiencias?field\\_tipo\\_de\\_experiencia\\_tid=All&province=C%C3%B3rdoba&page=2](https://www.conabip.gob.ar//experiencias?field_tipo_de_experiencia_tid=All&province=C%C3%B3rdoba&page=2)

del país para ver que hay muy poca oferta de cursos donde se usan las TICs.

La experiencia se organizó bajo modalidad taller de cuatro clases de dos horas de duración donde los participantes, estudiantes de escuelas secundarias de entre 16 y 18 años, pudieron soldar sus propias placas **Domótico**, para luego aprender a programar en Python y Micropython. La biblioteca popular Juana Manuela Gorriti realizó el proceso de convocatoria y selección de dos estudiantes, aclarando que se trataba de una experiencia de investigación en la que se registraría todo el proceso de trabajo de los participantes, para probar las distintas fases de construcción del hardware así como su programación por parte de los estudiantes.

Participaron dos estudiantes secundarios de 17 años. Durante el primer encuentro, se presentó el hardware armado y se enseñó a usarlo con python; primero, trabajaron con python y la librería turtle (parte de la distribución estándar Python), implementando las secuencias didácticas propuestas en su momento por Seymour Papert y su equipo de investigación (Papert, 1987), que sirvieron como introducción al trabajo con primitivas del lenguaje LOGO (Reggini, 1986) en Python. Los estudiantes trabajaron con dos netbooks y con herramientas para fabricación de placas electrónicas, todo proporcionado como parte del equipamiento necesario para el taller. Las actividades de desarrollo se filmaron con cámaras de video. Durante esta primera instancia de capacitación, y luego de la introducción al lenguaje con el uso de la librería turtle, los estudiantes trabajaron sobre hardware previamente ensamblado para usar como interface de programación y controlar algunos dispositivos eléctricos (luces leds, ventiladores) mediante Script escritos en lenguaje Python.

Es interesante ver como durante esta primera jornada de trabajo, el hardware fue presentado a los estudiantes para que tuvieran una primera aproximación a su uso: controlar dispositivos eléctricos mediante WIFI y Python. Si bien pudieron tocarlo y usarlo, había cierta incomodidad en trabajar con electrónica que no estaba “protegida” por una carcasa de plástico, razón por la cual los estudiantes (y una práctica que se repite en gente que no está acostumbrada a trabajar con electrónica) apenas hacían contacto con el hardware. Lo agarraban por los bordes con la mayor de las delicadezas y cuidado posible, aun cuando previamente se les explicó que era un hardware barato diseñado para ser “maltratado”.

Durante la segunda y tercera jornada de trabajo se dedicaron las horas a aprender a soldar y armar las placas **Domótico**. En total, se requirieron cuatro horas de trabajo para soldar dos placas, proceso que demandó desarmar y re ensamblar componentes quemados por sobre temperatura, por mala soldadura o que fueron instalados en posición incorrecta.

Si bien en la primera clase tuvieron acceso al hardware ya ensamblado, trabajar sobre su propio hardware permitió eliminar lo que podríamos definir como “miedo” a romper los

dispositivos y modificar los términos de la relación entre persona/máquina. Para reconocer sus placas, entre clase y clase, los estudiantes les ponían nombre y pasaban a considerarlas como propias. Desmontar la idea de que el hardware es caro y delicado es fundamental para disponer de las placas con fines educativos, que es para lo que fueron diseñadas. Esto fue posible cuando los estudiantes lograron armar sus propias placas, soldando y reparando los errores que habían cometido por falta de experiencia, llegando al punto de tironear de los componentes con pinzas para arrancarlos y remplazarlos por otros.



Figura 1: Mala soldadura, con escoria, sin embargo la placa seguía funcionando

Al usar componentes THT, grandes y resistentes a la manipulación inexperta, se pudo trabajar con el hardware reparando y reemplazando piezas, para luego comenzar a usar y programar las placas con Python. Ya no como una caja negra (Basel, 2012, 2020; Letzen et al., 2019) sino como dispositivos que pueden ser fabricados y reparados por ellos mismos. Alcanzando, así, un nivel de confianza que ayudó al uso del hardware como herramienta didáctica en programación, sin las preocupaciones iniciales de “romperlo” y “arruinarlo”.

### **Pensar el hardware para la enseñanza de programación**

Si bien es cierto que estas experiencias demuestran que es factible incorporar a una propuesta curricular el uso de placas diseñadas para ser fabricadas por estudiantes (Basel, 2020), también encontramos que el nivel de conocimiento específico por parte del equipo docente así como las herramientas de laboratorio necesarias agregan una nueva capa de complejidad logística al ya complicado escenario de implementar sistemas de robótica en escuelas públicas.

Desde esta perspectiva, es importante pensar en las posibilidades reales de trabajar con hardware de estas características en escuelas.

La fabricación de hardware permite situaciones de apropiación por parte de los estudiantes, facilitando la utilización de estas herramientas para la enseñanza de programación y desarrollando el conocimiento necesario para mantener y reparar (además de rediseñar y adaptar) los circuitos electrónicos. Sin embargo, también nos encontramos ante las complejidades logísticas necesarias para la construcción y testado de los componentes, así como la necesidad de un docente formado y preparado en electrónica, con las competencias necesarias para dictar clases en programación.

Algunas posibilidades que fueron exploradas como superadoras de estos inconvenientes consistieron en involucrar otros actores en el desarrollo de los kits de robótica, como universidades y escuelas técnicas, así como PyMEs y cooperativas locales especializadas en desarrollo electrónico, que fabricaron las placas para que los estudiantes de las escuelas secundarias pudieran usarlas, fomentando el involucramiento de diversos actores comunitarios en la práctica educativa.

## **Bibliografía**

- Adell, J., & Bernabé, Y. (2007). Software libre en educación. *Tecnología educativa. Madrid: McGraw-Hill*, 173-195.
- Bağcı, B. B., Kamaşak, M., & Ince, G. (2017). The Effect of the Programming Interfaces of Robots in Teaching Computer Languages. *International Conference on Robotics and Education RiE 2017*, 88-99.
- Barrera Lombana, N. (2015). Use of Educational Robotics as a Teaching Strategy in the Classroom. *Praxis & Saber*, 6(11), 215-234.
- Basel, V. (2012). *Proyecto ICARO: Robótica pedagógica con software y hardware libre*.
- Basel, V. (2020). *Hardware libre en el aula: Una experiencia de capacitación en el uso de recursos educativos abiertos en escuelas técnicas en Tucumán, Argentina*.
- Becker, H. (2019). *Datos, pruebas e ideas: Por qué los científicos sociales deberían tomárselos más en serio y aprender de sus errores*. Siglo XXI Editores.
- Blackley, S., Sheffield, R., Maynard, N., Koul, R., & Walker, R. (2017). Makerspace and reflective practice: Advancing pre-service teachers in STEM education. *Australian Journal of Teacher Education (Online)*, 42(3), 22-37.
- Candón-Mena, J. I. (2012). *Soberanía tecnológica en la era de las redes*. <http://idus.us.es/xmlui/handle/11441/27019>
- Dousay, T. (2017). An evolving makerspace for teacher education. *International Journal of Designs for Learning*, 8(1).

- Eric S. Raymond. (1998). *LA CATEDRAL Y EL BAZAR*.
- Feurzeig, W., Papert, S., Bloom, M., Grant, R., & Solomon, C. (1970). Programming-languages as a conceptual framework for teaching mathematics. *ACM SIGCUE Outlook*, 4(2), 13-17.
- González, I., González, J., & Gómez-Arribas, F. (2003). Hardware libre: Clasificación y desarrollo de hardware reconfigurable en entornos GNU/Linux. *VI Congreso de Hispalinux, Universidad Rey Juan Carlos I*.  
<http://ftp1.nluug.nl/ftp/pub/ftp/os/Linux/doc/LuCaS/Presentaciones/200309hispalinux/8/8.pdf>
- Haché, A. (2014). Soberanía tecnológica. *Dossier sobre Soberanía Tecnológica*, 9-18. Hart, T. D. (2003). *Open source in education*. May.
- Letzen, D., Basel, V., Massolo, A., & Ferrero, F. (2019). Robótica educativa con software libre y hardware de especificaciones abiertas para enseñanza de programación. *II JORNADAS ARGENTINAS DE DIDÁCTICA DE LA PROGRAMACIÓN*, 47.
- Ludeña, E. S. (2019). La educación STEAM y la cultura «maker». *Padres y Maestros/Journal of Parents and Teachers*, 379, 45-51.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- Papert, S. (1987). *Desafío de la mente*. Ediciones Galapagos.
- Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. BasicBooks.
- Pittí, K., Curto Diego, M. B., & Moreno Rodilla, V. (2010). Experiencias constructoras con robótica educativa en el Centro Internacional de Tecnologías Avanzadas. *Teoría de la Educación*, 11(1), 26.
- Reggini, H. C. (1986). *Alas para la mente: LOGO, un lenguaje de computadoras y un estilo de pensar*. Galápagos.
- Rojas, M. D. C., & Zúñiga, A. L. A. (2012). Propuesta comunitaria con robótica educativa: Valoración y resultados de aprendizaje. *Education in the knowledge society (EKS)*, 13(2), 91-119.
- Schrock, A. R. (2014). "Education in Disguise": Culture of a hacker and maker space. *InterActions: UCLA Journal of Education and Information Studies*, 10(1).
- Solomon, C. J., & Papert, S. (1976). A case study of a young child doing Turtle Graphics in LOGO. *Proceedings of the June 7-10, 1976, national computer conference and exposition*, 1049-1056.
- Stallman, R. M. & Lessig, L. (2007). *Software libre para una sociedad libre*. Traficantes de Sueños.
- Vidal Ledo, M. J., Alfonso Sánchez, I., Zacca González, G., & Martínez Hernández, G. (2013). Recursos educativos abiertos. *Educación Médica Superior*, 27(3), 307-320.

## **La operacionalización de los conceptos de Alfabetización, Pensamiento y Participación Computacional en el ámbito escolar. Revisión de Antecedentes**

Ma. Cecilia Martinez<sup>1</sup>, Natalia Monjelat<sup>2</sup>, Echeveste Maria Emilia<sup>1</sup>, Valeria Gonzalez<sup>4</sup>, Martín Torres<sup>4</sup>

cecimart@gmail.com, emilia.echeveste@unc.edu.ar, monjelat@irice-conicet.gov.ar, valeria.gonzalez.angeletti@unc.edu.ar, mtorres341@mi.unc.edu.ar

<sup>1</sup> CONICET - Universidad Nacional de Córdoba

<sup>2</sup> Instituto Rosario de Investigaciones en Ciencias de la Educación  
CONICET-Universidad Nacional de Rosario

<sup>4</sup> Facultad de Ciencias de la Comunicación  
Facultad de Filosofía y Humanidades  
Universidad Nacional de Córdoba

### **Resumen**

La noción de alfabetización digital se comenzó a discutir en los 90s como la capacidad de usar sistemas informáticos focalizando en los sistemas y manejo de información. Esta noción fue evolucionando a lo largo del tiempo. Actualmente se diferencia entre alfabetización digital, alfabetización computacional y pensamiento computacional para incluir habilidades y saberes necesarios para comprender, explicar y crear artefactos computacionales que permitan la transformación de datos aplicando conceptos propios de las Ciencias de la Computación. La disposición de programas abiertos, las comunidades de programadores, las interfaces accesibles a no expertos, los lenguajes de bloques, etc, permiten pensar en la formación de nuestros jóvenes en la comprensión de lenguajes y conceptos para su plena participación en la producción digital. Esta idea da lugar a la noción de participación computacional. En este trabajo realizamos una revisión sistemática de estudios para analizar cómo se han medido estas nociones, es decir, cuáles son los indicadores que nos permiten establecer diferentes niveles o brechas de alfabetización, pensamiento y participación computacional. Esta investigación es necesaria para avanzar en nociones de alfabetización computacional que incluyan habilidades propias de la computación y para informar con instrumentos de medición significativos el estado de situación de un grupo poblacional respecto a estos saberes. Es decir, para medir la brecha computacional.

**Palabras clave:** Brecha computacional, alfabetización computacional, indicadores .

### **Introducción**

Nuestro país tiene profundas desigualdades de acceso, uso y apropiación de saberes de tecnología digital. Mientras que las brechas de acceso y uso de dispositivos han sido documentadas y abordadas con políticas y programas de equipamiento y conectividad [1] la brecha de acceso a saberes y apropiación de saberes - y de las condiciones necesarias para distribuir ese saber- ha sido escasamente monitoreada y por tanto no ha sido visibilizada [2].

Definimos a la brecha de acceso y apropiación de saberes computacionales como la desigualdad de acercamiento a conocimientos que permiten comprender cómo funciona una computadora en su mínima expresión, para entender sus alcances y limitaciones, y para crear y transformar tecnología digital a partir de lenguajes y conceptos del campo de la computación [3]. El acceso a estos saberes se ha definido como “Alfabetización Computacional”. Siguiendo a diferentes autores diferenciamos Alfabetización Digital -que refiere a un modo de aprender ideas científicas, artísticas, técnicas, etc, a través del acceso y creación de representaciones computacionales donde el énfasis está puesto en la representación de la información- de la Alfabetización Computacional -que además incluye saberes necesarios para procesar información [4,5,6].

Dos referentes precisan la idea de brecha digital para el ámbito escolar. DiSessa [4] va a referirse a brecha de alfabetización digital y Jeannette Wing a pensamiento computacional [6]. En ambos casos, estas definiciones permiten dar cuenta de la brecha de saberes conceptuales que permiten comprender, participar, crear y transformar en un mundo digital. Jacob [7] integra estos dos conceptos para hablar de Pensamiento y Alfabetización Digital argumentando que la programación utiliza a las computadoras como “medio meta” o base material para expresar ideas y en ese sentido repone la idea de alfabetización en tanto proceso de apropiación de un lenguaje y sus reglas para expresar una idea a partir de sus símbolos y signos. Las principales demandas cognitivas que requiere la alfabetización computacional son para estos autores [4,6,7] la *abstracción* (que requiere identificar las variables o elementos esenciales de un problema para convertirlos en objetos que la computadora pueda manipular) *automatización* (que implica el desarrollo de instrucciones de manera que una máquina los pueda ejecutar) y el *análisis* (para determinar si las abstracciones y automatizaciones son correctas y pertinentes para las situaciones que aborda). En ese sentido el *análisis* requiere de la interacción del pensamiento humano con el poder de cómputo.

En este último tiempo se han introducido aportes de la pedagogía crítica para precisar la selección, jerarquización, organización y sentido de la enseñanza de la computación en el contexto de la escuela. En la pedagogía crítica subyace el sentido emancipatorio de la educación [28]. Desde este lugar, es necesario abordar las estructuras y sistemas que producen desigualdades y concentraciones de saberes que otorgan jerarquías sociales [28].

Recuperando la teoría crítica social de Habermas, Clear [29] identifica tres tipos de intereses cognitivos: 1) el interés técnico –asociado con el uso de herramientas-; 2) el interés práctico –asociado con la creación de saberes para controlar y definir procesos-; 3) y el interés emancipatorio –asociado con la capacidad de reflexión y comprensión sobre diferentes fenómenos para desarrollar autonomía, responsabilidad y libertad-. En esta misma línea, otros autores analizan la ausencia de diversos contenidos de computación en la escuela, que no permiten a la ciudadanía una mínima comprensión de cómo funcionan los sistemas computacionales a pesar de que estos influyen en la construcción de nuestras subjetividades. Diversos informes muestran que los contenidos de computación representan a un subconjunto de enfoques y saberes hegemónicos del campo, dejando afuera saberes del campo del software y hardware libre, redes comunitarias de internet, nociones que apelen a los intereses de las minorías y las mujeres y que permitan una mirada crítica de los sistemas de información [8]. En este sentido, el enfoque de la participación computacional plantea diferencias con el pensamiento computacional. La participación computacional implica, además de comprender conceptos; crear aplicaciones, formar parte de comunidades de software, remixar, recombinar, re-mezclar código y usar múltiples interfaces e instrumentos para abordar un problema computacional. Esto es, integrar una solución computacional a un conjunto más amplio con el que se atiende una problemática determinada en una comunidad. Una app para permitir la comunicación de personas cuya lengua no estaba incluida en un portal de trámites, un dispositivo programable en una torre para detectar incendios forestales, etc. Autores como Ko [9] advierten sobre los límites de los sistemas de cómputo para eludir problemas culturales como la discriminación por diferentes causas y dan cuenta de cómo los sesgos producidos por algoritmos a partir del material con el que son entrenados colaboran en profundizar las injusticias.

Siguiendo estas conceptualizaciones teóricas nos proponemos en este breve trabajo identificar cómo se ha medido la brecha de alfabetización, pensamiento y participación computacional de manera de avanzar en investigaciones que permitan visibilizar cómo circulan estos saberes en nuestras escuelas.

### **Metodología**

Se realizó una revisión sistemática de bibliografía existente en varias etapas en la base de datos Scopus, siguiendo la secuencia sugerida por [10] y utilizando como palabras claves los términos -tanto en español como en inglés-, de “alfabetización o pensamiento o participación computacional” y “escuela”. Se seleccionaron solo artículos científicos. Esta búsqueda arrojó 58 artículos de los cuales se incluyeron 17, ya que el resto no abordaban nuestro objeto de estudio en el ámbito escolar. El equipo analizó cada artículo identificando la manera en que definían los

conceptos de alfabetización, pensamiento o participación computacional y los indicadores e instrumentos que utilizaron para su medición o estudio.

Recuperamos el concepto de revisión sistemática de Okoli y Schabram [30] en tanto método explícito, sistemático y reproducible para identificar y sintetizar un cuerpo de bibliografía académica sobre un tema. Estos autores también proponen no simplemente compilar la información, sino también interpretarla de manera analítica, reconociendo cómo algunas de sus características han cambiando en el tiempo ofreciendo una síntesis crítica.

En este breve informe presentamos resultados preliminares de dicho análisis organizados en dos dimensiones: 1) definiciones del concepto de alfabetización computacional y 2) indicadores de los mismos.

En general se observa que la definición de alfabetización computacional en los trabajos analizados remiten a una comprensión de conceptos fundamentales de las Ciencias de la Computación y habilidades de prácticas computacionales que permiten desarrollar un programa informático que transforma datos [11,12,13,14,15,16,17,18,19,20,21]. Asimismo, la mayoría de los estudios reconocen -como condición necesaria para apropiarse de estos conceptos y prácticas- el aprendizaje de la programación y la alfabetización computacional. Se citan los mismos referentes en casi todos los artículos que hablan de este término siendo DiSessa, Wing Papert, Resnick y Brennan los más mencionados.

La diferencia entre los estudios, radica en la consideración de una dimensión crítica en los procesos de alfabetización computacional y en el desarrollo de nuevas alfabetizaciones computacionales a partir de áreas específicas. Respecto de la dimensión crítica, algunos estudios incluyen los conceptos de Participación Computacional, la perspectiva ética para analizar sesgos, beneficios, riesgos, la mirada política y social para entender el campo social y las regulaciones en donde se enmarcan las producciones digitales y la posibilidad de crear e intervenir artefactos computacionales para el bien común, con un resguardo ético [22,23,24,25,26]. En algunos casos, se menciona la perspectiva crítica en la propuesta del estudio pero finalmente no se consideran indicadores específicos y se mide solamente el aspecto conceptual y prácticas computacionales. El foco está puesto en la formación ciudadana que implica reconocer qué y cómo operan las computadoras, la lógica que existe en el armado de los programas, identificando las fortalezas y desventajas de las herramientas de programación.

Se incluyen los enfoques pedagógicos del Aprendizaje Basado en Problemas, centrado en un problema de impacto social promoviendo la participación social a través del vector digital. Se busca empoderar a los adolescentes para diseñar, crear y programar tecnologías en red

interactivas y tangibles, mostrando cómo las herramientas de programación distribuida pueden ofrecer a los jóvenes agencia para trabajar dentro de una variedad de estructuras de colaboración.

En relación a la identificación de áreas específicas de la alfabetización computacional o diferentes alfabetizaciones computacionales se mencionan en esta muestra las siguientes: erudición digital (como la capacidad de comprender cómo se construye un texto digital publicable), alfabetización en software (que incluye el dominio de ciertos lenguajes y plataformas programables), alfabetización en Inteligencia Artificial, alfabetización en Aprendizaje Automático, Alfabetización en Ciencia de datos (estas incluyen dominio de conceptos, prácticas y análisis crítico de estas áreas particulares), y Alfabetización crítica (que incluye la capacidad de diseñar, interpretar, y depurar).

A continuación, en la Tabla 1 presenta una síntesis de las dimensiones de Alfabetización Computacional encontradas en esta revisión, los principales indicadores y los trabajos de referencia.

Dimensión de Alfabetización Computacional	Definición	Artículos	Indicadores
Identificación y comprensión del funcionamiento de sistemas informáticos.	Los estudios que enfatizan esta dimensión incluyen saberes que permiten la identificación, explicación y producción de artefactos computacionales. Visualización de datos, reconocimiento de flujos, creación, depuración y diseño de programas. Reconoce el aprendizaje de la programación como fundante a estas capacidades. Incluyen la perspectiva de la seguridad informática y el trabajo en redes.	[11 12, 13,14,15,16, 17,18, 19,20,21]	En general los estudios que dieron cuenta de la identificación y comprensión de sistemas digitales usaron los siguientes indicadores: uso de conceptos y habilidades computacionales en una actividad de programación. que incluyen abstracción, diseño algorítmico, análisis, recolección y representación de datos, descomposición, paralelización, generación y reconocimiento de patrones, simulación, transformación de datos, lógica condicional, conexión a otros campos, modelización. Identificación de elementos de código, intervención de un lenguaje computación, comprensión de cómo se construye un texto digital en un código publicado. Prácticas de depuración y retoques de un código. Habilidades de creación y diseño de un artefacto computacional.

Creación, transformación, participación y análisis crítico y ético de artefactos computacionales.	Los estudios clasificados en esta dimensión incluyen la categoría de Pensamiento Computacional y además agrega la perspectiva de la Participación Computacional, la ética, las políticas de producción del conocimiento, el bien común, y los grupos detrás de los artefactos y los datos que se usan para tecnologías de Aprendizaje Automático.	[22,23, 24,25,26]	Los estudios que analizan la alfabetización computacional desde la perspectiva crítica incluyen entre sus indicadores la identificación de tecnologías de IA, y Aprendizaje Automático, programación de estas tecnologías, y análisis ético. Algunos casos analizan la creación de automatizaciones orientadas al bien común.
Actitudes personales en actividades computacionales	Los estudios incluidos aquí trabajan todos desde la perspectiva del Pensamiento Computacional que incluye conceptos y prácticas pero además consideran las actitudes hacia la disciplina.	[27, 13, 19]	Desde esta perspectiva se analizan los intereses de los estudiantes, autopercepción de competencia y confianza en sí mismo para resolver el problema, interés futuro en la disciplina a partir de la experiencia de aprendizaje.

## Conclusiones

Este breve informe de un trabajo en curso, relevó estudios que definieron y recolectaron datos en torno a la alfabetización computacional en sus diferentes dimensiones y perspectivas. Todos los artículos reconocen conceptos y prácticas computacionales como parte de esta alfabetización, y el aprendizaje de la programación como experiencia fundante de apropiación de estas prácticas y conceptos. En algunos artículos más recientes, comienza a incluirse la perspectiva de la pedagogía crítica que analiza los aspectos éticos, sociales y políticos de la tecnología digital.

Por otro lado, resulta notable la inexistencia de trabajos de investigación en nuestra región, lo que configura una importante vacancia, al tiempo que una necesidad que debe atenderse en pos de la elaboración de políticas públicas para promover la alfabetización computacional. En ese sentido este trabajo será insumo para construir indicadores que permitan analizar los procesos de alfabetización computacional en las escuelas en el marco de un proyecto más grande.

Asimismo, los datos relevados permitieron reconstruir que la Programación -y más

recientemente las Ciencias de datos e Inteligencia Artificial-, son las áreas que más peso tienen en estas propuestas de enseñanza. En ese sentido, la oferta educativa en el campo computacional parecería reproducir las prácticas computacionales hegemónicas. Otras áreas tales como la seguridad informática, redes, arquitecturas de computadoras, sistemas operativos libres y en general la perspectiva del software libre; tienen escasa o casi nula presencia en las propuestas. Propuestas ligadas al trabajo con hardware y software libre, pueden contribuir a delimitar alcances más heterogéneos de la Ciencias de la Computación, incluso promover una mirada crítica de los sistemas de información.

Finalizando, en la era de la Inteligencia Artificial -y sus mecanismos implicados-, se advierte necesario que el desarrollo de políticas públicas relacionadas a la producción del conocimiento en las Ciencias de Computación; contemplen una perspectiva crítica y participativa que promueva una adecuada apropiación de los conceptos y procesos, para luego pasar la creación y transformación activa de los mismos. De esta manera, poder avanzar en los diseños curriculares oficiales que permitan enmarcar las propuestas de enseñanza en esta área, requiere primero lograr ciertos consensos sobre qué se entiende por alfabetización computacional, y qué saberes y áreas la componen. Estas coordenadas también contribuirán al direccionamiento de la toma de decisiones y acciones en pos de elevar la calidad educativa, la construcción de un adecuado marco normativo, y al campo de la formación docente inicial y continua. Este trabajo ofrece reflexiones en estos sentidos.

## Referencias

1. Terigi, F. (2020). Escolarización y pandemia: alteraciones, continuidades, desigualdades. *REVCOM*
2. Scasso, Marino, Colombini y Bortolotto (2019) Evaluación de la Iniciativa Program.ar. Fundación Sadosky.
3. Vakil, S. (2018). Ethics, identity, and political vision: Toward a justice-centered approach to equity in computer science education. *Harvard Educational Review*, 88(1), 26-52.
4. DiSessa, A. A. (2000). *Changing minds: Computers, learning, and literacy*. Mit Press.
5. Jacob, S. R., & Warschauer, M. (2018). Computational thinking and literacy. *Journal of Computer Science Integration*, 1(1).
6. Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725. Author, F.: Article title. *Journal* 2(5), 99–110 (2016).

7. Jacob, S. R., & Warschauer, M. (2018). Computational thinking and literacy. *Journal of Computer Science Integration*, 1(1).
8. Burke, Q., O'Byrne, W. I., & Kafai, Y. B. (2016). Computational participation: Understanding coding as an extension of literacy instruction. *Journal of adolescent & adult literacy*, 59(4), 371-375.
9. Ko, A. J., Oleson, A., Ryan, N., Register, Y., Xie, B., Tari, M., ... & Loksa, D. (2020). It is time for more critical CS education. *Communications of the ACM*, 63(11), 31-33.
10. Gough, D. (2007). Weight of Evidence: A framework for the appraisal of the quality and relevance of evidence. *Research Papers in Education*, 22(2), 213-228. <https://doi.org/10.1080/02671520701296189>
11. Basogain, X., Olabe, M. Á., Olabe, J. C., & Rico, M. J. (2018). Computational thinking in pre-university blended learning classrooms. *Computers in Human Behavior*, 80, 412-419.
12. Dichev, C., & Dicheva, D. (2017). Towards data science literacy. *Procedia Computer Science*, 108, 2151-2160.
13. Terroba, M., Ribera, J. M., Lapresa, D., & Anguera, M. T. (2021). Propuesta de intervención mediante un robot de suelo con mandos de direccionalidad programada: análisis observacional del desarrollo del pensamiento computacional en educación infantil. *Revista de psicodidáctica*, 26(2), 143-151.
14. De Paula, B. H., Burn, A., Noss, R., & Valente, J. A. (2018). Playing Beowulf: Bridging computational thinking, arts and literature through game-making. *International journal of child-computer interaction*, 16, 39-46.
15. Lockridge, T., Paz, E., & Johnson, C. (2017). The Kairos Preservation Project. *Computers and Composition*, 46, 72-86.
16. Losh, E. (2014). From authorship to authoring: Critical literacy, expert users, and proprietary software. *Computers and Composition*, 33, 40-49.
17. Mannila, L., Leinonen, T., Bauters, M., & Veermans, M. (2023). Student and teacher co-agency when combining CT with arts and design in a cross-curricular project. *Computers and Education Open*, 4, 100132.
18. Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach

- computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126, 296-310.
19. Weintrop, D., & Wilensky, U. (2019). Transitioning from introductory block-based and text-based environments to professional programming languages in high school computer science classrooms. *Computers & Education*, 142, 103646.
  20. Kelly, A., Finch, L., Bolles, M., & Shapiro, R. B. (2018). BlockyTalky: New programmable tools to enable students' learning networks. *International Journal of Child-Computer Interaction*, 18, 8-18.
  21. Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, 97, 129-141.
  22. Vourletsis, I., & Politis, P. (2022). Exploring the effect of remixing stories and games on the development of students' computational thinking. *Computers and Education Open*, 3, 100069.
  23. Su, J., & Zhong, Y. (2022). Artificial Intelligence (AI) in early childhood education: Curriculum design and future directions. *Computers and Education: Artificial Intelligence*, 3, 100072.
  24. Ottenbreit-Leftwich, Anne T.; Kwon, Kyungbin; Brush, Thomas A.; Karlin, Michael; Jeon, Minji; Jantaraweragul, Katie; Guo, Meize; Nadir, Hamid; Gok, Fatih; Bhattacharya, Parama Ottenbreit-Leftwich, A. T., Kwon, K., Brush, T. A., Karlin, M., Jeon, M., Jantaraweragul, K., ... & Bhattacharya, P. (2021). The impact of an issue-centered problem-based learning curriculum on 6th grade girls' understanding of and interest in computer science. *Computers and Education Open*, 2, 100057.
  25. Irgens, G. A., Vega, H., Adisa, I., & Bailey, C. (2022). Characterizing children's conceptual knowledge and computational practices in a critical machine learning educational program. *International Journal of Child-Computer Interaction*, 34, 100541.
  26. Laupichler, M. C., Aster, A., & Raupach, T. (2023). Delphi study for the development and preliminary validation of an item set for the assessment of non-experts' AI literacy. *Computers and Education: Artificial Intelligence*, 4, 100126.
  27. Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2019). Exploring children's learning experience in constructionism-based coding activities through design-based research. *Computers in Human Behavior*, 99, 415-427.

28. Clear, T. (2004). Critical enquiry in computer science education. *Computer science education research*, 101–125.
29. Romero Moñivas, J. R. (2013). Del aprendizaje difuso al aprendizaje situacional. Una explicación estructural-conflictiva de las relaciones entre la tecnología y la educación en la sociedad de la información. *Revista de Sociología de la Educación-RASE*, 6(2):210–227.
30. Okoli, C., & Schabram, K. (2015). A guide to conducting a systematic literature review of information systems research.

## **Integración de las prácticas científicas y la programación con JavaScript para el aprendizaje de la genética en la escuela secundaria**

Gimena B. Fussero<sup>1</sup>, Maricel Occelli<sup>2</sup>, Nicolás Wolovick<sup>3</sup>, Héctor García<sup>4</sup>  
gimenafussero@unc.edu.ar, maricel.ocelli@unc.edu.ar, nwolovick@gmail.com,  
hrgarcia@escuelasproa.edu.ar

<sup>1</sup> Facultad de Ciencias Exactas, Físicas y Naturales - SECyT  
Universidad Nacional de Córdoba

<sup>2</sup> Facultad de Ciencias Exactas, Físicas y Naturales - CONICET  
Universidad Nacional de Córdoba

<sup>3</sup> Facultad de Matemática, Astronomía, Física y Computación  
Centro de Computación de Alto Desempeño  
Universidad Nacional de Córdoba

<sup>4</sup> Escuela ProA Sede Centro Pucará

### **Resumen**

La inclusión de las prácticas científicas (PC) en los ámbitos educativos brinda oportunidades para que las y los estudiantes desarrollen razonamientos y habilidades propias de las ciencias acercándose a cómo se genera y cómo se valida el conocimiento científico. Como PC se entienden a aquellas utilizadas por la comunidad científica para generar y evaluar conocimientos donde confluyen aspectos epistémicos, cognitivos y socioculturales. En este contexto, se considera que el desafío de la alfabetización científica consiste en generar instancias formativas en las cuales se desarrollen habilidades para analizar y tomar decisiones sobre asuntos sociocientíficos. En el caso de la Biología, este aspecto es especialmente importante ya que muchos de los interrogantes a los cuales se enfrentará el estudiantado durante su vida se encontrarán relacionados a cuestiones biológicas con impacto social y tecnológico. Desde la perspectiva de las Ciencias de la Computación se desarrolló el concepto de alfabetización computacional. Este concepto implica una serie de enfoques que se encuentran basados en las ideas de Papert, Kemeny y Kurtz sobre las potencialidades de la programación no sólo de la informática sino en otros campos. En línea con lo anterior, en la construcción del conocimiento científico, la relación entre las computadoras y las simulaciones reorientó las prácticas de la comunidad científica introduciendo nuevas formas de construir conocimientos. Estas transformaciones epistémicas no sólo se corresponden a las condiciones mismas del saber, sino, que reorganizan los espacios tecnológicos y los diferentes modos de saber (Rodríguez, 2019). Este aspecto podría incluirse en las clases de ciencias ya que ambos campos, la Biología y las Ciencias de la Computación, buscan contribuir a la formación científico-tecnológica de las personas. A partir de los referentes teóricos expuestos surge el interés por integrar las PC y la programación en la escuela secundaria lo cual podría permitir

que las y los estudiantes construyan saberes tendientes a contribuir a su alfabetización científica y computacional. En función de ello se propuso, en el marco de una tesis doctoral, una investigación basada en diseño con el fin de estudiar las oportunidades de aprendizaje que tienen lugar cuando se ponen en sinergia las PC y la programación en la escuela secundaria. En este trabajo se presentan resultados preliminares que se derivaron del primer ciclo que tuvo lugar durante agosto del año 2022 el cual implicó a los espacios curriculares de Biología y de Almacenamiento y estructura de datos II en un 4° año de una escuela ProA de la ciudad de Córdoba. Los principales resultados indican que este tipo de secuencias didácticas, donde se integran diferentes disciplinas, promueven la formación científico-tecnológica del estudiantado.

**Palabras clave:** Biología, Alfabetización científica, Alfabetización computacional, Investigaciones basadas en diseño (*DBR - Design Based Research*), Escuelas ProA.

### **Introducción**

Las prácticas científicas (PC) en los ámbitos educativos permiten generar contextos para que el estudiantado se aproxime al desarrollo de habilidades que son propias de las ciencias acercándose a cómo se construye, evalúa y valida el conocimiento científico (Crujeiras Pérez y Jiménez Aleixandre, 2015; Duschl y Jiménez Aleixandre, 2012; Jiménez Aleixandre y Crujeiras Pérez, 2017; Osborne, 2014). Por PC se entiende a aquellas que son puestas en juego por las y los científica/s al generar y evaluar el conocimiento científico donde confluyen aspectos epistémicos, cognitivos y socioculturales (Erduran y Dagher, 2014). En este escenario se considera que la actividad científica comprende tres procesos: investigación, evaluación y elaboración de explicaciones (Crujeiras Pérez y Cambeiro, 2018) los cuales se asocian a tres PC: indagación, argumentación y modelización (Jiménez Aleixandre y Crujeiras Pérez, 2017). Cuando dichas PC se trasladan a contextos educativos, las destrezas que se asocian a cada una se solapan como ocurre en los ámbitos científicos.

Desde la perspectiva del aprendizaje situado, donde el aprendizaje tiene lugar dentro del contexto de comunidades de práctica (Wenger, 2001), a través de las PC se pretende que el estudiantado no sólo desarrolle determinadas destrezas procedimentales sino también de razonamiento generando una participación reflexiva en la construcción y la evaluación del conocimiento (Berland, Schwarz, Krist, Kenyon, Lo y Reiser, 2016). Al respecto, un número considerable de investigaciones (Duschl y Jiménez Aleixandre, 2012; Jiménez Aleixandre y Crujeiras Pérez, 2017; Osborne, 2014) indican la importancia de involucrar al estudiantado en actividades cognitivas, discursivas y sociales que le permitan desarrollar razonamientos, habilidades y argumentaciones propias de la ciencia. En este sentido, existe un amplio acuerdo sobre la necesidad de que en el diseño de secuencias didácticas se considere cómo se genera y cómo se valida el conocimiento científico (Crujeiras Pérez y Jiménez Aleixandre, 2012;

Crujeiras Pérez y Jiménez Aleixandre, 2015; Duschl y Jiménez Aleixandre, 2012; Jiménez Aleixandre y Crujeiras Pérez, 2017; Osborne, 2014).

Si bien las actividades que generan la construcción de la ciencia escolar son diferentes a las que realiza la comunidad científica para construir conocimiento, enseñar PC ofrece nuevas realidades en los espacios educativos y promueve comunidades que producen conocimiento científico escolar (Lehrer y Schauble, 2006). En consecuencia, se advierte la importancia de promover situaciones escolares que incluyan a las prácticas asociadas a la indagación, a la argumentación y a la modelización del mismo modo que se encuentran presentes en las actividades de la comunidad científica lo que promovería la alfabetización científica (Osborne, 2014).

El desafío de la alfabetización científica es generar instancias formativas en las cuales se desarrollen habilidades para analizar y tomar decisiones sobre asuntos que involucran conceptos científicos (Justi, 2006). En el caso de la Biología, este aspecto es especialmente importante ya que muchos de los interrogantes a los cuales se enfrentarán las y los estudiantes durante su vida se encontrarán relacionados a cuestiones biológicas con impacto social y tecnológico (Reiss, 2018) donde la ciudadanía debe estar formada científicamente para la toma de decisiones para la acción en esos contextos. Considerando lo anterior, el objetivo de la educación científica no busca crear nuevos conocimientos, sino más bien ayudar al estudiantado a comprender un cuerpo de conocimiento existente, consensuado y acordado (Osborne, 2014). Esta perspectiva significa ir más allá del aprendizaje de teorías y conceptos hacia una participación de las prácticas a través de las cuales se construye el conocimiento biológico (Jiménez Aleixandre y Evagorou, 2018).

Desde las Ciencias de la Computación se desarrolló el concepto de alfabetización computacional, concepto que implica una serie de enfoques que, teniendo diferentes denominaciones -Informática para todos, Programación informática para todos, Pensamiento computacional- se encuentran basados en las ideas de Papert, Kemeny y Kurtz sobre las potencialidades de la programación en el aprendizaje, no sólo de la Informática sino en otros campos y disciplinas (Vee, 2013).

Según las y los informáticas/os, la alfabetización computacional se trata de una habilidad que debería enseñarse y sostenerse más ampliamente (Lee, 2014). Partiendo de dicha premisa se sostiene que la escritura y la programación han seguido trayectorias históricas y sociales que permitieron formas de comunicación y de conexiones similares a las tecnologías materiales. En este sentido, las investigaciones sobre la alfabetización computacional pueden ayudar a comprender mejor la dinámica social, técnica y cultural que se establecen mediante la programación siendo esta similar a la alfabetización textual la cual les permite a las personas

representar e interpretar ideas. En este contexto, la programación se está alejando del dominio exclusivo de la Informática y se está volviendo más relevante en otros campos como la Biología e incluso el estudio de la literatura y la historia (Vee, 2013).

La programación hace varias décadas se ha incorporado en los ámbitos escolares y en los últimos años su adhesión ha aumentado. Se percibe como una herramienta significativa de incluir en los diferentes niveles educativos ya que no se trata solo de una competencia cognitiva para diseñar códigos sino también de una competencia social y cultural (Kalelioğlu y Gülbahar, 2014). En síntesis, la programación desde la perspectiva de la alfabetización computacional abre un camino hacia una pedagogía de escritura más completa e inclusiva lo que significa que la programación no puede relegarse al dominio exclusivo de la Informática (Vee, 2013).

De hecho, en la construcción del conocimiento científico, las simulaciones computacionales reorientaron las prácticas de la comunidad científica introduciendo nuevas formas de producción científica (Lenhard, Küppers y Shinn, 2007). Este aspecto podría considerarse al momento de planificar secuencias didácticas para las clases de ciencias ya que ambos campos, la Biología y las Ciencias de la Computación, buscan contribuir a la formación de las personas garantizando el desarrollo de saberes, digitales y científicos, para que puedan desenvolverse plena y sustentablemente en el mundo. Además, en Argentina la Ley N° 26.206 estableció como objetivo “el desarrollo de competencias para el manejo de los nuevos lenguajes TIC” y el Ministerio de Educación (2017) propuso *Aprender Conectados* que busca garantizar la alfabetización digital para el aprendizaje de competencias y saberes necesarios para la integración en la cultura digital y en la sociedad del futuro haciendo hincapié en la programación y la robótica

En síntesis, se puede considerar que la inclusión de las PC en la educación en ciencias naturales les permite a las y los estudiantes construir conocimientos que contribuyan a su alfabetización científica. A su vez, la programación con JavaScript se presenta como una herramienta potencial para que el estudiantado desarrolle conocimientos desde la alfabetización computacional. Entonces, surge el interés por integrar las PC y la programación en la escuela secundaria lo cual podría permitir que las y los estudiantes construyan saberes tendientes a contribuir a su alfabetización científica y computacional. En función de ello se propuso para el trabajo de tesis un estudio de diseño que permita estudiar en profundidad las oportunidades de aprendizaje que tienen lugar cuando se ponen en sinergia las PC y la programación en la escuela secundaria.

### **Metodología**

Esta investigación se enmarca dentro del Doctorado en Educación en Ciencias Básicas y

Tecnología, carrera cogestionada por la Facultad de Matemática, Astronomía, Física y Computación, la Facultad de Ciencias Exactas, Físicas y Naturales y la Facultad de Ciencias Químicas de la Universidad Nacional de Córdoba. El trabajo de tesis se está realizando utilizando la metodología conocida como Investigaciones basadas en diseño (DBR) las cuales se orientan a generar conocimientos que permitan analizar y transformar a otros contextos educativos asentando el estrecho vínculo entre la enseñanza y el aprendizaje de un contenido en particular (Guisasola y Oliva 2020). Este tipo de investigaciones se caracterizan por responder a dos metas diferentes. Por un lado, una meta pedagógica relacionada a los aprendizajes que se buscan favorecer y promover en las y los estudiantes que participan en las secuencias didácticas y por el otro una meta teórica vinculada a los conocimientos que se pretenden construir a partir de diferentes análisis de los datos recabados (Ponte, Carvalho, Mata-Pereira y Quaresma 2016). En este caso en particular, la meta pedagógica se relaciona a fomentar aprendizajes sobre la epigenética integrando las prácticas científicas de la modelización, la argumentación y la indagación con la programación en JavaScript. En relación a la meta teórica, se busca aportar al campo de la didáctica de las ciencias acerca de la construcción de conceptos relacionados a la epigenética en diferentes contextos con el fin de evitar explicaciones reduccionistas que sólo consideren al determinismo biológico.

Las DBR se basan en ciclos iterativos de toma de datos, en el presente trabajo se presentan datos preliminares respecto al primer ciclo. La investigación se llevó adelante en la escuela ProA Sede Capital Centro (Córdoba, Argentina), una institución pública con orientación en Desarrollo de Software. Se trabajó en un 4° año (entre 15 y 16 años de edad) en los espacios curriculares de Biología y de Almacenamiento y estructura de datos II. La tesista co-diseñó la secuencia didáctica junto a la docente de Biología y al docente de Almacenamiento y estructura de datos II la cual se implementó durante 5 semanas entre los meses de agosto y septiembre de 2022 (fase de preparación). Durante el desarrollo de las clases se iban recolectando las diferentes actividades que realizaban las y los estudiantes en “papel” cuando se encontraban en Biología. Mientras que cuando construían modelos en la sala de informática en la materia Almacenamiento y estructura de datos II, luego de cada clase cada grupo enviaba a la tesista el link de un repositorio (GitHub) lo que permitió tener un seguimiento en la construcción, revisión y posterior evaluación de los modelos (fase de implementación). También se realizaron test (pretest y postest) antes de comenzar con el desarrollo de la secuencia didáctica y al finalizar la misma. Asimismo, se llevaron a cabo entrevistas a estudiantes y a la y el docente implicados en las actividades. Todos los datos recolectados con estos diferentes instrumentos permitieron realizar un análisis (fase de análisis) y el re-diseño de la secuencia que se implementó en el segundo ciclo iterativo (agosto-septiembre de 2023).

### **Resultados preliminares**

En la secuencia didáctica diseñada participaron 20 estudiantes. Las actividades que se plantearon para realizar en Biología fueron resueltas de manera individual mientras que las planificadas para llevar a cabo en Almacenamiento y estructura de datos II fueron realizadas en grupos de entre dos y tres integrantes (Fig. 1).



Figura 1: Estudiantes realizando las actividades planificadas en el laboratorio de informática ubicado en el primer piso de la institución. Cada grupo de dos o tres integrantes trabajaba con dos o tres computadoras según la disponibilidad.

Como se mencionó anteriormente, en este trabajo se presentan algunos resultados preliminares referentes al primer ciclo de toma de datos.

Al finalizar las actividades, las y los estudiantes realizaron una coevaluación de los códigos programados por otro grupo. Para ello se realizó el intercambio de los link a los diferentes repositorios de GitHub (Fig. 2) entre los diferentes grupos y las y los estudiantes analizaron los códigos considerando: el coding style, la documentación del código, el uso de loops, los nombres de las variables, las entradas y salidas y finalmente las estrategias de resolución. En relación al coding style, el 62,5% consideró que el estilo del código era bueno (*“El estilo del código está bien alineado y codificado”, “Coding style bien estructurado en cuanto a funciones, arreglos, variables y representación de la idea”*) mientras que el 37,5% restante indicó que ese aspecto debía mejorarse *“El coding style no está mal, pero hay varios errores, por ejemplo, hay espacios de más o falta de espacios”*). En el caso de la documentación del código, las proporciones se invirtieron siendo el 37,5% quienes consideraron que el código estaba bien documentado (*“Si está bien documentado en P5.js”*) y el 62,5% que al mismo le faltaba información (*“Falta documentación”, “No hay ningún comentario”*). Respecto a la utilización de loop, la mitad consideró que el código que analizó contenía loop (*“Tiene draw”, “Tiene una maps”*) mientras que el otro 50% indicó que debían utilizarse más loops (*“Tiene un solo loop para el nanómetro”*). Considerando a la denominación de las variables, el 87,5% manifestó que los nombres de las variables utilizadas eran precisos (*“Los nombres de las variables son*

precisos”) mientras que el 12,5% consideró lo contrario. Acerca de las entradas y las salidas, en el 50% de los casos se hacía referencia al mouse/teclado mientras que el 50% restante indicaba cuestiones referidas al código (“*Si tiene entrada el código y retorna una buena devolución*”). Finalmente, en cuanto a las estrategias de resolución en algunos casos (62,5%) indicaron qué funcionalidades se utilizaron (“*Se utilizan sliders y checkboxes combinados con maps y funciones matemáticas para brindar una mejor experiencia al usuario. Aun así, se usaron tal vez demasiados "if"*”) mientras que el 37,5% restante hizo valoraciones (“*Está bien planificado y organizado*”). Estos resultados indicarían que, por un lado, las y los estudiantes pudieron realizar una evaluación crítica de los códigos realizados por otro grupo de estudiantes y por el otro que los códigos realizados contarían con el esquema básico esperable en una programación.

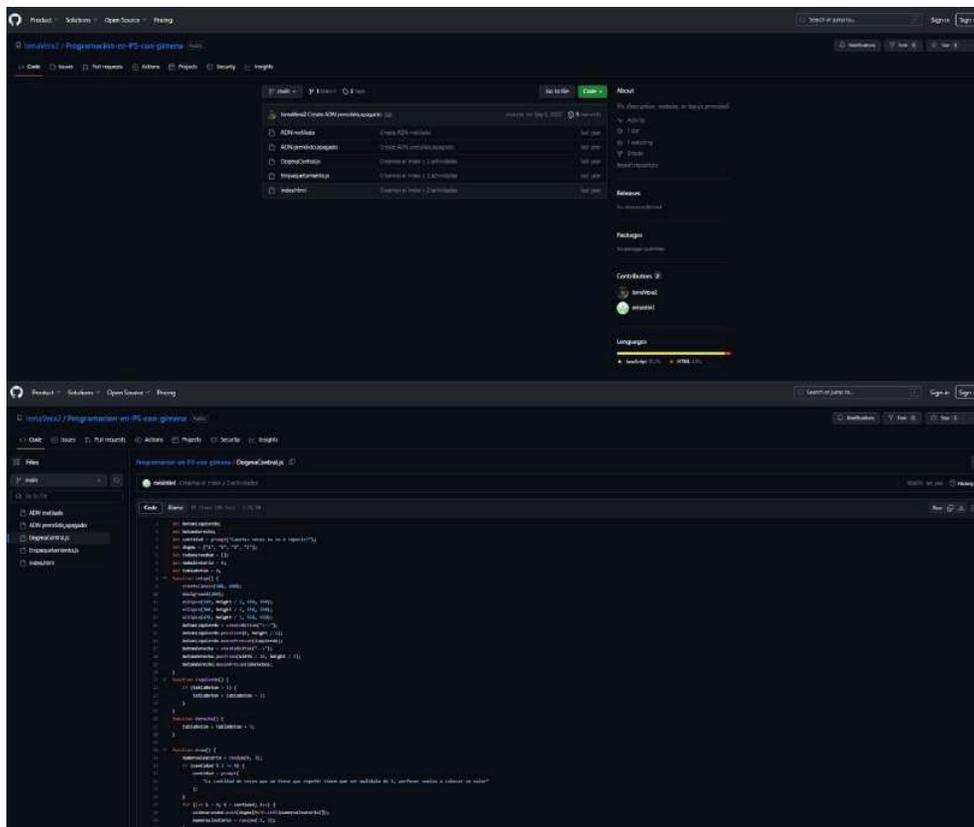


Figura 2: Captura de pantalla de uno de los repositorios de GitHub que las y los estudiantes enviaban por mail lo que permitió realizar un seguimiento semanal de las actividades que realizaban.

Al promediar la secuencia didáctica, se les consultó a las y los estudiantes respecto a las similitudes que podían encontrar entre el código biológico y un código programado en JavaScript en relación a la finalidad/función del código, al proceso de escritura y a la posibilidad de reprogramar el código. En cuanto a la finalidad del código biológico se encontraron respuestas del tipo “*Le indica a la célula que proteína tiene que generar*” mientras que el

propósito de un código en JavaScript es *“Crea funciones y programas”*. Respecto al proceso de escritura del código biológico se indica que *“Su proceso de escritura es A, T, C, G, pero luego de la transcripción la T se transforma en una U”* mientras que en JavaScript *“Su proceso de escritura es a través de arreglos, variables, funciones, etc.”* Para finalizar, en referencia a la reprogramación de los códigos, en el caso del código genético *“El código genético se puede reprogramar utilizando el sistema CRISPR-Cas9”* y en JavaScript *“La reprogramación es al ver algún error revisar el código y ver cómo se podría resolver”*. Las respuestas brindadas por parte del estudiantado permiten inferir, en un primer análisis, que pudieron establecer relaciones entre el código genético y los códigos que se programan en JavaScript. Esto cobra relevancia ya que este lenguaje de programación es uno de los más utilizados ya que se encuentra integrado en todos los navegadores web y aunque tiene sus raíces en el desarrollo web, el conocimiento de JavaScript ahora se puede aplicar a una gran cantidad de otros dominios (Arslan, 2018). De esta manera, se promovería de manera conjunta la alfabetización científica y computacional.

En otra instancia se le preguntó al estudiantado sobre qué habían aprendido a lo largo de las actividades, un 60% de las y los estudiantes hizo referencia a conceptos relacionados a la biología (*“El dogma central de la biología molecular y la epigenética”*), un 30% indicó que aprendió cuestiones relacionadas a la biología y a la programación (*“La metilación, cómo utilizar nuevas cosas de la programación, como funciona nuestro cuerpo con relación a los genes”, “Fases del ADN, sobre cómo se puede relacionar la biología con la programación”*) mientras que el 10% restante hizo alusión a otras cuestiones. Estas respuestas de las y los estudiantes se asemejan a los resultados obtenidos en otros trabajos de investigación (Fussero y Ocelli, 2022) en las cuáles se propusieron la integración de la modelización para el aprendizaje de la ingeniería genética en los cuáles las y los estudiantes pudieron conceptualizar y modelizar. Por su parte, Puttick y Tucker-Raymond (2018) reconocen que tanto la modelización como la abstracción son comunes a las ciencias naturales y a las ciencias de la computación siendo fundamentales para la comprensión de sistemas complejos. Resultados de sus investigaciones indican que las actividades propuestas fomentaron el desarrollo del pensamiento sistémico emergente y la modelización -incluyendo interacciones entre los componentes del sistema- a medida que el grupo de estudiantes programaba sus juegos en relación al cambio climático global, un fenómeno a escala macro pero con similitudes en cuanto a la complejidad del sistema micro que se programó en esta investigación.

### **Consideraciones finales**

La presencia de la tecnología como mediadora de las acciones de la vida cotidiana exige habilidades digitales no sólo para que las personas utilicen dichas tecnologías de modo crítico y

reflexivo sino para que se conviertan en productores de medios digitales. Los resultados preliminares de esta tesis de doctorado indicarían que a partir de las actividades propuestas las y los estudiantes pusieron en juego el desarrollo de conocimientos para su formación científico-tecnológica. Además, como la investigación se estructura desde un esquema metodológico basado en investigaciones basadas en diseño (DBR), esta tesis permite aportar conocimiento teórico y metodológico específico al campo de la didáctica de las ciencias naturales. Por una parte, se generará conocimiento sobre aquellos aspectos didácticos a considerar al momento de diseñar e implementar secuencias didácticas de esta naturaleza. Por otra parte, la teoría emergente del análisis pormenorizado realizado, aporta a reconocer las oportunidades de aprendizaje que se despliegan en función de las características del diseño y del contexto específico. En definitiva, en este tipo de investigaciones se genera conocimiento acerca de cómo se aporta a través de esta sinergia entre las prácticas científicas y la programación a la alfabetización científica y computacional de las y los jóvenes.

### Referencias bibliográficas

- Arslan, E. (2018). *Learn JavaScript with p7.js. Coding for Visual Learners*. Berkeley: Apress.
- Berland, L.K.; Schwarz, C.; Krist, C.; Kenyon, L.; Lo, A.S. & Reiser, B.J. (2016). Epistemologies in practice: Making scientific practices meaningful for students. *Journal of Research in Science Teaching*, 53(7), 1082-1112.
- Crujeiras Pérez, B. & Cambeiro, F. (2018). Una experiencia de indagación cooperativa para aprender ciencias en educación secundaria participando en las prácticas científicas. *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias*, 15(1), 1200-1209.
- Crujeiras Pérez, B. & Jiménez Aleixandre, M.P. (2012). Participar en las prácticas científicas. *Alambique*, 72, 12-19.
- Crujeiras Pérez, B. & Jiménez Aleixandre, M.P. (2015). Desafíos planteados por las actividades abiertas de indagación en el laboratorio: articulación de conocimientos teóricos y prácticos en las prácticas científicas. *Enseñanza de las Ciencias*, 33(1), 63-84.
- Duschl, R.A. & Jiménez Aleixandre, M.P. (2012). Epistemic foundations for conceptual change. En S.M. Carver y J. Shrager (Eds.), *The journey from child to scientist: Integrating cognitive development and the education sciences* (pp. 245–262). Washington, DC: American Psychological Association.
- Erduran, S. & Dagher, Z. (2014). *Reconceptualizing the Nature of Science for Science Education: Scientific Knowledge, Practices and Other Family Categories*. Dordrecht: Springer.
- Fussero, G.B. y Occelli, M. (2022) Construcción de modelos de Ingeniería Genética a través de la programación con Scratch. *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias* 19(2),2802. doi:

10.25267/Rev\_Eureka\_ensen\_divulg\_cienc.2022.v19.i2.2802

- Guisasola J. & Oliva J.M. (2020) Nueva sección especial de REudEDUC sobre investigación basada en el diseño de secuencias de enseñanza. *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias* 17 (3), 3001.
- Jiménez Aleixandre, M.P. & Crujeiras Pérez, B. (2017). Epistemic practices and Scientific practices in Science Education. En K.S. Taber y B. Akpan (Eds.), *Science Education an International Course Companion* (pp. 69-80). Rotterdam: Sense Publishers.
- Jiménez Aleixandre, M.P. & Evagorou, M. (2018). Argumentation in Biology Education. En K. Kampourakis y M.J. Reiss (Eds.), *Teaching Biology in Schools. Global Research, Issues, and Trends* (pp. 263-274). Nueva York: Routledge.
- Justi, R. (2006). La enseñanza de ciencias basada en la elaboración de modelos. *Enseñanza de las Ciencias*, 24(2), 173-184.
- Kalelioğlu, F. & Gülbahar, Y. (2014). The effect of teaching programming via Scratch on problem solving skills: A discussion from a learner's perspective. *Informatics in Education*, 13(1), 33-50.
- Lee, K.D. (2014). *Python Programming Fundamentals*. Londres: Springer-Verlag.
- Lehrer, R. & Schauble, L. (2006). Cultivating Model-Based Reasoning in Science Education. En R.K. Sawyer (Ed.), *The Cambridge handbook of: The learning sciences* (pp. 371-387). Nueva York: Cambridge University Press.
- Lenhard, J.; Küppers, G. & Shinn, T. (2007). *Simulation: Pragmatic Constructions of Reality*. Nueva York: Springer-Verlag.
- Osborne, J. (2014). Teaching Scientific Practices: Meeting the Challenge of Change. *Journal of Science Teacher Education*, 25(2), 177-196.
- Ponte J.P., Carvalho R., Mata-Pereira J., Quaresma M. (2016) Investigação baseada em design para compreender e melhorar as práticas educativas. *Quadrade* 25(2), 77-98.
- Puttick, G. & Tucker-Raymond, E. (2018). Building Systems from Scratch: an Exploratory Study of Students Learning About Climate Change. *Journal of Science Education and Technology*, 27, 306–321.
- Reiss, M.J. (2018). Worldviews in Biology Education. En K. Kampourakis y M.J. Reiss (Eds.), *Teaching Biology in Schools. Global Research, Issues, and Trends* (pp. 263-274). Nueva York: Routledge.
- Rodríguez, P. M. (2019). *Las palabras en las cosas: saber, poder y subjetivación entre algoritmos y biomoléculas*. Buenos Aires: Cactus.
- Vee, A. (2013). Understanding Computer Programming as a Literacy. *Literacy in Composition Studies*, 1(2), 42-64.
- Wenger, E. (2001). *Comunidades de práctica. Aprendizaje, significado e identidad*. Barcelona: Ediciones Paidós Ibérica S.A.

## **La construcción contextualizada de materiales didácticos digitales interactivos, desde una perspectiva de Soberanía Tecnológica**

Flores Perrero Ana Carolina

caro\_rrpp@hotmail.com

UNC - ISFD Nuestra Madre de la Merced

### **Resumen**

Desde el Diseño Curricular del Nivel Primario de Córdoba, se promueve el juego como actividad necesaria para el desarrollo del niño, mencionando que “dado que las tecnologías de la información y la comunicación son, actualmente, herramientas útiles para acceder a un conocimiento científico actualizado, es necesario incluir actividades que las contemplen. Inicialmente, se recomienda trabajar con juegos interactivos”.

Es factible observar cómo docentes al momento de implementar un recurso interactivo, recurren a plantillas existentes en sitios web como Educaplay o Wordwall.

Los testimonios conducen a pensar que hay un divorcio, o por lo menos ausencia de diálogo, entre lo que necesita el Nivel Primario y la formación en TIC de los docentes de un Nivel Superior (específico), agentes seleccionados para esta innovación.

De lo expuesto, se deriva el planteamiento de la propuesta:

La construcción y deconstrucción contextualizada de materiales didácticos interactivos digitales, desde una perspectiva de Soberanía Tecnológica. Un caso de estudio de vinculación entre estudiantes de Tercer año de un Profesorado de Educación Primaria y estudiantes de Nivel Primario.

Para el logro del fin propuesto, se selecciona Scratch por tratarse de un software libre destinado al desarrollo de juegos interactivos. Su filosofía es de "piso bajo, techo alto", con el cual es fácil empezar, llegando a concretar proyectos de alta calidad.

De este planteamiento del tema, se deriva la pregunta principal del problema de innovación: ¿por qué es importante recurrir a la soberanía tecnológica al momento de diseñar materiales didácticos digitales interactivos en Nivel Primario?

Debido a la profunda transformación impulsada por la emergencia de la cultura digital, como docentes, nos convertimos en necesarios mediadores entre el mundo y el estudiante en lo que hace a la incorporación de la tecnología, recurriendo a la triangulación contenido-contenedor-contexto. Puesto que como expresa Cobo (2016) “Cuando las TICs son

adoptadas en un contexto que estimula y reconoce la creación individual y colectiva, pueden contribuir de manera mucho más consistente a enriquecer los procesos de aprendizaje” (p.59).

**Palabras clave** prosumidor, Nivel Primario, Soberanía Tecnológica, Scratch, formación docente, deconstrucción.

### Introducción

¿Qué sucede cuando el docente no encuentra un “recurso” digital interactivo contextualizado a su grupo de estudiantes?

Es factible observar cómo docentes de Nivel Primario, al momento de implementar un recurso interactivo, recurren con frecuencia a plantillas existentes en sitios web como Educaplay o Wordwall.

Por otro lado, la propuesta surge de la preocupación de la directora del Instituto de Nivel Superior quien expresa: “lo que estamos necesitando es que los estudiantes (de nivel terciario) tengan herramientas para poder trabajar en virtualidad, que les permitan generar recursos propios y habilite a los chicos a aprender mejor”.

Es así que se advierte la necesidad del Nivel Superior de formar docentes con soberanía tecnológica, capaces de generar materiales educativos digitales interactivos propios para el grupo clase en el que se están desempeñando. Por el otro, a los niños, al ejecutar materiales no creados por ni para ellos, se les está quitando la posibilidad de desarrollar mayores habilidades en la computadora, adecuadas a cada nivel. En última instancia, todos están renunciando con ello a la soberanía tecnológica que, como miembros activos de la sociedad, podrían ejercer.

De lo expuesto, se deriva el planteamiento del tema:

La construcción contextualizada de materiales didácticos interactivos digitales, desde una perspectiva de Soberanía Tecnológica. Un caso de estudio de vinculación entre estudiantes de Tercer año de un Profesorado de Educación Primaria y estudiantes de Nivel Primario.

De este planteamiento del tema, se deriva la pregunta principal del problema de innovación: ¿por qué es importante recurrir a la soberanía tecnológica al momento de diseñar materiales didácticos digitales interactivos en Nivel Primario?

A partir de la/s respuesta/s vinculada/s con esa pregunta, surgen otros interrogantes que la complejizan y profundizan:

- ¿De qué manera se puede articular el futuro docente con el estudiante de Nivel Primario

desde una perspectiva de Soberanía Tecnológica?.

- ¿Permite la programación en Scratch promover el concepto de soberanía tecnológica?

De las preguntas planteadas, surge el objetivo principal de esta propuesta:

Concientizar sobre la aplicación de soberanía tecnológica en la construcción de materiales didácticos digitales interactivos en Nivel Primario, a través del uso de software libre.

En el presente trabajo, se podrá observar cómo, a través de los recursos creados por las estudiantes de Nivel Superior, es posible la formación de un docente que, recurriendo al modelo TPACK, alcance soberanía tecnológica para generar contenedores en forma de recursos didácticos interactivos, adecuados a cada contexto, con el fin de realizar prácticas significativas, innovadoras, adaptadas al ámbito en el cual el estudiante de nivel primario deberá desarrollar sus potencialidades, en un mundo donde lo digital cobra cada día mayor preponderancia.

El estudio que este trabajo se propone parte de una definición de paradigma sociocrítico interpretativo. En efecto, se trata de que el docente, desde el período de su formación de grado como Profesor de Educación Primaria, vaya asumiendo su rol y tomando conciencia de la necesidad de ser prosumidor digital con soberanía tecnológica en el futuro ejercicio de su profesión.

En este paso a docente prosumidor, junto a esta innovación se busca que el estudiante de nivel superior aprenda haciendo, a través de una expansión del aula tradicional hacia una práctica real con TIC, en donde ejecute una actividad concreta vivida en primera persona, como parte de una enseñanza creativa apuntando a generar docentes digitales, no analógicos.

Se pretende que al crear sus propios recursos adquiera nuevas estrategias pedagógicas que generen impacto y motivación en los estudiantes de nivel primario, que podemos considerar nativos digitales, generando ambientes formativos donde sean capaces de crear y construir, pasando a ser productor y consumidor de contenidos (Merellano-Navarro, et.al, 2016, p. 939).

En relación con lo expuesto, la metodología de incorporación de TIC en el aula que sustenta esta innovación es la TPACK, método acuñado por Mishra y Koehler (2006) junto con Schmidt, Baran, Thompson y Shin (2009), es un modelo en el que se trata de incluir los conocimientos relacionados con los contenidos disciplinares que son llevados a la práctica a través de la pedagogía y la tecnología, el mismo, como mencionan Alcalá, L y Cuevas, O (2017), "se apoya en la idea formulada por Shulman (1986) sobre el Conocimiento Pedagógico del Contenido (PCK), en el cual los docentes deben poseer conocimientos relacionados con el contenido de las materias que imparten, así como conocimientos en pedagogía", y, en el caso del modelo TPACK, se le suma el conocimiento Tecnológico de manera interrelacionada con el resto de los

contenidos.

Sin embargo, más allá de las críticas, la metodología TPACK, se considera que merece una especial valoración en relación con la propuesta de innovación que se está llevando a cabo en este estudio. De hecho, supone un docente dispuesto a investigar para utilizar tecnologías a partir del uso pedagógico y curricular del contenido de manera contextualizada y procura incentivarlo para convertirse en prosumidor, con un fin democratizador de un conocimiento con fuerte presencia digital.

“La soberanía es un concepto que determina la posibilidad que tiene un pueblo para ejercer la máxima autoridad en un territorio determinado. En el ámbito de la soberanía tecnológica, estas iniciativas tienen por intento generar opciones ante aquellas tecnologías que son comerciales” (Carrai P., 2020)

Desde este punto de vista, con esta innovación orientada al área tecnológico-pedagógico-didáctica en el Nivel superior, se pretende formar a un docente con soberanía tecnológica que trabaje el contenido que se le asignó con un recurso creado por él mismo, no por otro, ni siquiera con una plataforma que le exige una plantilla propia y hasta a veces paga de la empresa que lo programó, se busca que el docente cree, construya e innove sus propios recursos, a partir del contexto en el que estará inserto. Para ello debe contar con buena predisposición a fin de lograr su empoderamiento, conociendo nuevas herramientas que le permitan construir recursos.

Por su parte, el dossier Ritimo, hace alusión a que una de las formas de lograr Soberanía Tecnológica es recurrir a la utilización de Software Libre, por las libertades que este nos ofrece; en relación con él, cita a Richard Matthew Stallman, creador del concepto y fundador de la Free Software Foundation.

Volviendo a la innovación que nuestro trabajo propone, se considera necesario, a fin de lograr el empoderamiento con soberanía tecnológica, un docente investigador, tal como lo venimos mencionando, inquieto y partícipe activo de estas propuestas que le permiten mantenerse actualizado con respecto a la temática y formar criterios para las decisiones tecnológico-pedagógico-didácticas que diariamente toma como curador de contenido.

Se elige Scratch como el software libre que le permitirá ejercer soberanía tecnológica al docente, al momento de crear sus recursos.

El mismo utiliza un lenguaje de programación basado en objetos y uno de los pioneros de este lenguaje es Alan Kay, quien fomenta esta manera de programar, luego de estudiar durante muchos años el aprendizaje de los niños.

Kay, preocupado por la subutilización de las computadoras en las escuelas, afirma, “creo que el gran problema es que las escuelas tienen muy pocas ideas sobre qué hacer con las computadoras una vez que los niños las tienen”. (p.2)

Por otro lado defiende: “El juego es el mecanismo integrado de la naturaleza para el aprendizaje más profundo del niño”. Esta metodología del juego es la que se propone generar a estudiantes de Nivel Terciario para trabajar el contenido con sus estudiantes

Es importante mencionar que Alan Kay, cuenta en su formación con dos referentes claves: Jerome Bruner y Seymour Papert.

A partir de estas ideas, reconociendo la importancia de la programación orientada a objetos como una actividad fundamental que deberían realizar los niños desde pequeños, surge la propuesta de trabajar con Scratch, uno de los software de programación, libres por excelencia, posible de implementarse en Nivel Primario

Scratch es un lenguaje de programación por bloques, programación orientada a objetos. Posee una interfaz gráfica muy simple y de fácil manejo que facilita el aprendizaje autónomo y el desarrollo de la creatividad.

Asimismo “como su propio nombre indica, los usuarios de Scratch pueden reutilizar las partes creadas por otros usuarios, para crear nuevos proyectos y de esta forma aprovechar el potencial del trabajo colaborativo” (Resnick et al. 2012).

### **Aspectos Metodológicos**

Una vez delineados los antecedentes y conceptos que sustentan esta propuesta, y sabiendo que el estudio de caso de manera intrínseca se dirige hacia estudiantes de Nivel Superior, quienes aplicarían su práctica a estudiantes de Nivel Primario, se esboza la metodología aplicada.

Sujetos:

La elección de los sujetos surgió en primer lugar de la necesidad de la Directora de un Instituto de Formación Superior.

Se consideró relevante realizar un diagnóstico sobre estos sujetos a fin de evidenciar el problema inicial planteado.

Por otra parte, también se requirió de la participación de estudiantes de Nivel Primario, con quienes también se realizó un diagnóstico a fin de contextualizar los recursos didácticos interactivos elaborados.

También fue necesario contar con sujetos especializados en las diferentes áreas para las que se crearían los recursos: Docentes de nivel primario, docente de práctica y docentes de materias específicas (Matemática, Lengua, Ciencias Sociales y Ciencias Naturales)

Procedimiento:

El estudio se basó en tres etapas:

Primera etapa; previa a la inserción en las escuelas de nivel primario.

La primera etapa realizada en el gabinete de computación del Nivel Superior e incluyó diversas actividades de sensibilización para la incorporación de las TIC en el aula, conceptos anteriormente desarrollados, que ayudan, progresivamente, a los futuros docentes a asumir una posición de soberanía tecnológica.

Cabe destacar que la modalidad de cursado en Nivel Superior a partir de la pandemia es de aula extendida, es decir, se realizan encuentros en modalidades presenciales físicos en el edificio escolar y semana de por medio presenciales sincrónicos y asincrónicos.

Segunda etapa, inserción de las estudiantes en las escuelas asociadas

Hasta aquí la primera parte del proyecto, sin necesidad de contar con otros recursos humanos en la innovación, solamente las estudiantes del Nivel Terciario.

A partir de ahora se propone actividad integrando el resto de los recursos. Empezó a desarrollarse en el momento que se ingresó al aula de Nivel Primario.

Es cuando se les invitó a partir del contenido que les pide la docente de la escuela asociada; programar un juego que deberán implementar en el aula y que para favorecer en los niños el desarrollo de las capacidades fundamentales, el recurso deberá ser deconstruido en el aula junto a ellos, para finalizar su programación.

En este momento el trabajo empezó a ser cada vez más ubicuo.

Tercera etapa, después de la inserción

Por medio de una bitácora o narrativa personal las estudiantes de Nivel Superior, realizaron una evaluación del proceso realizado analizando los conceptos incorporados.

Cabe destacar que la evaluación se fue realizando desde el comienzo del proceso a modo de evaluación formativa puesto que se evaluó cada grupo, se evaluó también la planificación y el proceso de diseño en conjunto con las estudiantes de Nivel Terciario.

Técnicas e instrumentos:

Si bien es una innovación con participación, se decidió que los instrumentos de recolección de datos fueran en primer lugar el aula virtual, donde quedó depositado el avance que iban realizando las estudiantes.

Por otro lado, en el momento de implementación en el aula de Nivel Primario, el registro mediante imágenes, fotografías, videos y grabaciones de audio que permitió el posterior análisis.

También se realizaron observaciones a los participantes para analizar luego su interacción.

Asimismo, en la tercera etapa donde se les solicita una bitácora o narrativa personal del proceso realizado. La misma debía ser construida a lo largo del proceso y entregada el último día por parte de las estudiantes del Nivel Terciario.

### **Análisis**

Una vez implementada la innovación, corresponde ahora realizar un análisis de todo lo acontecido.

Análisis del contexto

Respecto al diagnóstico de las estudiantes de Nivel Superior nos encontramos con que futuras docentes con muchas ganas de incorporar las TIC pero poca y casi nula idea de Software Libre y la importancia de la Soberanía Tecnológica.

Con el diagnóstico realizado desde adentro de las escuelas primarias por las estudiantes de Nivel Superior, se supo que había conexión a internet aunque fluctuaba según el espacio en el que se encontraba el aula. Además, también se conoció que en una de las escuelas utilizaban Scratch en la materia Informática, con contenidos propios sin transversalidad.

En cambio, en la otra, a partir de lo que observaron las mismas practicantes en el momento que estaban insertas en las instituciones, no se le daba mayor importancia a las TIC, simplemente recurrían al celular de los estudiantes cuando necesitaban buscar alguna información. Esto se pudo constatar personalmente cuando se realizaron observaciones a las estudiantes y las docentes de grado manifestaban que era complicado realizar propuestas con TIC.

Asimismo, se destacó que en una de las escuelas tenían horarios flexibles, las docentes eligen el momento en que los estudiantes se toman el recreo.

Este diagnóstico que hicieron las propias estudiantes les sirvió para ver cómo era cada grupo,

una de ellas con mucho temor al llegar a la clase de TIC manifestaba: “Profe mi grupo es terrible, no se si lo voy a lograr”

Análisis que realizaron los estudiantes que aplicaron la innovación

Es tan rico el material que volcaron en sus bitácoras las estudiantes de nivel superior cuando realizaron el análisis de su práctica, que merece la pena destacar resumidamente algunas de ellas, en tanto que, por su objetividad, brinda un panorama claro de lo que sintieron las principales afectadas en esta innovación.

En todos los casos se facilitó la construcción de materiales didácticos interactivos digitales contextualizados, mediante la incorporación de software libre en la propuesta didáctica, desde las primeras actividades planteadas, en las que se les hizo hincapié en el proceso de diseño, al respecto las estudiantes expresan:

Estudiante 1:

En cuanto a la planificación, el proceso de realización fue todo un desafío. En un primer momento me costó llevar a cabo la utilización del software elegido, es decir, se me dificultó la comprensión del mismo. Considero que lo que me ayudó fue ponerme en acción, comenzar a investigar las herramientas que el recurso me brindó y una vez que pude interpretarlas, me puse en práctica para la realización del juego digital.

La etapa previa de realización, fue llevar a cabo el boceto, el mismo fue muy relevante para luego construir el recurso, ya que nos permitió plasmar nuestra idea, aquello que íbamos a utilizar para después construir la idea final. Con respecto al software elegido para la construcción del recurso: Scratch, me pareció una herramienta novedosa y a su vez, desconocida para mí.

Como manifiesta esta y varias estudiantes, al ser un software nuevo, generó distintos sentimientos, pero a medida que fue avanzando el tiempo, tuvieron mayor confianza y pudieron llevar adelante la innovación pretendida. Otra estudiante expresa al respecto:

La planificación fue muy útil, ya que ayudó a organizar de qué manera se irían proyectando los botones y las imágenes en pantalla. La construcción del juego interactivo fue un proceso largo. Al tratarse de una herramienta didáctica-digital nueva, su uso, su interiorización llevó horas de producción, diversas formas de idealizar la composición de bloques, diferentes intentos, con errores, aciertos y con sugerencias brindadas por la docente del taller. Luego de lograr una apropiación y comprensión del lenguaje que prevalece en la aplicación, pude llevar a cabo la construcción del Scratch para los estudiantes de 6to grado, quienes trabajaban con el contenido Interculturalidad en América Latina.

En esta estudiante, se manifiesta claramente todo el proceso que lleva la programación desde la descomposición pasando por la ejercitación con pruebas y errores y el proceso de diseño que, siguiendo sus pasos desde la planificación, facilitó la concreción del recurso didáctico. Finalmente y con respecto al mismo objetivo otra estudiante señaló:

Al momento de practicar la programación, lo que más me fue útil fue la curación de contenido, el boceto (aunque muchas veces me olvidaba de él), y la previa capacitación sobre programas de libre software para realizar mis programaciones. Uno de los errores que tuve fue programar sin antes tener una planificación detallada, lo cual la profe corrigió y me explicó por qué era mejor planificar, realizar el boceto y luego ir a la puesta en práctica. El programa que más me gusto fue Scratch, porque te brinda la oportunidad de crear ilimitadamente, de explorar las diferentes creaciones de miles de personas y aprender de ellos, con la opción "ver dentro, se puede observar que códigos utilizaron y combinaron en sus producciones", a mi parecer esto les da la oportunidad a los usuarios de conocer una manera diferente de crear y muchas veces, cuando uno tiene un bloqueo, sirve de gula. Y por último, pero no menos importante, el programa permite la opción de remix, en donde podemos crear en base a códigos ya establecidos por otros usuarios, dándonos la oportunidad de añadir o modificar estructuras ya predeterminadas.

Respecto a la incentivación de los estudiantes de Nivel Primario en la producción de materiales didácticos digitales interactivos lo cual debían lograrlo las estudiantes de Nivel Superior, para lo cual también hubo que prepararlas y animarlas. Se puede decir que el objetivo se logró a través de las palabras que las mismas estudiantes de nivel superior expresan:

...cada clase les preguntaba a los estudiantes ¿Conocen Scratch? ¿Alguna vez lo utilizaron? ¿Pudieron averiguar algo?", me encontré con diversas respuestas por parte de ellos: muy pocos no conocían, la mayoría conocían pero no lo habían utilizado, y dos de ellos lo conocían a la perfección ya que habían realizado cursos de programación. A medida que iban pasando las clases, ellos mismos me comentaban que habían estado investigando y probando el programa.

El motivarlos en unas clases previas al trabajo en Scratch me sirvió para que ese día todos supiesen lo que estábamos realizando, ya que en la Institución no tienen la materia de TIC y los recursos tecnológicos son escasos.

Otra estudiante dijo:

Comencé la clase diciéndole a los estudiantes que haríamos un repaso de las culturas de América Latina jugando en un programa de Scratch, pero que había un problema... la señora había dejado una parte del juego sin programar y necesitaba de la ayuda de ellos para

finalizarlo.

Primero jugamos para que los estudiantes comprendan cómo era la dinámica, luego, al llegar a la pregunta final, los botones no respondían y es allí cuando debimos comenzar la parte de la deconstrucción entre todos.

Cuando empezamos a deconstruir yo guiaba a los estudiantes y les preguntaba qué bloques creían que debíamos colocar, pero luego de un rato ellos mismos eran quienes programaban por su cuenta sin necesidad de que intervenga. Los estudiantes pedían pasar a la computadora para arrastrar ellos mismos los bloques, cuando planifiqué mi clase no pensé en que eso fuera a pasar, pero en ese momento me pareció una idea genial, por lo que algunos pasaron y pudieron manejar el programa. Algunos comentarios e intervenciones que los estudiantes hacían fueron: "Seño, ¿por qué no copias y pegas todo el bloque?" "Los botones que dicen control deben ser para controlar algo" "En ese botón tenemos que poner el sonido de correcto porque pusimos el color verde" "¿La opción de preguntas la hiciste con una variable seño?" "Ahh ponemos que envíe mensaje final porque ya no hay más preguntas

Cuando finalizamos el momento de la deconstrucción, los estudiantes querían pasar a jugar al juego que hablamos programado entre todos: lo que hicimos fue que comiencen a pasar de a dos para que todos puedan jugar y comprenderlo.

Lo que demuestra que los estudiantes de ambos niveles educativos terminaron muy motivados aprendiendo las ventajas de construir su propio material didáctico. Finalmente otra estudiante expresó:

Una vez presentado el recurso, los estudiantes manifestaron mucho entusiasmo y curiosidad, incluso algunos, sin necesidad de explicación previa, habían comprendido el funcionamiento del mismo. En este punto todos mis miedos desaparecieron. Siguiendo mis indicaciones todos los estudiantes participaron de la desconstrucción del recurso, que se basó en crear una pregunta entre todos a partir del contenido dado, y a su vez, realizar las respuestas de las mismas. Algunas respuestas que recuerdo, son las indicaciones que ellos me daban para construir una programación libre, debido a que nos sobraba tiempo, les propuse crear lo que se les ocurriera en el momento. En base a la consigna, creamos todos juntos un rap sobre el universo, les entusiasmo mucho la idea de agregar disfraces y accesorios a los personajes, incluso le pusimos música de fondo. Fue una experiencia muy gratificante para ambos, y sin dudas, superadora para mí.

Observaciones vinculadas con las estudiantes de Nivel Superior.

El testimonio propio de las estudiantes que aplicaron la innovación significó una gran

satisfacción en tanto el objetivo del trabajo estaba logrado. Se considera fundamental sintetizar a continuación algunos puntos claves que observó el docente innovador, para este trabajo:

Cuando se les presentó Scratch surgieron diversas opiniones, tal como fue señalado en el testimonio de las mismas estudiantes.

Quienes en su escuela secundaria tuvieron computación, se mostraron más ágiles en descubrir por sus propios medios para qué servía cada parte. Asimismo, se reflejó en esta instancia el aprendizaje compartido, colaborativo entre las dos estudiantes que lo conocían y quienes no, el aprendizaje del programa se iba transmitiendo de compañero en compañero.

Por otra parte, advirtieron que podían encontrar ayuda también en videotutoriales.

La presencia del docente de TIC en esta instancia fue sobre todo motivadora, guía, eran fundamentales las preguntas que se les hacía por ejemplo ¿si quiero que el personaje se mueva, qué bloque podré utilizar? y si quiero que sume puntos ¿qué tendré que hacer? ¿Si quiero guardarlo a donde iré? ¿y ahora, como hago para que avance más?, con estas preguntas se generaban situaciones problemáticas que debían separar en partes, abstraer, y recurrir a la lógica para poder resolverlo. Preguntas que servían además para explorar de manera progresiva el entorno.

Se recurrió también aquí a la deconstrucción que luego tendrían que hacer con sus estudiantes de nivel primario, puesto que la docente de TIC presentaba algunos ejemplos de lo que se podía hacer, más allá de lo que ya habían visto en el video del “niño programador” (<https://www.youtube.com/watch?v=9hUjhIfs-bw>).

Ejemplos que los estudiantes de Nivel Superior debían deconstruir para comprender cómo estaba hecho, por ejemplo, preguntas y respuestas.

Hubo que indicarles en este momento, la existencia de la comunidad de Scratch, donde hay recursos compartidos por otros programadores, de esta forma también aprendieron a construir y compartir su recurso.

Poco a poco empezaron a “aprender haciendo” los beneficios de un software libre, empoderándose como docentes cada vez más a través de la soberanía tecnológica.

El momento más tenso de la innovación, fue la segunda etapa, cuando ya con los conceptos incorporados debían crear sus recursos.

Por un lado las estudiantes ya no trabajaban en grupo, algunas tenían pares pedagógicos pero no todas porque estaban asignadas a distintos grados.

Por otro lado, estaban con el temor de si iban a poder lograrlo o no, el mismo temor anticipado por la Inspectora Sandra López, quien con gran entusiasmo participó de un Taller de esta segunda etapa.

A medida que iban avanzando los días, el trabajo de la docente de TIC con las estudiantes era cada vez más individualizado, por tratarse de distintos recursos.

Lo que debían tener en cuenta era que no podían ponerse a programar si la planificación y el boceto no estaban aprobados.

Algunas estudiantes se pusieron a trabajar desde el momento en que les asignaron el tema, otras, lo dejaban para el final. Algunas programaban antes de que estuviera aprobada la planificación. Otras, tardaban en confeccionar la planificación.

Al momento de implementarlo se presentaban diversas situaciones que implicaron un acuerdo respecto de su resolución:

Se decidió que las estudiantes que no presentaron su planificación a tiempo, no continúen adelante con la innovación, por la falta de tiempo para programar el recurso adecuadamente y además, por su falta de compromiso con lo asumido.

Estas decisiones fueron tomadas a fin de favorecer el objetivo principal que busca esta innovación, es decir, un empoderamiento de su parte en el trabajo con las TIC en orden a adquirir soberanía tecnológica en su rol.

Quienes siguieron las indicaciones lograron que los estudiantes participaran activamente en la deconstrucción del recurso.

Quienes estaban en la escuela de tiempo dispuesto por el docente, lograron realizar la actividad de manera más completa, con participación de todos los estudiantes, llegando a construir su conocimiento tanto respecto del contenido solicitado por la docente como en la programación.

En la otra escuela, donde los módulos son de 40 minutos, se observó en las estudiantes una tendencia a apurarse para llegar con tiempo.

Asimismo, algunas estudiantes decidieron dividir la actividad en dos días, en uno de ellos se presentaba la herramienta y se invitaba a los niños a explorar en sus hogares, en el siguiente, se la aplicaba con el contenido que se le había asignado.

## **Conclusión**

En primer lugar es importante destacar que el camino iniciado, forma parte de una Tesis de Maestría; en la cual se puede percibir con mayor extensión de testimonios y datos relacionados al trabajo realizado.

Con lo aquí expuesto se puede apreciar que los objetivos se han conseguido.

De hecho que, a partir de esta propuesta, hay muchas posibilidades y perspectivas de análisis que permitirían extender y mejorar este proyecto de innovación.

No obstante, queda demostrado que se puede satisfacer una novedosa necesidad en la formación de formadores, como es el hecho fundamental del trabajo que la soberanía tecnológica nos permite realizar con los estudiantes a partir del diseño compartido de recursos didácticos interactivos.

Cabe destacar que, para que el recorrido fuera posible, se presentaron factores determinantes:

En primer lugar, estudiantes de Nivel Superior con deseo de movilizarse, aprender e impulsar un cambio de la realidad a partir de lo analizado. Estudiantes capaces de investigar para poder generar piezas comunicativas, en este caso recursos interactivos significativos para el aprendizaje del niño.

Instituciones y personal de ambos niveles, abiertas y dispuestas a acompañar el proceso de innovación con la convicción de que un mejor camino es posible y necesario, el de la producción y no solo del consumo de recursos digitales,

Estudiantes de nivel primario inquietos, capaces de ejercitar el pensamiento, con deseos de jugar, compartir y programar para aprender.

Un software libre por excelencia como Scratch, que permitió sortear los diferentes obstáculos que se podrían presentar como la escasa conectividad en las escuelas.

Tal como lo expresaba Alan Kay, se manifestó muy claramente la facilidad con que los niños aprenden a programar con el software seleccionado, por el trabajo colaborativo que se les propuso, lo que facilitó que el aprendizaje sea compartido y entre todos fueran construyendo el recurso.

Lo expuesto se advirtió en ambas instancias, tanto en la formación de los estudiantes de nivel superior como en los escolares del primario.

Fue interesante ver el modo en que a partir de una misma consigna, cada practicante fue guiando a sus estudiantes para llegar al objetivo final, adaptando la propuesta al contexto grupal, logrando mejores resultados la que más libertad de pensamiento y co-construcción, les

ofreció a sus estudiantes.

Esto quizás se dio por ser la primera vez en que los niños se enfrentan a esta situación o, tal vez, por la misma confianza de la docente respecto de sus estudiantes para llegar a resolver la situación problemática.

La experiencia vivida alienta a continuar aplicando la metodología de trabajo en años posteriores del cursado del profesorado (no sólo en tercer año), al igual que en el trabajo diario con los niños de nivel primario, trabajando de manera integral tal como lo propone el modelo TPACK tecnología, pedagogía y desarrollo curricular.

Asimismo, la misma propuesta se está realizando durante el presente año con los mismos objetivos y algunas adecuaciones en su metodología, en el profesorado de Nivel Inicial.

Para finalizar, se considera que es fundamental en todo este proceso, preservar en alto la bandera de la Soberanía Tecnológica que, como docentes, podemos disfrutar y ejercer si somos conscientes de su presencia fundamental en las aulas, para lograr que el afuera y el adentro de la escuela funcionen de manera interrelacionada.

### Referencias bibliográficas

- Alcalá, L. y Salazar, O. (2017) *Adaptación del modelo TPACK para la formación del docente universitario*. Congreso Nacional de Investigación Educativa. San Luis Potosí <https://comie.org.mx/congreso/memoriaelectronica/v14/doc/2150.pdf>
- Brennan, K. y Resnick, M., (2012). *Nuevos marcos de referencia para estudiar y evaluar el desarrollo del pensamiento computacional*. American Educational Research Association (AERA). <http://eduteka.icesi.edu.co/pdfdir/EvaluarPensamientoComputacional.pdf>
- Carrai, P (2020) *¿Por qué es importante la soberanía tecnológica?*. Educ.ar <https://www.educ.ar/recursos/155423/por-que-es-importante-la-soberania-tecnologica#:~:text=La%20soberan%C3%ADa%20es%20un%20concepto.el%20ejercicio%20de%20esta%20autoridad>.
- Cobo, C. (2016). *Innovación Pendiente*. Penguin Random House. Madrid. España
- Consejo Federal de Educación (2018). *Núcleos de Aprendizajes Prioritarios*. Argentina. <https://www.educ.ar/sitios/educar/resources/150123/nap-de-educacion-digital-programacion-y-robotica/download>
- Kay, A. (2007). *A powerful idea about ideas*. TED. [https://www.ted.com/talks/alan\\_kay\\_a\\_powerful\\_idea\\_about\\_ideas?fbclid=IwAR1NV3brxdgGO4ODpiF7D8CAOtaMjaG6\\_4rAs4b\\_Ot\\_mGtGLnOcV2UJR P8 U](https://www.ted.com/talks/alan_kay_a_powerful_idea_about_ideas?fbclid=IwAR1NV3brxdgGO4ODpiF7D8CAOtaMjaG6_4rAs4b_Ot_mGtGLnOcV2UJR P8 U)
- Ministerio de Educación de la Provincia de Córdoba (2011). *Diseño Curricular de la Educación Primaria*. <https://www.igualdadycalidadcba.gov.ar/SIPEC>

CBA/publicaciones/EducacionInicial/DCJ%20EDUCACION%20INICIAL%  
web%208-2-11.pdf 20

Ministerio de Educación de la Provincia de Córdoba(2015). *Diseño Curricular del Profesorado de Educación Primaria de la Provincia de Córdoba*. [https://dges.cba.infed.edu.ar/sitio/upload/Disenio\\_Curr\\_Primary\\_Inicial\\_2015.pdf](https://dges.cba.infed.edu.ar/sitio/upload/Disenio_Curr_Primary_Inicial_2015.pdf)

Riemens, P (2014). *Soberanía Tecnológica. Dossier Ritimo*. <https://www.ritimo.org/IMG/pdf/dossier-st1-es.pdf>

The Dynabook Revisited (2002). *A Conversation with Alan Kay. The Book & The Computer*. [www.squeakland.org/resources/articles/article.jsp?id=1007](http://www.squeakland.org/resources/articles/article.jsp?id=1007)

Unesco (2020). *¿Qué se espera que aprendan los estudiantes de América Latina y el Caribe?*. <https://unesdoc.unesco.org/ark:/48223/pf0000373982>

## **Propuestas de modelización para fortalecer el desarrollo profesional de docentes de Computación, Matemática y Educación Tecnológica.**

María Emilia Echeveste <sup>1,2</sup>, Araceli Coirini <sup>2</sup>, Lourdes Aguiar Cau <sup>2</sup>, Jonathan Alonso <sup>2</sup>, Nicolás Balmaceda <sup>2</sup>, Érica Zarate <sup>2</sup>, Talia Monjes <sup>2</sup>, Fabricio Carinelli <sup>3</sup>  
*emilia.echeveste@unc.edu.ar, araceli.coirini@unc.edu.ar, lourdes.aguiar.cau@mi.unc.edu.ar, jonymalonso@gmail.com, nicolas.balmaceda@mi.unc.edu.ar, erica.zarate@mi.unc.edu.ar, talia.monjes@mi.unc.edu.ar, fabricarinelli@mi.unc.edu.ar*

<sup>1</sup> CONICET

<sup>2</sup> *Grupo de Enseñanza de la Ciencia y la Tecnología  
Facultad de Matemática, Astronomía, Física y Computación  
Universidad Nacional de Córdoba*

**Palabras claves:** Modelización- Educación en Ciencias de la Computación- Formación profesional

### **Resumen**

En una sociedad cada vez más digitalizada, es necesario identificar estrategias que favorezcan el acceso al conocimiento y la adquisición de competencias que permitan un desarrollo profesional docente que garantice la formación de una ciudadanía crítica, creativa y con habilidades cognitivas acordes a los desafíos del siglo XXI. En esta línea, la modelización, como abordaje pedagógico, nos permite pensar en estrategias para llevar a cabo propuestas de aula que desarrollen un pensamiento computacional y matemático y, al mismo tiempo, abordar problemáticas del mundo real en sinergia con el empleo de tecnologías digitales (TD).

Como parte de una propuesta de investigación recientemente financiada por MINCYT y SECYT-UNC, presentamos los inicios de un proyecto de investigación-acción participativa que configura, analiza y resuelve un problema de investigación con actores que están involucrados en él, agrupando en la acción de investigar aquellos hechos que los/las afecte intentando generar transformaciones (Gellert et al., 2013). Se busca recopilar información de contextos educativos reales -cinco escuelas secundarias de gestión pública y privada de la Ciudad de Córdoba- en donde se ponen en juego prácticas educativas de modelización mediadas por TD de forma colaborativa entre docentes, futuros docentes e investigadores para las áreas de Computación, Matemática y Educación Tecnológica.

En esta oportunidad desarrollaremos el comienzo del trabajo realizado en las dos instituciones educativas en las que se trabaja con docentes de la orientación en Informática. Ambas escuelas contaban con ideas previas vinculadas a proyectos educativos; lo que se busca con esta iniciativa de modelización es generar una organización y sistematización de esa propuesta

didáctica a los fines de formalizar las acciones realizadas en las aulas y evaluar su alcance dentro de esa práctica situada con proyección a favorecer el desarrollo de aprendizajes significativos.

### **Introducción**

Datos registrados en Pisa 2018 (OCDE, 2019), procesados por el Banco Internacional de Desarrollo - Centro de Información para la Mejora de los Aprendizajes (BID-CIMA), evidencian en América Latina la existencia de disparidades en las habilidades digitales de docentes entre países, tipos de escuela y entornos socio-económicos. Se registró que menos del 60% de los docentes de secundaria presentaron habilidades técnicas y pedagógicas para integrar dispositivos digitales en procesos educativos (Lugo et al., 2020). En lo que respecta a las Ciencias de la Computación (CC), en los últimos 10 años se iniciaron propuestas de formación docente en estos temas, encabezadas principalmente por la Fundación Sadosky con iniciativas de fuerte impacto social, tales como Program.AR, que busca llevar la enseñanza y el aprendizaje de las CC a las escuelas argentinas. A su vez, se generaron espacios de formación docente, como en el caso particular de Córdoba, con la creación de la Tecnicatura en Desarrollo de Software en el Instituto de Educación Superior "Simón Bolívar"; la Especialización Docente de Nivel Superior -que se dictó en conjunto entre FAMAF-UNC, el Ministerio de Educación de la Provincia de Córdoba y la Fundación Sadosky- o espacios de formación continua en informática a cargo del Instituto de Estudios Pedagógicos (ISEP).

La Educación en Ciencias de la Computación, como disciplina novel dentro de este campo educativo, aún se encuentra re-construyendo su didáctica en el área, la cual se origina en los años 60 con los trabajos de Papert utilizando plataformas computacionales como LOGO, en donde se potenciaba el desarrollo de los procesos de pensamiento lógico-matemático. En ese momento, Papert (1980) acuñó el término pensamiento computacional ya que el objetivo de la inclusión de la computadora en las escuelas era mejorar y potenciar los procesos de pensamiento, la creatividad tecnológica y la expresividad que la computación permite.

La matemática y las CC se han vinculado desde sus orígenes tomando, en reiteradas ocasiones, referencias y bases situadas en la trayectoria investigativa del campo de la matemática, lo que permite incorporar y articular la computación, la educación y las matemáticas como una transformación propia de la práctica educativa.

Como abordaje pedagógico, la modelización nos permite pensar en estrategias para llevar a cabo propuestas áulicas para el desarrollo de un pensamiento computacional y matemático y, al

mismo tiempo, abordar problemáticas del mundo real. Enfoques que refieren a los núcleos de las Ciencias de la Computación (Denning et al., 1989) otorgan importancia a la modelización de relaciones entre el flujo de la información, la experimentación a través de programas sencillos y la resolución de problemas dentro de las restricciones de una máquina, en donde aprender a programar requiere del empleo de un conjunto de conocimientos técnicos asociados a la manipulación simbólica por medio de una computadora.

De esta manera, consideramos que procesos similares a la modelización matemática se pueden observar en procesos vinculados a conceptos de las Ciencias de la Computación en asignaturas escolares del área de la informática. Este proyecto busca producir material teórico educativo que aporte a las áreas de vacancia a partir del diseño y análisis de propuestas de modelización mediadas por tecnologías digitales de forma colaborativa entre un grupo de docentes e investigadores para las áreas de Matemática, Educación Tecnológica y Computación.

En el desarrollo de este escrito, presentamos a continuación las referencias teóricas que enmarcan nuestro trabajo de investigación. Luego exponemos el enfoque metodológico utilizado y posteriormente nos adentramos en los dos contextos escolares en los que actualmente estamos implementando proyectos de modelización en Computación. Finalmente, vinculamos la base teórica que nos da sustento con las experiencias que se están desarrollando y esbozamos unas tensiones que van apareciendo en la práctica junto a consideraciones finales.

### **Referencias teóricas del proyecto**

El proceso de modelización, esencial en la producción del conocimiento científico, se muestra sinérgico con el empleo de tecnologías digitales (TD) en las distintas fases que conforman el proceso (Villarreal et al., 2018; Villarreal y Mina, 2020). La noción de Modelización Matemática (MM), se considera en educación como una práctica de enseñanza- aprendizaje que establece una relación entre el mundo real y la matemática, la cual puede motivar el proceso de aprendizaje y ayudar al aprendiz a establecer raíces cognitivas sobre las cuáles construir importantes conceptos matemáticos (Blomhøj, 2021). Para este trabajo tomamos los postulados de Bassanezi (2012) en relación a la MM, el cual organiza el trabajo en el aula en los siguientes subprocesos: 1) elección de tema y formulación de problema(s), 2) experimentación, 3) abstracción, 4) resolución, 5) validación, 6) modificación y 7) aplicación. En Echeveste (2020) se pudieron observar indicadores de la presencia de las fases de Bassanezi (2012) en proyectos de software desarrollados por estudiantes de escuelas secundarias. Además, el proceso de modelización contempla la actividad de resolución de problemas, lo cual muestra gran potencialidad con los aportes que Brennan y Resnick (2012) realizan al pensamiento computacional, que incluye tres dimensiones claves: conceptos computacionales; prácticas computacionales y perspectivas computacionales.

Si bien reconocemos los debates y tensiones en torno al término pensamiento computacional popularizado por Jeannette Wing en 2006, consideramos que uno de los acuerdo al respecto establece una clara relación con la resolución de problemas, ya que constituye un enfoque que incluye (al menos) el análisis de datos, la organización lógica de los mismos, la búsqueda de soluciones a modo de secuencia de pasos ordenados, la valoración, identificación e implementación de posibles resoluciones con el fin de obtener la solución más eficiente en relación a los pasos del algoritmo, el uso de recursos en el proceso de resolución, etc. (Grover y Pea, 2013; López-Lñesta, et al, 2019)

En contextos educativos, la modelización ofrece la posibilidad de crear escenarios que permiten vincular problemáticas y conocimientos específicos con el mundo real. Esto se entrama con el registro de que las actividades de materias que despiertan un mayor interés estudiantil en el área de la computación se vinculan a propuestas de enseñanza ligadas a materializar el conocimiento aprendido y encontrarle utilidad y sentido a aquello que están aprendiendo (Echeveste, 2022). Sin embargo, tal como lo mencionan Schapachnik y Bonello (2020) no cualquier tipo de problema es un problema computacional ni cualquier enseñanza de la algoritmia y la programación favorece el tipo de desarrollo cognitivo que buscamos en el proyecto.

En Córdoba, si bien la modelización como actividad matemática y la integración de tecnologías digitales como herramientas mediadoras en el aprendizaje del estudiantado, se mencionan en los Diseños Curriculares para la Educación Secundaria, en la escuela es todavía escaso y poco significativo el trabajo con modelización y tecnologías. Por lo tanto, recopilar información de contextos educativos reales en donde se ponen en juego prácticas educativas de modelización tendrá impacto en los desarrollos en el área.

### **Marco metodológico**

Esta investigación tiene un enfoque cualitativo, enmarcando esta decisión metodológica en lo que se denomina investigación intensiva; ya que los recortes empíricos serán acotados a estudios en profundidad. Esto “posibilita entramar procesos socioestructurales con los procesos y relaciones vividos y significados por los sujetos” (Achilli; 2005, p 64). Se realiza una investigación-acción participativa, ya que desde este enfoque se configura, analiza y resuelve el problema de investigación con actores que están involucrados en él y que se agrupan en la acción de investigar aquellos hechos que los afecte, buscando su transformación (Gellert et al., 2013). En el caso del proyecto general, serán las y los docentes de las cinco escuelas de nivel secundario seleccionadas quienes participen del diseño y la implementación de la propuesta, por lo tanto serán parte y objeto de estudio al recuperar su posterior experiencia luego de desarrollar las actividades en sus aulas. Cuatro docentes de matemática -correspondientes a

dos escuelas-, un docente de educación tecnológica -correspondiente a una escuela- y cuatro docentes de computación -correspondientes a dos instituciones. Para esta ocasión desarrollaremos el trabajo realizado hasta el momento en estas últimas dos escuelas con orientación en informática.

El modo de recolección de datos se realizó a través de reuniones meet y visitas a la institución presencial en las que fuimos revisando la propuesta de aula en conjunto. En esta etapa de seguimiento nos manejamos mediante teléfonos, audios y otro tipo de material audiovisual registrado por los docentes del trabajo en sus clases. Este último registro se hace a posteriori y es reflexivo por parte del docente en relación a su propia práctica.

De esta manera, la metodología utilizada, reflexiva e investigativa, interrelaciona teoría y práctica y toma “como punto de partida y de llegada la práctica profesional docente y, como mediación, las teorías educativas y la investigación sobre la práctica de cada uno” (Lucchesi & Fiorentini, 2013, pp 14).

A continuación, presentamos los casos analizados a los que llamamos *escuela A* y *escuela B* para guardar anonimidad, de la misma manera que serán ficcionados los nombres de los y las docentes mencionadas.

### **Presentación de los casos**

A inicio del segundo semestre de 2023 hemos establecido vínculo con las cinco escuelas. Tres de ellas ya acordadas previamente en 2022, mientras que en las dos escuelas con orientación en informática se realizó durante el corriente año, a través de la Secretaría de Promoción de Ciencia y Nuevas Tecnologías del Ministerio de Educación de la Provincia de Córdoba.

Como inicio del proyecto, tomamos contacto con las diferentes instituciones educativas y, una vez conocida la propuesta y aceptada su participación, tuvimos una reunión general entre integrantes del equipo de investigación, autoridades de las escuelas y sus docentes. En esta instancia compartimos algunas líneas teóricas de base en torno a la modelización, el pensamiento computacional y las tecnologías digitales y comenzamos a delinear los proyectos que se llevarían al aula.

Como primeros datos generales de esa reunión pudimos identificar que la idea de modelización que subyace está ligada principalmente a un modelo como representación o partiendo de modelos ya realizados y no como propuesta de enseñanza.

#### *Escuela A*

La Escuela A es una institución de gestión pública ubicada en un barrio de la Ciudad de

Córdoba en cercanías al centro y el casco histórico. Cuenta con orientación en informática y se encuentra participando en actividades como ferias y jornadas de divulgación, lo cual le permite mantenerse al día con las últimas tendencias y avances tecnológicos.

El equipo de trabajo que hemos conformado cuenta con la participación de tres docentes de diferentes años y materias, más el acompañamiento del vicedirector, quien ejercía anteriormente como docente de la institución en materias del área y cuenta con una formación en analista en sistemas y tecnología educativa.

La propuesta de proyecto: Esta institución tenía como objetivo establecer una 'escuela inclusiva'. Y aunque habían realizado un pequeño trabajo previo, llevaban bastante tiempo con la idea de realizar un proyecto que abarque a toda la escuela, sin embargo, aún no la habían implementado. Cuando les propusimos colaborar en este proyecto, les brindamos el impulso necesario para concretar esa propuesta. Esto derivó en que se coordinaran horarios de encuentro para pensar actividades transversales y colaborativas que permitan confluir diferentes asignaturas en un solo proyecto. Además en nuestro equipo de investigación contamos con la participación de un colega no vidente lo cual enriquece significativamente el intercambio que se produce en la construcción del proyecto.

El proyecto se desarrollará en los cursos de 4to y 5to año, cada uno de los cuales cuenta con alrededor de 30 estudiantes. En la siguiente tabla se presentan datos generales de los docentes del ciclo orientado, y, a continuación, cómo se está iniciando el proyecto.

<b>Docente</b>	<b>Año</b>	<b>Asignatura</b>	<b>Formación</b>
Profesor C	4to	Sistemas digitales de información	Analista en sistemas de información
Profesor P	5to	Sistemas digitales de información	Analista universitario de sistemas
Profesor J	5to	Alfabetización Digital	Analista de sistemas de computación. Profesor de Informática para escuela Secundaria

El profesor C empezó con los proyectos en septiembre, dividiendo a sus estudiantes en 6 grupos y proporcionando diferentes temáticas. Reconocemos tres subgrupos de acuerdo a su temática. El primero, designado a la creación y construcción de tótems, a partir de computadoras en desuso, con información semanal coordinada con dirección. El segundo subgrupo se divide en dos equipos: uno dedicado a la creación de juegos utilizando materiales reciclables, y, el otro, enfocado en la construcción de una máquina soldadora que facilitará la producción de dichos juegos. Finalmente, el tercer subgrupo se dedicará a la construcción de un

lápiz 3D con el propósito de desarrollar teclados adaptados para personas con hipoacusia y discapacidad visual. Paralelamente, otro equipo se encargará de la producción de los filamentos necesarios para el lápiz 3D a partir de materiales reciclados introduciendo de esta manera conocimientos vinculados a software y hardware.

El profesor J abordará nociones vinculadas a la alfabetización digital y junto a su grupo de estudiantes acondicionará la escuela con la conectividad necesaria para que las computadoras se adecuen a los equipos con software lectores de pantallas. Se les propuso a los y las estudiantes que realicen una indagación sobre la adaptabilidad de los equipos informáticos, cómo haría una persona con dificultades visuales para usar una computadora, y a partir de ahí surgieron distintos elementos de software y hardware, como así también la posibilidad de generar tiras adhesivas que tengan relieve para los teclados. También se exploraron aplicaciones de celulares y se les propuso que observen su funcionamiento para posteriormente acondicionar las netbook de las escuela con lo que vayan desarrollando y lograr una computadoras adaptada

El profesor P junto con su curso de 5to año, trabajarán a partir del trabajo que realizará el Profesor C, ya que eso será de insumo para la configuración de redes. Este docente comentó que suele introducir su planificación con el diseño y planificación de una pyme que en este caso se vinculó al tema de trabajo colectivo.

Durante una reunión con los docentes y los integrantes de este proyecto, se dieron diversas sugerencias. Una de ellas se refería a los teclados adaptados. La idea inicial de los estudiantes era agregar correspondencias en braille a las teclas. Sin embargo, se reconoció que no todas las personas con discapacidad visual dominan el braille, ya que es un sistema complejo que implica aprender un nuevo idioma. Por esta razón, se planteó la idea de llevar a cabo una investigación sobre lectores de pantalla. Esto permitiría identificar aquellos que funcionan de manera eficiente, así como aquellos que presentan limitaciones. Esta herramienta facilitaría el uso de las netbooks de la escuela para cualquier persona con discapacidad visual, independientemente de su conocimiento en braille.

### *Escuela B*

Esta escuela se encuentra ubicada en una zona rural que funciona como Anexo, el cual depende de un IPEM (Instituto Provincial de Enseñanza Media) con sede en la Ciudad de Córdoba. No hay colectivos directos para acceder a la institución ya que la misma se encuentra rodeada de campos sembrados. Si bien tienen pedido un colectivo, a la escuela asisten principalmente en bicicleta o caminando y, quienes son mayores, lo hacen en motocicleta.

Su matrícula completa es de 60 estudiantes, correspondientes a toda la secundaria, que se

encuentra dividida en pluricursos con dos divisiones para 1ero, 2do y 3er año, y una para el último tramo: 4to, 5to y 6to, que poseen orientación en informática. La mayoría de los y las estudiantes residen en asentamientos cercanos y muchos trabajan como jornaleros en los campos con sus familias, como peones en las cosechas, en la cortadera de ladrillos o trabajos vinculados con el carbón. Reportan tener grandes dificultades con la conexión a internet y utilizan chip de prepagas.

La propuesta de proyecto: Cuando tomamos contacto con la coordinadora de la institución, nos comentó que tenía un proyecto que no se había podido concretar el año anterior pero sí tenían identificado el problema al que querían dar solución. Este año lo habían presentado a la Feria de Ciencias y desde ahí participaríamos colaborativamente con nuestro equipo de investigación. El mismo consistía en la reutilización de agua de lluvia y diseño de riego automático y respondía a dos grandes problemáticas: la primera, vinculada a las inundaciones reiteradas que tiene la escuela en las épocas de lluvia o de riego de los campos, y, la segunda, ligada a la intención de tener una huerta considerando que, en épocas de receso escolar, no contaban con quien pudiera regar y ahuyentar a los animales de la zona. Es allí donde la docente M, recientemente incorporada a la escuela, con un gran porcentaje de las horas en los cursos del ciclo orientado, retoma este proyecto con la intención de utilizar Arduino Uno y sensor de humedad de suelo.

A continuación, presentamos un cuadro con datos generales del proyecto y, luego, tal como realizamos en la descripción de la escuela A, nos explayaremos en presentar lo desarrollado hasta el momento.

Docente	Año	Asignatura	Formación
Profesora M	4to	Alfabetización Digital	Técnica en desarrollo de software
	5to	Desarrollo de soluciones informáticas	
	6to	Desarrollo de soluciones informáticas	

La profesora M nos presentó una primera planificación en donde se les daría a quienes cursan 4° año la identificación de dispositivos (entrada y salida) y armado de power point; a quienes están en 5° año la investigación de Arduino y diagrama del código; y a 6° la escritura de lenguaje código. Si bien menciona que esta diferenciación es a los fines de responder a la modalidad pluricurso, en general suelen trabajar bastante en conjunto y la separación no es tajante.

### Vinculación con la Modelización

Ambos proyectos se encuentran en la primera etapa del proceso de Modelización: elección del tema y formulación de problema(s), que luego deriva en una investigación para recoger datos, lo cual da lugar a la formalización del problema y a la selección de las variables esenciales que se involucran en un proceso de abstracción. Acá entra a jugar lo que Schapachnick y Bonello (2020) llaman problema computacional. En esta instancia se sistematiza y conceptualiza una posible solución computacional ante el problema a resolver. Tal como mencionamos, a medida que se avanza con la delimitación del problema, se van obteniendo datos experimentales que ayudan a la formulación de definiciones, por lo tanto, también se empieza a avanzar en los subprocesos 2 y 3, correspondientes a la experimentación y la abstracción del problema, según el esquema de Basanezzi (2012).

En la *escuela A* se retoma un tema ya pensado en momentos anteriores, en especial por una participación previa en donde un grupo de estudiantes coordinados por el vicedirector colaboraron con un Instituto para personas con déficit auditivo. A pesar de que en la escuela no haya estudiantes con discapacidad visual, se plantea ¿por qué esperar para adaptar la institución? A partir de la *formulación del problema*, se da lugar a diversas propuestas estudiantiles (comienza la *abstracción* del problema y su *experimentación*) para adecuar tanto los espacios como las herramientas, incluyendo la creación de carteles en braille, la instalación de tótems informativos y la adecuación de las computadoras, entre otras iniciativas.

En el caso de la *escuela B* se parte de una situación problemática local y propia tanto de la escuela como de la población de alrededores, ya que la huerta tiene el fin de ser utilizada por la comunidad y las familias del grupo de estudiantes.

Como parte de la consigna de trabajo, la docente les pidió que recolecten información sobre Arduino (y sus pines), input-output, sensores y una búsqueda de sistemas de riego. En palabras de la docente: *“Tienen que ver como implementarlo en tarros de 200 litros ubicados en distintos sectores que se pueden comunicar para que en un solo lugar se ponga la bomba de agua, que se vaya llenando a medida que se va vaciando.”*

En la clase se desarrolló una lluvia de ideas en donde iban comentando según lo explorado y delimitando la mejor solución al problema que se planteaban. Por ejemplo para la huerta se trabaja con la hipótesis de que realizar huerta vertical era la mejor opción atendiendo al riesgo de los animales de la zona, por el tipo de tierra y las inundaciones. En cuanto al sistema de riego se barajaron opciones de tipo manual, goteo y aspensor, como así también donde ubicarla por cuestiones de sol y sombra. Dentro de las dudas que iban teniendo surgió la posibilidad de consultar a un arquitecto para que los asesore en las cuestiones estructurales.

Otra parte del proyecto consistía en mirar una parte de un código y analizar qué decía. El mismo correspondía a los valores de los sensores de humedad, para luego implementarlo en el ID del Arduino. Los contenidos a trabajar que aparecen en la planificación corresponden a: Hardware (periféricos de entrada y salida) automatización de productos tecnológicos, placas controladoras, sensores, actuadores; y a Algoritmos; Estructuras de control y Programación en Arduino.

Como parte del trabajo en conjunto con el equipo de investigación se le recomendó para los contenidos de 4to año desplazar el uso de herramientas de ofimática y se le sugirió como tecnología digital la utilización de Trello, un software de administración de proyectos, para que puedan llevar un mayor registro de las actividades. A su vez, el objetivo en ambos proyectos (escuela A y escuela B) es sistematizar todo lo que se realizará para darle un formalización en términos de propuesta didáctica.

Para finalizar mencionaremos algunos desafíos que comenzaron a aparecer en el desarrollo del proyecto junto con datos que nos permiten seguir pensando cómo continuar este proyecto de investigación.

### **Algunos desafíos de la práctica y consideraciones finales**

El primer gran desafío como equipo de investigación fue corrernos del lugar docentes universitarios/as que les llevaría a los equipos docentes de secundaria actividades ya diseñadas para que luego sean implementadas y, aclarar en cambio, la propuesta de pensar en conjunto una propuesta didáctica de fondo vinculada a la modelización. Otro desafío que se enuncia tanto en los docentes de la escuela A como en la docente de la escuela B es la integración de distintos contenidos correspondientes a asignaturas y años diferentes. Ya sea en el caso A, por tener diversas asignaturas, o en el caso B por ser un pluricurso en el que se les tiene que asignar actividades diferentes según el año que estén cursando.

En muchos proyectos en los que se trabaja con desarrollo de software en las escuelas se utilizan propuestas de aprendizaje basado en problemas o desde la perspectiva de metodologías ágiles. Esas propuestas tienen puntos en común con los pasos de la modelización desde la trayectoria pedagógica estudiada por la matemática. Quedará como tema para seguir trabajando cómo se da la articulación con aquellas propuestas o incluso en qué puntos discrepan.

Trabajar desde la modelización nos permite pensar en estrategias para llevar a cabo propuestas áulicas que desarrollen un pensamiento computacional y matemático y, al mismo tiempo, aborden problemáticas del mundo real.

Como parte de lo que se desarrollará en lo que resta del proyecto, se ha observado que los docentes de las escuelas con orientación en informática comparten su formación inicial de Analista en Sistema con el docente de la materia de educación tecnológica de la tercera institución que forma parte del proyecto. Asimismo el docente de matemática del cuarto colegio que participa en la investigación es licenciado en tecnología educativa. Esto muestra que, en la actualidad, se está dando un cruce de disciplinas en las aulas, cuestión que nos interesa aprovechar para articular futuras propuestas de clases.

## Bibliografía

- Achilli, E. (2005) *Investigar en antropología social: los desafíos de transmitir un oficio*. Ed.Laborde libros. Rosario
- Bassanezi, R. C. (2012). *Temas y modelos* (1.ed.). Campinas, Brasil: UFABC.
- Blomhøj, M. (2021). Modelización Matemática - Una Teoría para la Práctica. *Revista De Educación Matemática*, 23(2). Recuperado a partir de <https://revistas.unc.edu.ar/index.php/REM/article/view/10419>.
- Bonello, M. B., & Schapochnik, F. (2020). Diez preguntas frecuentes (y urgentes) sobre pensamiento computacional. *Virtualidad, Educación y Ciencia*, 11(20), 156-167.
- Brennan, K., y Resnick, M. (2012) New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association*, Vancouver, Canada.
- Denning, P. J., Comer, D. E., Gries, D., Mulder, M. C., Tucker, A., Turner, A. J., y Young, P. R. (1989). Computing as a discipline. *Computer*, 22 (2), 63-70. Final Report of the ACM Task Force on the Core of Computer Science, in cooperation with the IEEE Computer Society.
- Echeveste, ME (2020) “Estudiar Programación en la Escuela Secundaria Técnica. Análisis de la relación con el conocimiento de jóvenes que cursan la orientación en programación”. Tesis de Doctorado en Ciencias de la Educación. Facultad de Filosofía y Humanidades. Universidad Nacional de Córdoba.
- Echeveste, ME (2022) “La búsqueda por el sentido a aprender a programar y su relación con la forma en la que se enseña en las escuelas.”. Simposio Argentino de Educación en Informática (SAEI) en las Jornadas Argentinas de Informática (JAIIO 2022)
- Gellert, U., Becerra Hernández, R. & Chapman, O. (2013) Research methods in mathematics teacher education. En M. A. Clements et al. (Eds.) *Third International Handbook of Mathematics Education* (p. 327-360). New York: Springer.
- Grover, S., y Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, 42(1), 38-43.
- López-Iñesta, E., Ros-Esteve, M., y Diago, P. D. (2019). Desarrollo de destrezas de

pensamiento computacional con actividades desenchufadas para la resolución de problemas matemáticos . V Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2019).

- Lucchesi, D. y Fiorentini, D. (2013) Refletir e investigar a própria prática de ensinar/aprender matemática na escola. En D. Lucchesi et al. (Orgs.) *Análises narrativas de aulas de matemática* (p. 11-23). Campinas: Pedro & João Editores.
- Lugo, M. T., Ithurburu, V. S., Sonsino, A., y Loiacono, F. (2020). Políticas digitales en educación en tiempos de Pandemia: desigualdades y oportunidades para América Latina. Edutec. *Revista Electrónica De Tecnología Educativa*, (73), 23-36.
- OECD (2019) PISA 2018 Assessment and Analytical Framework. Paris: OECD Publishing.
- Papert, S. (1980). *Mindstorms—Children, Computers and Powerful Ideas*. New York: Basic Books, Inc.
- Villarreal, M. E., & Mina, M. (2020). Actividades Experimentales con Tecnologías en Escenarios de Modelización Matemática. *Bolema: Boletim de Educação Matemática*, 34, 786-824.
- Villarreal, M.; Esteley, Cristina & Smith, S. (2018). Pre-service teachers' experiences within modelling scenarios enriched by digital technologies. *ZDM Journal of Mathematics Education*, 50(1-2): 327-341.

## **Acercar la Inteligencia Artificial a la escuela. Una experiencia de extensión de la UNLP**

Isabel Kimura, Claudia Queiruga, Claudia Banchoff Tzancoff, Sofía Martín  
ikimura@linti.unlp.edu.ar, {claudiaq,cbanchoff}@info.unlp.edu.ar, smartin@linti.unlp.edu.ar

Laboratorio de Investigación en Nuevas Tecnologías Informáticas  
Facultad de Informática  
Universidad Nacional de La Plata

### **Abstract.**

Este trabajo presenta una experiencia sobre el abordaje de contenidos de Inteligencia Artificial (IA) en el marco del proyecto de extensión de la UNLP denominado “Extensión en vínculo con escuelas secundarias”. Es en este sentido, que la IA, a partir de la reciente irrupción de herramientas de IA generativa al alcance de un clic, como el chatbot ChatGPT para textos o Dall-e para imágenes, entre tantas otras, traen consigo la novedad de producir cosas que no existían, transformando sustancialmente la manera en que se produce y consume contenido, entre otras actividades. Desde el proyecto consideramos relevante incluir contenidos que permitan comprender cómo funcionan estas nuevas tecnologías que indudablemente afectan el mundo del trabajo, del estudio y de la mayoría de las actividades productivas. En este trabajo se describen dos experiencias implementadas con estudiantes del segundo ciclo de la educación secundaria, en el contexto del proyecto de extensión antes mencionado, en el que se abordaron contenidos de Aprendizaje Automático utilizando las extensiones disponibles en MIT AppInventor para voces sintéticas y reconocimiento de imágenes. Se trabajó en modalidad taller, recuperando los conocimientos previos en programación de aplicaciones móviles con MIT AppInventor e indagando sobre elementos de IA presentes (o no) en las aplicaciones que se fueron construyendo durante el mismo. A la vez se presentaron casos de usos de IA para voces sintéticas de personas públicas y se debatió sobre las implicancias éticas. A partir de observaciones y de preguntas realizadas, antes, durante y después de los talleres, se observa, por un lado la existencia de grupos de estudiantes que autónomamente incursionaron en temas de IA previo al taller y otros que solo conocían el término pero no habían experimentado su uso y no lo relacionaban con aplicaciones de uso cotidiano. De las voces de los estudiantes surgió la reflexión que los contenidos que se publican en Internet podrían alimentar las IA y en este sentido la relevancia de ser conscientes de esta realidad.

**Keywords:** Inteligencia Artificial, Aprendizaje Automático, MIT AppInventor, Escuela Secundaria

### **Introducción**

La Inteligencia Artificial (IA) es un campo de conocimiento que nace en los años cincuenta, sin embargo en la última década se ha materializado en artefactos de uso cotidiano, ejemplo de ello son los asistentes virtuales como Siri, Alexa y Google Now, en videojuegos, en los sistemas de recomendaciones de Amazon, Mercado Libre, Netflix, entre otras plataformas, resultando en herramientas útiles para los consumidores a la vez que plantean cuestionamientos sobre la privacidad y la posible creación de burbujas informativas. Más recientemente, la irrupción de herramientas de IA generativa, como el chatbot ChatGPT para textos o Dall-e para imágenes, ambas herramientas desarrolladas por la empresa OpenAI, comenzaron a influir fuertemente en sus usos en educación. Es por ello que resulta relevante, desde la educación obligatoria, conocer y comprender cómo funcionan las tecnologías que operan detrás de estos productos y servicios tecnológicos y a la vez promover una perspectiva crítica de sus usos que contemple la intervención y la construcción con IA.

Las siguientes iniciativas educativas que exploran las posibilidades de la introducción de la IA para el aprendizaje en el aula de la escuela y que resultan de interés para este trabajo fueron seleccionadas como antecedentes relevantes: Developing Middle School Students' AI Literacy (Lee I., et al., 2021), Enfoque didáctico para la enseñanza transdisciplinar de la Inteligencia Artificial (Bertochi S., et al., 2021), Co-Designing Machine Learning Apps in K–12 With Primary School Children (Toivonen T., et al, 2020). El criterio de selección de las mismas se basó en la flexibilidad que ofrecen al momento de introducir estos contenidos, dado que no están sujetas a asignaturas ni conocimientos previos en programación. Estas experiencias se analizaron teniendo en cuenta:

- El nivel educativo para el que están orientadas.
- Si los algoritmos de IA que se intentan enseñar hacen foco en la relevancia de los datos disponibles que alimentan los modelos de IA diseñados.
- Si en las experiencias se discuten o no las implicancias éticas del uso de los datos y de las predicciones obtenidas.
- Los materiales que se usan para el desarrollo de las experiencias.

### **Experiencias de enseñanza de IA en el proyecto “Extensión en vínculo con escuelas secundarias”**

“Extensión en vínculo con escuelas secundarias” es un proyecto que forma parte del programa de extensión “Educación para la inclusión” de la UNLP y que se focaliza en un trabajo de articulación con escuelas secundarias, planteando actividades con docentes y estudiantes,

dando a conocer la disciplina Informática como campo de conocimiento que contribuye a la formación de jóvenes en un área escasamente desarrollada en la escuela secundaria y fuertemente vinculada al campo laboral y al desarrollo socio-productivo. Se trabaja con docentes y estudiantes de escuelas secundarias de gestión pública de la provincia de Buenos Aires de la región de La Plata, Berisso, Ensenada y Magdalena y las escuelas preuniversitarias de la UNLP, en los siguientes ejes:

- Generación de espacios de diálogo y comunidades de aprendizaje entre la Facultad de Informática y escuelas secundarias de la región, que retomen la discusión sobre el rol de la informática en las propuestas de formación, como campo de estudio.
- Conceptualización de la educación como derecho social y revalorización de la educación pública.
- Promoción de las carreras de Informática entre estudiantes de escuelas secundarias

En el contexto de este proyecto de extensión se realizaron 2 experiencias de acercamiento a temas de IA con estudiantes y docentes de escuelas secundarias del ciclo superior, en modalidad taller: “Un primer acercamiento al Aprendizaje Automático y sus usos”, sistematizadas en la tabla 1.

<b>Edición JADiCC 2022 (Agosto 2022)</b>			
<b>Destinatarios</b>	<b>Objetivos de aprendizaje</b>	<b>Contenidos</b>	<b>Materiales y estrategias didácticas</b>
Estudiantes del nivel secundario del segundo ciclo, de un bachillerato tecnológico y otro con orientación en arte, de la provincia de Corrientes.	Identificar los usos de ML y DL en aplicaciones de uso cotidiano y sus implicancias éticas.  Interactuar con aplicaciones móviles experimentales que hacen uso de técnicas de ML y DL.	<b>Conceptos de IA</b>	
		Aprendizaje automático (ML)	Extensión de APPInventor LookExtention (para identificar imágenes)
		Aprendizaje profundo (DL)	Proyecto FakeVoices de APPInventor
		<b>Cuestiones éticas de la IA</b>	

	Programar una aplicación móvil que usa módulos basados en técnicas ML y DL.	Deepfakes	Analizar videos y determinar si fueron creados o no por IA
<b>Edición Facultad de Informática (Noviembre 2022)</b>			
<b>Destinatarios</b>	Identificar los usos de ML y DL en aplicaciones de uso cotidiano y sus implicancias éticas.	<b>Conceptos de IA</b>	
Estudiantes del nivel secundario técnico de la Especialidad “Programación” de la ciudad de “La Plata”	Interactuar con aplicaciones móviles experimentales que hacen uso de técnicas de ML y DL.	Aprendizaje automático (ML)	Extensión de APPInventor LookExtention (para identificar imágenes)
	Crear y entrenar un modelo a partir de datos propios sobre un dominio específico.	Aprendizaje profundo (DL)	Proyecto FakeVoices de APPInventor
	Programar una aplicación móvil que usa módulos basados en técnicas ML y DL.	Entrenamiento de modelos	Google Teachable Machine
		<b>Cuestiones éticas de la IA</b>	
		Deepfakes	Analizar videos y determinar si fueron creados o no por IA

Tabla 1. Un primer acercamiento al Aprendizaje Automático y sus usos: experiencias en la enseñanza de IA en el proyecto “Extensión en vínculo con escuelas secundarias”

A continuación se describen las dos experiencias desarrolladas.

### **Primera experiencia: JADICC 2022**

La primera experiencia transitada por el equipo de este proyecto, en torno a la enseñanza de IA, se realizó en las JADICC 2022 (Jornadas Argentinas de Didáctica de la Ciencia de la

Computación) en la UNNE, en el marco de la convocatoria de presentación de talleres (Kimura I., et al, 2022, pp 12). El taller fue diseñado para trabajar con estudiantes del nivel secundario con conocimientos básicos de programación y algorítmica y se enfocó en, por un lado, la introducción de las diferentes técnicas de IA que se utilizan en las aplicaciones de uso cotidiano, específicamente en aplicaciones móviles, y por otro lado en la dimensión ética de la IA. En este sentido el enfoque usado para el desarrollo del taller incluyó saberes propios de la disciplina Informática, en tanto comprender cómo se construye/crean artefactos tecnológicos (en este caso apps) que usan IA, es decir que “hay por detrás” de una aplicación que reconoce imágenes o reproduce voces y poder experimentar un proceso de construcción de una app que incluye IA. La problematización de estas tecnologías en cuanto a sus usos y dinámicas de inclusión/exclusión, también formaron parte de la discusión de la IA en este taller.

A continuación se describen los destinatarios, los objetivos de aprendizaje, la selección de contenidos de IA trabajados, las actividades realizadas y las herramientas utilizadas, para el desarrollo del taller.

- **Destinatarios:** participaron del taller 20 estudiantes de 6to año, 13 del Colegio Informático "San Juan de Vera" y 7 estudiantes del Instituto Superior de Música "Prof. Carmelo H. De Biasi", de la ciudad de Corrientes. El primer grupo corresponde a estudiantes que se encuentran transitando un bachillerato tecnológico, con conocimientos de programación y de la herramienta MIT App Inventor. Por otro lado, el segundo grupo de estudiantes se encuentran transitando un bachillerato con orientación en música y sin conocimientos previos en programación. Previendo la disparidad de formaciones previas en relación a contenidos de programación, se envió a los docentes a cargo de los grupos de estudiantes, un material para introducirlos en temas de programación y en el uso de MIT App Inventor. De esta manera, el tiempo para el desarrollo del taller, 2 horas, se enfocó en los contenidos específicos de IA.

- **Objetivos de aprendizaje:**

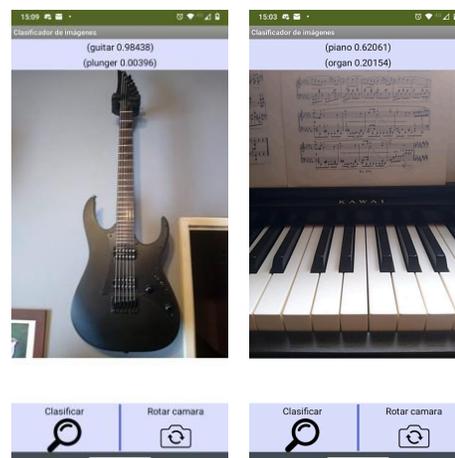
- Interactuar con aplicaciones móviles experimentales que hacen uso de técnicas de aprendizaje automático y aprendizaje profundo.
- Programar una aplicación móvil simple que hace uso de módulos basados en técnicas de aprendizaje automático y aprendizaje profundo.

- **Contenidos de IA:** se trabajaron conceptos de aprendizaje automático y aprendizaje profundo para reconocer imágenes y crear sonidos. La selección de estos contenidos se basó, por un lado, en que las nociones del funcionamiento del aprendizaje automático y profundo se pueden explicar de una manera sencilla a través de ejemplos concretos y conocidos y, por otro, en la existencia de materiales didácticos pensados para personas sin formación en Informática

(o escasa) y con conocimientos básicos de matemática. En relación a esto último, el proyecto “Artificial Intelligence with MIT App Inventor”<sup>1</sup> impulsa el desarrollo de materiales didácticos con especial atención en la experimentación de tecnologías de IA en clave de creación digital y no en el mero uso.

- **Actividades realizadas:**

**Reconocimiento de imágenes:** concretamente se propuso a los estudiantes que usen la app “Clasificador de imágenes” disponible en MIT AppInventor para reconocer imágenes capturadas con las cámaras de sus celulares y observen el grado de confianza con que la app reconoce dichas imágenes. A modo de ejemplo, en la Figura 1 se muestra el grado de confianza que arroja la app al reconocer diferentes imágenes. De esta manera, se introdujeron dos conceptos claves en el aprendizaje automático: el grado de confianza con que se reconocen las imágenes y el entrenamiento de modelos de clasificación de aprendizaje automático. En relación al grado de confianza, se indagó acerca de los valores arrojados, por ejemplo en el caso de la guitarra (Figura 1) el grado de confianza es alto dado que es cercano a 1, sin embargo en la otra imagen, de la misma figura, el grado de confianza arrojado es menor. En estos casos se hicieron preguntas sobre las implicancias que esto podría tener al utilizar una app que incluya una tecnología de reconocimiento de imágenes como la descrita, para la toma de decisiones. Por ejemplo, la identificación automática entre un arma de juguete y un arma de verdad, o en



imágenes médicas la existencia o no de un tumor. Las conclusiones a las que se arribaron fueron que el modelo entrenado para la aplicación desarrollado le faltan datos de entrenamiento y por lo tanto los resultados no son determinantes.

Figura 1. Uso de la app “Clasificador de imágenes” disponible en App Inventor

<sup>1</sup> El proyecto “Artificial Intelligence with MIT App Inventor” del MIT (Massachusetts Institute of Technology) ofrece un conjunto de materiales y secuencias didácticas para enseñar IA, orientado a espacios escolares y de capacitación, a través del uso MIT App Inventor. <https://appinventor.mit.edu/explore/ai-with-mit-app-inventor>

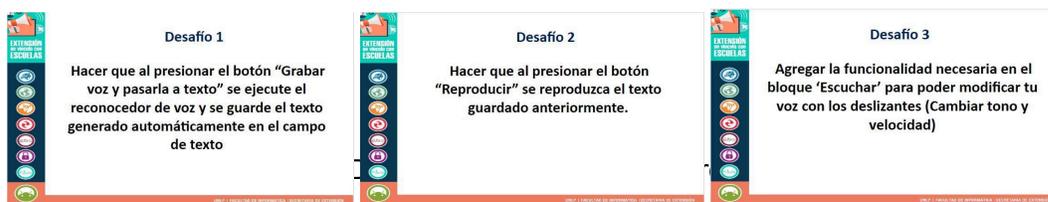
Previo al uso de la app “Clasificador de imágenes”, se propuso a los estudiantes que completen la aplicación, para ello debían usar la extensión de MIT App Inventor LookExtention<sup>2</sup>. A diferencia de las propuestas que se enfocan en el desarrollo de un proyecto de software desde cero, en este taller se decidió tomar un proyecto existente y agregarle funcionalidad de IA, mediante el uso del modelo de clasificación ofrecido por LookExtention. La limitación del tiempo disponible para el desarrollo del taller, fue el motivo que guió la decisión de trabajar con un proyecto existente.

**Creación de voces sintéticas:** en esta actividad se propuso a los estudiantes que completen y usen la app FakeVoices<sup>3</sup> con la intención de identificar la existencia (o no) de IA en la creación de voces. En esta actividad se trabajó el concepto de DeepFake (o ultra falso) y los conflictos éticos que trae aparejados, al problematizar la creación de discursos mediante la manipulación de voces de personas reales con técnicas de IA.

FakeVoices ofrece dos funcionalidades en las que interviene la IA:

- a) conversión de texto a voz, mediante un conjunto predeterminado de voces sintéticas en un idioma determinado por la configuración del dispositivo.
- b) conversión de voz a texto que reconoce “on the fly” (al vuelo) el idioma del hablante y lo transcribe.

Se propuso a los estudiantes que resuelvan una secuencia de desafíos, a partir de un proyecto FakeVoices compartido por los docentes, como se muestra en la figura 2.



En esta actividad los estudiantes observaron que las voces grabadas se reproducen con la voz predeterminada por el dispositivo (motor Google TTS) y no con la voz original. Además pueden alterarla modificando la configuración mediante el cambio de tono y velocidad. Durante el desarrollo de la actividad se indaga sobre la autenticidad de las voces creadas, y se arriba a la conclusión que como las voces “suenan naturales”, como si hablase una persona, es muy difícil

<sup>2</sup> LookExtention es una extensión de MIT APP Inventor que se usa para reconocimiento de imágenes e incluye un modelo entrenado con redes neuronales:  
<https://mit-cml.github.io/extensions/>

<sup>3</sup> FakeVoices: es un proyecto de MIT AppInventor, cuyo objetivo es reproducir un discurso en diferentes voces e idiomas que se generan al cambiar la velocidad y el tono del discurso.  
[https://appinventor.mit.edu/explore/resources/ai/fake\\_voices\\_unit](https://appinventor.mit.edu/explore/resources/ai/fake_voices_unit)

distinguir si se trata de una voz real o creada artificialmente (o voz sintética).

---

Se continuó trabajando en torno a esta idea mediante una serie de videos (DODO Island, 2021) (Vocal Synthesis, 2020) de casos actuales y conocidos por la mayoría de los estudiantes, en los que se manipulan las voces mediante el uso IA: Caso del actor Val Kirmer: se trata de un uso autorizado y legítimo, que permite que el actor recupere su “identidad vocal” desde una perspectiva de derecho.

a) Caso del rapero Jay-Z: se trata de un plagio en donde se pone la voz del cantante en un discurso que él no hizo, pudiendo comprometer su imagen y honor.

A modo de conclusión, las reflexiones a las que arribaron los estudiantes dan cuenta que la creación de voces sintéticas puede ser utilizada para recuperar derechos (como la voz), o también para causar perjuicios a las personas.

### **Segunda experiencia: Facultad de Informática**

Esta segunda experiencia recoge las observaciones del taller de JADiCC e incorpora algunas adecuaciones a su contenido vinculadas a la construcción y entrenamiento de modelos del Aprendizaje Automático.

Algunas de las reflexiones de los estudiantes de la experiencia previa, cuestionaban la clasificación de las imágenes del módulo LookExtention dado que algunos objetos no los reconoce como tal, ejemplo de ello fue que al intentar clasificar las partes de una computadora, la CPU la identificó como un módem. Claramente surge la idea que los datos con los que fue entrenado el modelo de LookExtention es insuficiente. Por ello, el equipo que lleva adelante este proyecto decidió incorporar la fase previa a la clasificación automática que es la de construir un modelo y entrenarlo. La herramienta seleccionada fue Google Teachable Machine (GTM) dado que es simple de usar y de comprender, accesible vía web.

Por otro lado, esta segunda experiencia de taller se desarrolló en dos encuentros de 2 horas de duración cada uno favoreciendo la inclusión de contenido sobre construcción y entrenamiento de modelos. A continuación se describen los destinatarios, los objetivos de aprendizaje, la selección de contenidos de IA trabajados, las actividades realizadas y las herramientas utilizadas, para el desarrollo del taller.

- **Destinatarios:** participaron del taller 10 estudiantes de 4to año de la EEST N° 9 de La Plata, de la especialidad Programación. En este caso se trata de estudiantes con formación científico-tecnológica del nivel secundario técnico de la provincia de Buenos Aires. El grupo contaba con conocimientos previos en programación y de la herramienta MIT App Inventor.

- **Objetivos de aprendizaje:**
  - Comprender cómo se construyen los modelos de clasificación para el Aprendizaje Automático.
  - Crear y entrenar un modelo a partir de datos propios sobre un dominio específico.
- **Contenidos de IA:** se continuó trabajando conceptos de Aprendizaje Automático y Aprendizaje Profundo para reconocer imágenes y, se incorporaron contenidos sobre creación y entrenamiento de modelos de clasificación. En relación a esto último, el proyecto GTM impulsado por Google, fue seleccionado como herramienta para acercar contenidos sobre creación de modelos para el Aprendizaje Automático dado que de manera simple, sin requerir conocimientos previos específicos de IA ni de programación, permite crear y entrenar modelos ágilmente mediante interacción visual. Además los modelos entrenados pueden ser incorporados a sitios web y aplicaciones, alentando los usos de la IA en la creación digital
- **Actividades realizadas:** la novedad en esta experiencia fue la incorporación de los conceptos de construcción, entrenamiento y testeo de los modelos que sustentan el Aprendizaje Automático.

**Creación de modelos de clasificación de imágenes:** se propuso a los estudiantes que piensen en un dominio de interés, para ello se dialogó sobre algunos ejemplos como clasificación de imágenes de animales, llegando a la conclusión que era necesario ser más específicos, por ejemplo clasificar imágenes de gatos o perros y se discutió sobre el propósito del modelo, interpellando sobre ¿para qué lo vamos a usar?

¿qué nos interesa clasificar? ¿gatos y perros o las razas a las que pertenecen?. Estas preguntas guiaron el diseño del modelo y permitieron introducir el concepto de “Clase”, que es clave para comprender cómo se construye un modelo de clasificación. La figura 3 muestra las diferentes fases del proceso de Aprendizaje Automático para clasificar imágenes, trabajado en la experiencia, que incluye dos clases de imágenes, perros y gatos:

Fase 1: construcción del modelo mediante la carga de imágenes de las 2 clases de interés, perros y gatos. En este ejemplo, se cargaron 10 muestras de perros y 7 de gatos.

Fase 2: algoritmo de entrenamiento, puede configurarse. Fase 3: testeo del modelo construido.



Figura 3. Fases de construcción de un modelo de clasificación para identificar imágenes de perros y gatos.

Durante la Fase 1, se guió a los estudiantes acerca del tipo de imagen a incluir en el modelo, en el sentido que sean representativas de la clase a la que pertenecen y a la cantidad de imágenes, dado que estos dos factores influyen en la calidad de la clasificación o el grado de confianza.

**Entrenamiento de modelos:** en esta fase se entrena el modelo de clasificación, en base a las imágenes provistas en la fase anterior. Esta tarea resultó sumamente transparente para los estudiantes, dado que se usó la configuración predeterminada por GTM, resumiéndose en presionar un botón. El equipo que llevó adelante el taller consideró que en esta primera experiencia con modelos de entrenamiento, incluir configuraciones de parámetros específicos de los algoritmos de entrenamiento no aportaría a la comprensión de cómo opera el Aprendizaje Automático para clasificar.

**Testeo de modelos de entrenamiento:** se les propuso a los estudiantes que testeen el modelo construido usando imágenes que no hayan utilizado en la fase 1 con la intención que puedan observar la calidad del modelo, es decir cuán certero resulta. En la figura 3 se puede observar que la imagen del gato es reconocida con un grado de confianza del 100% a pesar que esa imagen no formaba parte de la muestra. También se observó durante esta fase que varios estudiantes intentaron desafiar al modelo presentando imágenes de otros animales, concluyendo que el resultado no es determinante. Este punto resultó interesante para debatir sobre la relevancia de los datos de entrenamiento, concluyendo que cuánto más diversa sea la muestra dentro de una misma clase de objetos, el modelo construido es más confiable.

### Primeras conclusiones

Las iniciativas desarrolladas en el contexto del proyecto “Extensión en vínculo con escuelas secundarias” en torno al acercamiento de IA a estudiantes de escuelas secundarias del área de influencia de la UNLP arrojaron algunos datos acerca, por un lado, del interés de los estudiantes sobre “el detrás de escena de la IA” y por otro, sobre sus potenciales usos y

reflexiones en torno a las implicancias éticas. Las observaciones de las experiencias desarrolladas nos permitieron entender que a pesar que el término general IA es relativamente familiar para los jóvenes y en algunos casos pueden identificar dónde se usa la IA, la problematización de las diferentes formas de construir algoritmos basados en modelos de aprendizaje automático, cómo se prueban esas IA y el valor de la interpretación, resultaron conceptos desconocidos y de sumo interés. Mediante el intercambio con los estudiantes y la puesta en común de las actividades, surgió la reflexión que los contenidos que se publican en Internet alimentan las IA y en este sentido la relevancia de ser conscientes de esta realidad. En la misma línea de pensamiento, a través del análisis de casos reales en los que se usan IA se abordó la dimensión ética de estos usos, siendo éste un tema de debate y discusión.

En relación a las herramientas seleccionadas para el desarrollo de los talleres, MIT AppInventor y GTM, resultaron simples de entender y usar. En el caso de la primera de estas, la mayoría de los estudiantes venían trabajando en sus escuelas en la construcción de aplicaciones móviles, es por ello que desde el punto de vista del uso de la herramienta no se presentaron inconvenientes al introducir las extensiones de IA. En relación al uso GTM, al ser una herramienta simple y muy guiada, los jóvenes pudieron construir sus propios modelos de aprendizaje para clasificar imágenes. Aquí la relevancia estuvo puesta en comprender conceptos como “clases”, “grados de confianza” y “tamaño de la muestra de datos” e interpretar y evaluar los modelos construídos. En relación al tiempo destinado para el desarrollo del taller, en general fue suficiente, sin embargo la disparidad de conocimientos previos de los estudiantes es un punto a tener en cuenta en la planificación. Por otra parte, diseñar instrumentos didácticos que permitan analizar en términos objetivos la dimensión ética de los modelos construídos con las herramientas utilizadas, es un tema de trabajo futuro.

## Referencias

Bertochi, S., Navarro, J., Rodríguez, J., Cecchi, L. (2021). Enfoque didáctico para la enseñanza transdisciplinar de la inteligencia artificial. XVI Congreso de Tecnología en Educación & Educación en Tecnología - TE&ET 2021, La Plata.

Code Intf. EPCIA: Escuela de Pensamiento Computacional e Inteligencia Artificial (n.d). Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado. Ministerio de Educación y Formación Profesional, Gobierno de España.  
<https://code.intef.es/epcia/>

DODO Island (2021). VAL KILMER y su nueva voz generada con INTELIGENCIA ARTIFICIAL [Video]. YouTube.  
[https://www.youtube.com/watch?v=Fahwy\\_sBPTQ&t=799s](https://www.youtube.com/watch?v=Fahwy_sBPTQ&t=799s)

IBM (2023). AI vs. ML vs. DL vs. Neural Networks: What's the difference?.  
<https://www.ibm.com/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-network>

[s/](#)

INFoD (Instituto Nacional de Formación Docente). Ministerio de Educación de la Nación, Argentina (n.d.). Alfabetización mediática e informacional en la era de la inteligencia artificial (IA). Programa Nacional de Formación Docente. <https://infod.educacion.gob.ar/cursos/2010#>

Kimura I., Cao A., Garea A., Freccero B., Marcón P., Queiruga C. (2022). Un primer acercamiento al aprendizaje automático y sus usos (Taller). Actas de las Segundas JADiCC 2022 Corrientes: UNNE. Facultad de Ciencias Exactas. ISBN 978-987-3619-81-6. <https://repositorio.unne.edu.ar/handle/123456789/50765>.

Lee, I., Ali, S., Zhang, H., DiPaola, D. and Breazeal, C.. (2021). Developing Middle School Students' AI Literacy. Proceedings of the 52nd ACM Technical Symposium on Computer Science Education.

Marsland, Stephen (2015) Machine Learning: An Algorithmic Perspective – 2nd Ed. CRC Press.

MIT DAILY Curriculum. Disponible en: <https://raise.mit.edu/daily/>

The Artificial Intelligence for K-12 initiative (2020). The Artificial Intelligence (AI) for K-12 initiative (AI4K12) is jointly sponsored by AAAI and CSTA. The AI4K12 Initiative. <https://ai4k12.org/>

Toivonen T., Jormanainen I., Kahila J., Tedre M., Valtonen T. y Vartiaine H. Co-Designing Machine Learning Apps in K–12 With Primary School Children (2020). IEEE 20th International Conference on Advanced Learning Technologies (ICALT), Tartu, Estonia, 2020, pp. 308-310, doi: 10.1109/ICALT49669.2020.00099.

Vocal Synthesis. (2020). Jay-Z raps the "To Be, Or Not To Be" soliloquy from Hamlet (Speech Synthesis) [Video]. YouTube. <https://www.youtube.com/watch?v=m7u-y9oqUSw>.

Nielsen, Michael A. (2015). Neural Networks and Deep Learning, Determination Press.

## Taller AYDS-IS: propuesta didáctica para la integración de Análisis y Diseño de Sistemas e Ingeniería de Software

Marcela Daniele, Marcelo Uva, Franco Brusatti, Ariel Arsaute, Daniela Solivellas, Mariana Frutos  
{marcela,fbrusatti,uva,aarsaute,dsolivellas,mfrutos}@dc.exa.unrd.edu.ar

Departamento de Computación  
Facultad de Ciencias Exactas, Físico-Químicas y Naturales  
Universidad Nacional de Río Cuarto

### Resumen

En este artículo se presenta una propuesta didáctica con modalidad taller, que funciona como eje integrador del desarrollo de las asignaturas cuatrimestrales y correlativas, “Análisis y Diseño de Sistemas” e “Ingeniería de Software”, que corresponden al tercer año de las carreras de Computación que ofrece la Facultad de Ciencias Exactas, Físico-Químicas y Naturales de la Universidad Nacional de Río Cuarto. *Taller AyDS-IS* es una propuesta didáctica que, fundamentalmente, permite articular los contenidos de las asignaturas y afianzar los aprendizajes, a partir del desarrollo y gestión de un proyecto de software sobre la base de un trabajo colaborativo y sostenido durante todo el año lectivo. En el marco de *Taller AyDS-IS*, se plantea la solución genérica a un problema a partir del desarrollo de un sistema de software. Para ello, se lleva adelante un proyecto donde se exponen y abordan metodologías ágiles, y diversas tecnologías y herramientas actuales que permiten el diseño, desarrollo y gestión del proyecto. Los resultados han sido muy positivos y enriquecedores, logrando que los estudiantes adquieran una experiencia cuasi-real y culminen el *Taller AyDS-IS* con aprendizajes significativos, que les permite abordar con mayor agilidad sus trabajos finales de carrera, potenciar su graduación e inserción laboral, como así también fortalecer la relación Universidad-Medio socio productivo.

**Palabras Clave:** Ingeniería de Software, Análisis y Diseño de Sistemas, Taller de desarrollo de software.

### Introducción

La propuesta didáctica presentada en esta experiencia áulica, tiene como objetivo principal fortalecer la integración curricular y enfoques interdisciplinarios en la formación de los estudiantes de las carreras de computación de la UNRC. *Taller AyDS-IS*, desarrollado en las asignaturas, cuatrimestrales y correlativas, denominadas “Análisis y Diseño de Sistemas” e “Ingeniería de Software”, busca articular, integrar y aplicar los contenidos brindados a los estudiantes en un contexto práctico y cercano a la realidad de desarrollo de software en la

Industria. El trabajo propuesto implica adquirir y aplicar conocimientos de metodologías, tecnologías/herramientas para el desarrollo de software y gestión del proyecto, y en los últimos años se enfocan principalmente en el desarrollo de ambientes frontend y backend necesarios para la implementación de aplicaciones web.

La ejecución del taller se divide en dos etapas. La primera de ellas, en la asignatura Análisis y Diseño de Sistemas, está focalizada en las actividades establecidas por una metodología ágil que guía la construcción de un producto de software que incluye todas las etapas del ciclo de vida de desarrollo desde la ingeniería de requerimientos hasta la prueba del producto. En una segunda etapa, en la asignatura "Ingeniería de Software", se incluyen las actividades necesarias que posibiliten una gestión exitosa del proyecto, trabajando con técnicas y tecnologías que posibiliten incorporar las actividades que conforman la planificación del proyecto de software, incluyendo estimación, calendarización, análisis de riesgos, gestión de la calidad y gestión del cambio.

El equipo docente, con una experiencia consolidada de más de diez años en el área, incorpora y propone actualizaciones y nuevos desafíos, por lo que el taller se define como un espacio dinámico y en evolución que se ajusta a las necesidades de la industria. Esto colabora a la formación de los estudiantes brindando un ambiente de aprendizaje actualizado y acorde a las demandas del mercado laboral.

Además de integrar los contenidos de las asignaturas "Análisis y Diseño de Sistemas" e "Ingeniería de Software", el taller permite a los estudiantes hacer uso de sus conocimientos en resolución de problemas, base de datos, diseño de algoritmos, y otras áreas temáticas estudiadas en la carrera.

El trabajo colaborativo es una parte esencial del taller, ya que les brinda a los estudiantes la oportunidad de trabajar en equipos de desarrollo, fijando objetivos comunes, analizando posibles soluciones, intercambiando saberes, resolviendo conflictos, tomando y fundamentando decisiones, ocupando diferentes roles establecidos por la metodología utilizada, entre otros. Esto resulta fundamental para que los estudiantes puedan desarrollar lo que llamamos las "habilidades blandas" fundamentales e indispensables para formar equipos de trabajo en su futura vida profesional.

*Taller AyDS-IS* se desarrolla con un enfoque dinámico y de exploración que incentiva a los estudiantes a adquirir estrategias para mantenerse actualizados en un campo tan cambiante como la industria del software. La capacidad de adaptarse y aprender nuevas tecnologías es esencial en el mundo actual, estimulando así el desarrollo de estas habilidades necesarias para mantenerse competitivo en el mercado laboral.

A continuación se describe la propuesta detallando el contexto de aplicación y los contenidos trabajados. Luego se presenta la metodología de trabajo y técnicas, tecnologías y herramientas aplicadas. Finalmente, se presentan algunos resultados observados en las distintas ediciones del dictado del taller.

### **Descripción de la experiencia**

A continuación se describe el contexto dentro del cual se desarrolla la propuesta conjuntamente con los contenidos curriculares trabajados.

**Contexto:** La propuesta didáctica se enmarca dentro de las asignaturas “Análisis y Diseño de Sistemas” e “Ingeniería de Software” en la UNRC. Estas asignaturas son fundamentales e integrales para la formación de futuros profesionales en el campo de la computación y el desarrollo de software. "Análisis y Diseño de Sistemas" es una asignatura que se ubica en el primer cuatrimestre de tercer año de las carreras de Analista en Computación, Profesorado y Licenciatura en Ciencias de la Computación de la UNRC, mientras que "Ingeniería de Software" se encuentra en el segundo cuatrimestre del mismo año, y es correlativa a la primera. En la actualidad, el equipo docente a cargo del desarrollo de estas asignaturas está conformado por una profesora adjunta con dedicación simple a cargo de los teóricos, una Jefe de Trabajos Prácticos encargada del diseño y coordinación de los prácticos, y tres Ayudantes de Primera (uno con dedicación exclusiva y los otros con dedicación semi-exclusiva). Uno de los docentes auxiliares está a cargo de coordinar la ejecución y seguimiento de los equipos de estudiantes que realizan el taller. Este docente posee una alta experiencia en desarrollo de software para la industria nacional e internacional, ya que se desempeña también en ámbito productivo. Esta combinación de experiencia académica y profesional en el equipo docente, favorece la constante actualización e incorporación de nuevas tecnologías en los contenidos abordados y presentados para la realización del *Taller AyDS-IS*.

La cantidad promedio de estudiantes que cursan estas asignaturas es de 30 (treinta) por año. Esto facilita un ambiente propicio para el aprendizaje colaborativo y la interacción estrecha entre estudiantes y con docentes.

**Contenidos:** Cada año se propone una situación problema para ser abordada a través de una solución informática con un proyecto de software, que pasa por todas las etapas del ciclo de vida de desarrollo de un software como así también por las actividades sombrija necesarias para el gerenciamiento del mismo. Por ejemplo en la edición 2022 y en el contexto del Mundial de Fútbol QATAR 2022, se presentó a los estudiantes el desafío de crear un sistema de pronósticos deportivos al estilo PRODE. Este sistema tuvo como objetivo principal la carga de un calendario de partidos de la Copa del Mundo, junto con las predicciones de los usuarios.

Cada equipo tuvo la libertad de diseñar la lógica del sistema, incluyendo la asignación de puntos, penalizaciones por predicciones incorrectas y otros aspectos de la mecánica del juego. El entusiasmo generado por este desafío llevó a algunos equipos a proponer extensiones para permitir pronósticos en diferentes ligas de fútbol, mientras que otros se centraron en enriquecer las funcionalidades del sistema. En el taller, se aplicó con éxito la metodología ágil SCRUM, adaptándola a los tiempos y a la dinámica educativa. Los equipos tuvieron que planificar y estimar sus actividades para cumplir con el cronograma de entregas establecido para el taller, lo que situó a los estudiantes en roles de gestión y administración de proyectos. La necesidad de equilibrar sus compromisos con el taller y otras asignaturas demandó una cuidadosa planificación y organización.

El proyecto se desarrolló haciendo uso del framework Sinatra, que proporcionó un entorno ágil para el desarrollo web. Ruby se destacó como el lenguaje de programación principal utilizado en la implementación del sistema, lo que permitió un desarrollo eficiente y flexible. Para la gestión del proyecto se utilizó Pivotal Tracker y para la gestión de la configuración del software GIT y la plataforma colaborativa GitHub. Todas estas tecnologías fueron integradas a fin de lograr el seguimiento y control de las actividades de acuerdo a lo planificado. Para mantener y verificar la calidad del código, se emplearon gemas de RSpec para la generación de Test Suites, lo que permitió incrementar la calidad del código logrando la identificación temprana de fallas. La combinación de estas tecnologías y herramientas desempeñaron un papel fundamental para el éxito de la propuesta.

En la edición 2023, se planteó el desarrollo de un sistema educativo mediante la modalidad de una trivía interactiva. Cada equipo decidió el dominio temático que deseaba enseñar. El sistema debía permitir gestionar usuarios, proporcionar contenido educativo sobre el tema elegido y luego ofrecer funcionalidades de evaluación para medir el grado de aprendizaje alcanzado. La libertad de elección del tema entusiasmó enormemente a los equipos quienes actualmente se encuentran desarrollando una amplia variedad de sistemas educativos. Algunos proyectos han optado por enseñar el lenguaje de señas, artes culinarias, normas de tránsito, programación, matemáticas, entre otras.

El conjunto de herramientas y tecnologías utilizadas en esta edición es similar a la edición 2022. Sin embargo, se han incorporado algunas herramientas adicionales al proyecto para mejorar la calidad del software. Entre estas se encuentra SimpleCov, una gema que permite medir el grado de cobertura logrado por los tests. El taller 2023 se encuentra actualmente en ejecución.

El *Taller AYDS-IS* propone una amplia variedad de temas relacionados con el desarrollo de software, los cuales están fundamentados en los contenidos curriculares de las asignaturas

"Análisis y Diseño de Sistemas" e "Ingeniería de Software", trabajados de manera teórica y práctica. Los temas principales que se trabajan en el taller son los siguientes:

**Metodologías Ágiles (particularmente SCRUM):** En el contexto actual de la industria del desarrollo de software, las metodologías ágiles han ganado una gran popularidad debido a su capacidad para adaptarse rápidamente a los cambios y entregar productos de alta calidad en tiempos cortos. SCRUM es una de las metodologías ágiles más utilizadas, y en el taller, se profundiza en sus principios y prácticas para que los estudiantes comprendan cómo aplicarla exitosamente.

**El Lenguaje de Modelado Unificado (UML):** Este lenguaje provee una serie de modelos gráficos para la representación de los sistemas en diferentes momentos del ciclo de vida de un producto de software. Durante las etapas de análisis del proyecto, se hace uso de varios tipos de diagramas de UML: Diagramas de Clases, Diagramas de Objetos, Diagramas de Actividades y Diagramas de Secuencia, entre otros. Estos diagramas permiten visualizar las clases que conforman el sistema, sus relaciones y atributos, así como representar instancias de las clases y cómo interactúan entre sí en un momento específico del proceso.

**Patrones de Diseño:** Los patrones de diseño son soluciones probadas y repetibles a problemas comunes en el desarrollo de software. En el taller, se aplican patrones de diseño y se fomenta su utilización para mejorar la calidad, la eficiencia y la flexibilidad del código.

**Testing y Generación de Test Suites:** La calidad del software es fundamental, y el testing juega un papel clave. Durante el taller, los estudiantes aprenden a generar test suites basadas en criterios de cobertura de caja blanca estudiados en las asignaturas de Análisis y Diseño de Sistemas. Esta práctica permite asegurar que diferentes partes del código sean probadas, mejorando la calidad y robustez del software desarrollado.

Para mejorar la eficiencia y la repetibilidad del proceso de testing, se incorporan herramientas que permiten la automatización de casos de prueba. Los estudiantes aprenden a utilizar estas herramientas para realizar pruebas automáticas y facilitar la identificación temprana de errores en el software.

**Gestión de Proyectos, Planificación y Estimación en Desarrollos Ágiles:** La gestión adecuada de proyectos es esencial para el éxito en el desarrollo de software. Se abordan las técnicas de planificación como por ejemplo, diagramas de PERT y Gantt junto a técnicas de estimación específicas de las metodologías ágiles como por ejemplo Planning Poker para lograr cumplir con los tiempos establecidos para el proyecto.

**Gestión de Configuración de Software:** La gestión de la configuración implica controlar y rastrear cambios en el código y los artefactos del proyecto. Se instruye a los estudiantes sobre las mejores prácticas para la gestión de configuración. En el taller se brindan detalles técnicos para la creación y manipulación de ramas dentro de un sistema de control de versiones como por ejemplo Git y se dan nociones de los principales flujos de trabajo adoptados por la Industria: flujo de trabajo centralizado, modelo de ramificación por función (Feature Branch Workflow), modelo de ramificación Git-Flow, etc.

**Métricas Orientadas al Proceso, al Producto y las Personas:** El uso de métricas en el desarrollo de software es esencial para evaluar el progreso del proyecto, medir la calidad del software y evaluar el rendimiento del equipo. Se exploran diferentes métricas y cómo aplicarlas de manera efectiva para mejorar el proceso de desarrollo y el producto final.

La combinación de estos conceptos permite a los estudiantes adquirir una comprensión integral y práctica.

### **Metodología**

El taller AYDS-IS constituye una de las instancias evaluativas que junto a parciales, exposición de un tema, resolución de uno o más problemas integradores de seguimiento, conforma y completa el proceso evaluativo de las asignaturas mencionadas. La planificación de los contenidos de las asignaturas involucradas se realiza de manera estratégica, buscando que sean abordados en momentos previos y cercanos en el desarrollo del taller. Esta planificación permite al estudiante establecer vínculos significativos entre los contenidos teóricos y las prácticas aplicadas en la actividad del taller, para lo cual se cuenta con un detallado cronograma que sirve de guía para el equipo de formadores y estudiantes.

El taller se planifica en un encuentro semanal de dos horas, donde el docente coordinador presenta el proyecto y proporciona detalles sobre la metodología, herramientas y actividades a cumplimentar. Los estudiantes trabajan en sus proyectos y reciben orientación del cuerpo docente. Entre otras tareas, los equipos deben investigar y adoptar las herramientas y técnicas propuestas dentro de sus proyectos. Esta modalidad promueve la interacción activa de los estudiantes y les brinda la oportunidad de aplicar rápidamente lo aprendido en un entorno práctico. Por otro lado, el taller prevé horarios de consulta tanto de manera virtual como presencial, para brindar asesoramiento adicional y responder las inquietudes de los equipos de desarrollo. Cada equipo requiere un tiempo extra de trabajo semanal de aproximadamente 6 horas fuera de los encuentros presenciales para investigar, realizar reuniones y tomar decisiones. Los equipos deberán realizar el análisis del problema, generar los modelos correspondientes y posteriormente durante los encuentros presenciales presentarán sus

propuestas de solución. Luego, junto al cuerpo docente, se establecerá la frontera del sistema definiendo el documento de Especificación de Requisitos de Software (ERS) y Minimum Viable Product (MVP), de acuerdo a los tiempos posibles y la complejidad de las funcionalidades propuestas. Seguidamente, se estiman y planifican los *sprints*. Cada equipo inicia el primer *sprint* de su proyecto, realiza el diseño, la implementación y la prueba de las funcionalidades planificadas. Luego de finalizado el *sprint*, se desarrolla una reunión de retrospectiva analizando el avance logrado, dificultades y las decisiones tomadas. Dicho análisis permite ajustar detalles y metodologías de trabajo para iniciar el siguiente *sprint*.

En el primer cuatrimestre del 2023, se planteó el desarrollo de un proyecto de software del tipo preguntas y respuestas (trivia) que permitiera al usuario autoevaluar sus conocimientos sobre un tema particular y/o ampliar información sobre el tema, brindando a los equipos la libertad de definir su dominio. Este tipo de propuestas, promueve la libertad, creatividad y la innovación en el desarrollo de los proyectos, ya que los estudiantes pueden abordar temas que les apasionen y estén alineados con sus intereses y habilidades.

Al finalizar cada cuatrimestre los equipos realizan una presentación en donde exponen el proceso realizado, mostrando no sólo el producto final, sino también comentando sus experiencias, dificultades y conocimientos adquiridos.

El taller proporciona a los estudiantes la oportunidad de integrar y aplicar de manera concreta los conocimientos y habilidades técnicas adquiridas en las asignaturas, permite una formación más sólida y prepara a los estudiantes para enfrentar los desafíos del mundo laboral en el campo del desarrollo de software.

### **Técnicas, herramientas y tecnologías utilizadas**

El desarrollo del taller se caracteriza por mantener un stack de herramientas altamente dinámico, lo que favorece su continua actualización y adaptación a las últimas tendencias en el desarrollo de software. Esta dinámica se refleja en la actualización o variación de distintos Object-Relational Mapping (ORM), gestores de bases de datos (DBMS), frameworks y lenguajes de programación utilizados. Además de las herramientas específicas utilizadas para la codificación del producto de software, se incluye una serie de herramientas que conforman el stack utilizado para la creación de modelos, para la gestión y planificación del proyecto en general.

A continuación, se enumeran las herramientas y tecnologías que han sido incluidas en las últimas ediciones del taller:

**Lenguajes de Programación:** Los estudiantes han trabajado con una variedad de lenguajes de programación. Ruby, reconocido por su flexibilidad y simplicidad, es utilizado en el desarrollo web y otras aplicaciones. JavaScript es ampliamente utilizado en el desarrollo web para implementar la lógica del cliente. Además, HTML es el lenguaje utilizado para estructurar y dar formato a los contenidos web, mientras que CSS es utilizado para definir el estilo y la presentación de los elementos en las interfaces de usuarios web.

**Frameworks Web:** El desarrollo del taller se enfoca en la creación de una aplicación web, para lo cual se utiliza un framework de desarrollo específico. En las últimas ediciones del taller se han utilizado los frameworks de desarrollo web Sparck y Sinatra. Spark es un micro framework web para Java, inspirado en Sinatra. Al igual que Sinatra, Spark se caracteriza por su sencillez y eficiencia, lo que permite a los estudiantes construir aplicaciones web robustas. Sinatra: es también un micro framework web en Ruby, conocido por su simplicidad y facilidad de uso. Sinatra es ideal para desarrollar aplicaciones web rápidas y eficientes, especialmente para proyectos pequeños o prototipos.

La elección del framework web determina la arquitectura y las herramientas disponibles para el desarrollo de la aplicación web.

**Object-Relational Mapping (ORM):** En el desarrollo del taller, los equipos han utilizado ORM para definir la interacción entre el modelo de datos y la base de datos. En el desarrollo del taller de las últimas ediciones del taller, se han utilizado dos ORM: SEQUEL y ActiveRecord. SEQUEL es un ORM para Ruby, conocido por su simplicidad y flexibilidad. Permite a los estudiantes interactuar con la base de datos mediante objetos y métodos, simplificando las operaciones CRUD y manejando consultas complejas. Por otro lado, ActiveRecord, parte del framework Ruby on Rails y ofrece una forma sencilla de interactuar con la base de datos a través de consultas orientadas a objetos. Ambos ORM son dos alternativas robustas y confiables que facilitan el proceso de desarrollo y permiten a los estudiantes centrarse en la implementación de funcionalidades y la lógica de negocio.

**Gestión de proyectos:** La gestión adecuada del proyecto de desarrollo durante la ejecución del taller es fundamental para lograr cumplir con los objetivos planteados. Para ello se incorporan herramientas que faciliten esta tarea. Algunos de los gestores utilizados por los equipos han sido dos aplicaciones especialmente diseñadas para la gestión de proyectos desarrollados utilizando metodologías ágiles, estos son Pivotal Tracker y Trello.

Pivotal Tracker permite a los equipos planificar, priorizar, rastrear y gestionar el progreso de sus proyectos. Los estudiantes pueden crear historias de usuario que representan las

funcionalidades del sistema, asignar puntos de historia para estimar su complejidad y establecer un plan de iteraciones. Además brinda una visión clara y en tiempo real del avance del proyecto, lo que facilita la toma de decisiones e identificación de posibles inconvenientes. Por otro lado Trello, es otra herramienta popularmente utilizada para la gestión colaborativa de proyectos, cuenta con una interfaz visual e intuitiva. Trello permite a los estudiantes crear tableros que representan las etapas del desarrollo del proyecto, listas que contienen tareas y tarjetas que representan las funcionalidades a implementar. Los estudiantes pueden asignar tarjetas a miembros del equipo, establecer fechas límite y mantener un registro detallado de su progreso.

**Herramientas para la gestión de configuración de software:** La gestión eficiente de la configuración del software es esencial para el desarrollo colaborativo de proyectos. En el taller, se utiliza Git como sistema de control de versiones. Git establece un ambiente distribuido y ampliamente utilizado en el desarrollo de software. Los equipos de estudiantes pueden mantener un historial completo de cambios, trabajar en ramas de manera aislada y fusionar cambios de manera ordenada. También pueden deshacer cambios en caso de errores, brindando seguridad adicional en el proceso de desarrollo.

Los proyectos de los estudiantes son gestionados por Pivotal Tracker (o algún otro gestor de proyectos) y alojados en GitHub. Esta última es una plataforma de alojamiento y colaboración para proyectos que utilizan Git como sistema de control de versiones. Los equipos en el taller se benefician de GitHub, especialmente en lo que respecta a la gestión de workflows para el manejo de ramas. Los sistemas de gestión de proyectos son vinculados a los repositorios como GitHub para mantener una documentación en tiempo real del avance de cada tarea que se realiza dentro del proyecto.

### **Resultados Observados**

A lo largo de las distintas ediciones del Taller AYDS-IS se han observado resultados sumamente positivos y alentadores. A pesar de la diversidad de los equipos de estudiantes que participan cada año, se ha mantenido una muy buena dinámica de trabajo.

Una ventaja destacable del Taller, es la adquisición de la experiencia con metodologías ágiles. Los equipos tienen la oportunidad de trabajar con una metodología ágil como SCRUM. Esta experiencia les permite comprender los beneficios de una gestión de proyectos ágil, donde la adaptabilidad, la colaboración y la comunicación efectiva son fundamentales para el éxito del desarrollo de software.

Los estudiantes trabajan en equipos colaborativos, lo que fomenta la toma de decisiones conjuntas y la distribución efectiva de tareas y responsabilidades. Esta dinámica de trabajo en

equipo fortalece las habilidades de comunicación de los estudiantes y su preparación para la vida profesional.

Cada uno de los equipos de estudiantes que afrontan el Taller, tienen la oportunidad de autogestionarse, organizando su trabajo y definiendo sus propias estrategias para cumplir con los objetivos del proyecto. Esto les permite adquirir estrategias y desarrollar actitudes y habilidades para el aprendizaje autónomo a lo largo de su vida profesional tan necesario en los tiempos actuales.

Por otro lado, el uso de tecnologías y metodologías nuevas suele implicar una curva de aprendizaje inicial de media a alta, por lo que está cuidadosamente considerado y planificado ya que puede requerir un esfuerzo adicional por parte de los equipos para adaptarse y comprender completamente su funcionamiento.

Es de destacar, que a partir de la dinámica y organización del taller, junto al desarrollo de toda la asignatura y cambios introducidos en la modalidad de evaluación, fundamentado en una valoración mucho más centrada en el proceso de aprendizaje, se han fortalecido y mejorado significativamente las tasas de graduación de la carrera Analista en Computación fundamentalmente, ya que los estudiantes están mejor preparados para afrontar con mayor agilidad el Proyecto Final de esta carrera, y generando consecuentemente un lógico impacto positivo en el abordaje y finalización de la Licenciatura. También se visualiza un impacto favorable para la inserción laboral y una mejora en los puestos laborales a los que acceden los graduados de las carreras de computación de la UNRC.

## Referencias

Beck, K., Beedle, M., Bennekum, A. V., Cockburn, A., Cunningham, W., Fowler, M., & Thomas, D. (2001). Agile Manifesto for Software Development. Recuperado de <https://agilemanifesto.org/>

Active Record. (s.f.). GitHub. <https://github.com/rails/rails/tree/main/activerecord>

Sinatra Active Record. (s.f.). GitHub. <https://github.com/sinatra-activerecord/sinatra-activerecord>

Docker. (s.f.). <https://www.docker.com>

Hypertext Transfer Protocol -- HTTP/1.1. (2003). RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1. Recuperado de <https://www.ietf.org/rfc/rfc2616.txt>

Ruby Version Manager - RVM. (s.f.). <https://rvm.io/rvm/install>

Bundler (RubyGem). (s.f.). <https://bundler.io/gemfile.html>

Beck, K. (2004). Extreme Programming Explained: Embrace Change. Addison-Wesley.

- Rubin, K. S. (2012). Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley Professional.
- Cohn, M. (2005). Agile Estimating and Planning. Prentice Hall.
- Sutherland, J. V. (2014). SCRUM: The Art of Doing Twice the Work in Half the Time. Crown Business.
- Scrum Alliance. (s.f.). <https://www.scrumalliance.org/>
- Ruby. (s.f.). <https://www.ruby-lang.org/en/>
- Sinatra. (s.f.). <http://sinatrarb.com/>
- ERB. (s.f.). <http://www.kuwata-lab.com/erubis/users-guide.html>
- Sequel. (s.f.). <http://sequel.jeremyevans.net/>
- IEEE. (1998). IEEE 830-1998 - IEEE Recommended Practice for Software Requirements Specifications. IEEE. Recuperado de <http://standards.ieee.org/findstds/standard/830-1998.html>

## **Actividades de aplicación en curso de análisis de los datos para Ingeniería sobre sesgo de género, abordaje de la equidad e inclusión**

Verónica Herrero  
veronica.herrero@uesiglo21.edu.ar

Universidad Siglo 21

### **Resumen**

En la educación universitaria en general, es relevante generar consciencia sobre los sesgos y posibles efectos negativos del uso de datos en distintas etapas. En particular, en las disciplinas que aplican análisis y puesta en valor de los datos para la toma de decisiones, resulta imperativo desarrollar esta comprensión junto con las diferentes competencias de pensamiento crítico requeridas. Este principio tiene relación con la formación de profesionales involucrados y socialmente responsables, en todas las áreas disciplinares.

El trabajo presenta aspectos del desarrollo de una experiencia de involucramiento activo con aspectos vinculados con los sesgos y las precauciones en el análisis de datos en un curso universitario de una carrera de Ingeniería. La propuesta desarrollada incluye la lectura crítica, la reflexión, la puesta en común y el uso de posibles herramientas innovadoras por parte de los propios protagonistas del aprendizaje.

**Palabras claves:** Gestión de datos – Género en datos – Educación universitaria – Actividades inmersivas

### **Introducción**

El análisis de datos requiere capacidad de indagar, disponer de apertura a nuevas perspectivas, así como sistematicidad y rigurosidad metodológica. Notoriamente, el contenido de los programas académicos de esta disciplina no suele enfocarse de manera tan amplia. Predominan los temas y las actividades en la enseñanza de análisis de datos que ponen foco sólo en elementos técnicos. Por una parte, se comprende este énfasis. Deben sintetizarse en estos programas elementos provenientes de distintas ciencias y hay una carga importante de destrezas que se adquieren desde sus prácticas. La inquietud que surge tiene relación con los aspectos de abordaje integral del análisis de datos que pueden quedar fuera de los programas o de los abordajes aplicados.

El principal propósito de la actividad que se desarrolló fue introducir las diferentes problemáticas asociadas con sesgos en los datos y sus consecuencias. El enfoque incluye a los distintos grupos (predominantemente minoritarios) que pueden ser afectados por las decisiones en la manera de tratar los datos. Adicionalmente, se buscó una reflexión profunda y crítica sobre las creencias que atribuyen a todo análisis cuantitativo un halo de total objetividad.

Cuestionar estas calificaciones favorecedoras de lo cuantitativo, que por sí misma también contiene sesgos y prejuicios, resulta relevante en la formación de futuros profesionales de una disciplina enfocada en la innovación desde la tecnología. El objetivo de este trabajo es compartir y generar reflexiones acerca del proceso de vinculación de los estudiantes con material que interpela al análisis tradicional de datos, así como detectar oportunidades de aplicar metodologías participativas y creativas, en materias afines.

### **Marco conceptual**

La diversidad en la universidad como espacio social, se relaciona con la concepción de sujetos que interactúan asumiendo y respetando “expresiones culturales, identidades plurales y originales” (Alvarez Castillo y Garcia-Cano Torrico, 2022, p. 22) en los ámbitos de realización concreta de sus actividades. Los grupos que interactúan en el contexto universitario, estudiantes, docentes y personal de apoyo, junto con el resto de los actores con los cuales se relacionan en el entorno de aprendizaje y fuera de él. Los diferentes espacios de actuación corresponden a situaciones donde se pone en juego una visión, que trascenderá las experiencias previas y terminará de forjar las características de cara a la convivencia en la sociedad en general, en el presente y en futuro.

La posibilidad de cuestionar y cuestionarse ayuda en diversos ámbitos del desarrollo, en este caso sometiendo también a escrutinio los resultados que provienen de la ciencia, que en gran medida goza de un halo de perfecta objetividad (D'Ignazio, Klein, Livio, Niederer y Colombo, 2021, Mato, 2018). En este contexto de ideas, la actividad desarrollada tiene en cuenta la búsqueda de herramientas concretas que, en el ámbito del contenido de una materia, en una carrera de las denominadas “duras”, puedan expandir la mirada y considerar la importancia y el valor de contribuir a un contexto más diverso, más inclusivo, más equitativo (UNESCO, 2020). Para el caso de la educación en general y particularmente la universitaria, se detectan patrones recientes de notoria falta de heterogeneidad (Juárez Núñez y Comboni Salinas, 2020).

Se busca que el análisis de datos en ámbitos académicos, pero también en la futura vida profesional de los estudiantes, cuando como profesionales, trate la información, conscientes de los sesgos y sus consecuencias. Las actividades desarrolladas también intentan considerar rasgos propios del docente innovador y orientado a la inclusión Fernández Martínez, et al. (2020): creando un clima de confianza en el aula, ofreciendo alternativas de aprendizaje y motivación a través de resoluciones con iniciativas propuestas por los estudiantes. El enfoque general de abordaje se inscribe dentro de la adopción de la Universidad de la educación positiva (Seligman, 2011).

### **La temática y su vínculo con el programa de la materia**

El programa de la materia es notoriamente técnico y se basa fundamentalmente en las nociones de big data, en cuanto a fenómeno como a la evolución histórica, las cuestiones relativas a la obtención de datos masivos y al almacenamiento.

Si bien se trata de conceptos técnicos y se ejemplifican a través de casos prácticos de empresas, se detectó un vacío en el tratamiento de aspectos como los sesgos, la confidencialidad, la privacidad de datos sensibles y los posibles usos socialmente indeseados o limitados de datos a gran escala.

La propuesta descrita se enfocó en complementar el enfoque con aspectos trascendentes vinculados con las decisiones extra técnicas en el contexto de big data y la sociedad de la información en general.

Asimismo, la problemática no se encuentra actualmente comprendida en ninguna de las materias obligatorias del plan de estudio de la carrera, y el aporte incorpora por una parte contenidos relevantes no priorizados, como también, ejercita metodológicamente la crítica y el debate respetuoso.

La idea de analizar los diferentes sesgos con énfasis en los correspondientes al género se vincula con la evidencia y los estudios disponibles que dan cuenta de las enormes desigualdades en materia de género que atraviesan los espacios relacionados con la tecnología (Degli Esposti, 2021, CEPAL, 2017).

### **La propuesta de actividad en el marco de la evaluación académica**

El diseño instruccional del cursado en modalidad presencial en la Universidad Siglo 21 contiene entre otros elementos evaluativos, una sección enfocada en el proceso. Esta instancia de evaluación se centra en diferentes actividades de construcción y desarrollo a lo largo del cursado, de modo que promueven la realización de tareas y elaboraciones a partir de cuatro abordajes didácticos sugeridos: proyectos, problemas, casos y retos. En esta materia se optó por trabajar con la aplicación de Didáctica por proyectos, en la cual se enmarcó el desarrollo de la experiencia.

La actividad de proceso en sí es evaluable y tiene en cuenta una planificación semestral, una serie de entregables y una rúbrica que establece los criterios que se tienen en cuenta para evaluar, así como la importancia relativa y las evidencias que deben considerarse para asignar la nota.

Por normativa institucional, los alumnos reciben en el primer día de clases ambos instrumentos, la planificación semestral (denominada planificación unificada) y las consignas de la nota de proceso.

La experiencia se desarrolló en el curso de Gestión de datos e información de la carrera de Ingeniería en innovación de la Universidad Siglo 21, en el curso del año 2023, en modalidad presencial. El curso tuvo 24 estudiantes. La materia corresponde al cuarto año de la carrera. Si bien los contenidos principales se orientan a aspectos técnicos de la gestión de grandes volúmenes de datos y diversas implicaciones tecnológicas y de negocios, se complementan los contenidos teóricos con actividades de aplicación. La actividad evaluativa de procesos se planteó considerando una bibliografía asociada con la cuestión de género y diferentes dimensiones de la discriminación presentes en los datos. El texto usado como base corresponde a Data Feminism (de las autoras Catherine D'Ignazio and Lauren Klein, en su versión traducida al español recientemente).

La actividad consistió en equipos muy reducidos (2 o 3 estudiantes), desarrollando cada uno de los capítulos desde un enfoque descriptivo (conceptos, ejemplos e ideas) y crítico (considerar otros aspectos no establecidos, discusiones y miradas en oposición justificada con lo planteado). Además, y quizá lo que resulta de alguna manera innovador, fue proponer a los estudiantes la presentación del contenido de maneras totalmente libre en cuanto a los recursos, formatos y materiales, con la restricción de uso de 30 minutos de la clase como máximo. Los grupos presentaron un capítulo semanalmente hasta completar todo el texto.

### Los desarrollos de los estudiantes

Los alumnos eligieron las maneras de compartir cada temática utilizando diversos recursos, entre los que se destacan los aspectos multimediales, utilización de redes sociales y generación de actividades lúdicas y competitivas en el auditorio (el resto del curso que no exponía en cada clase).



Figura 1: Escena de un video estilo TikTok producido por los estudiantes

Fuente: Publicación en cuenta generada por los alumnos del curso en Instagram, del video relacionado.

Uno de los grupos eligió representar a través de un video la historia de discriminación interseccional (género / raza) de una de las pioneras de la computación, relatada en el texto. Para ello, generaron un video con estilo de la red social TikTok (Figura 1) entre las alumnas expositoras y editaron la historia. Una actividad que tuvo un trabajo de producción que comentaron que llevó varias horas, para compartir y sensibilizar al curso con este hecho histórico. Otros grupos eligieron usar juegos, uno de verdadero / falso, recogiendo las creencias del curso sobre diferentes afirmaciones, que parecen a priori falsas por contener hechos de notoria discriminación, pero que sin embargo son verdaderos.



Figura 2: Alumnos respondiendo si creían que las afirmaciones eran Verdaderas o Falsas, utilizando banderitas con colores rojo (Falso) y verde (Verdadero)

Fuente: registro propio durante el desarrollo de la actividad.

Un juego que usaron otros alumnos, concentrados en focalizar en cuatro conceptos clave fue un juego de ahorcado. Una vez que se develaba la palabra, explicaban y daban contexto en el marco del tema estudiado.

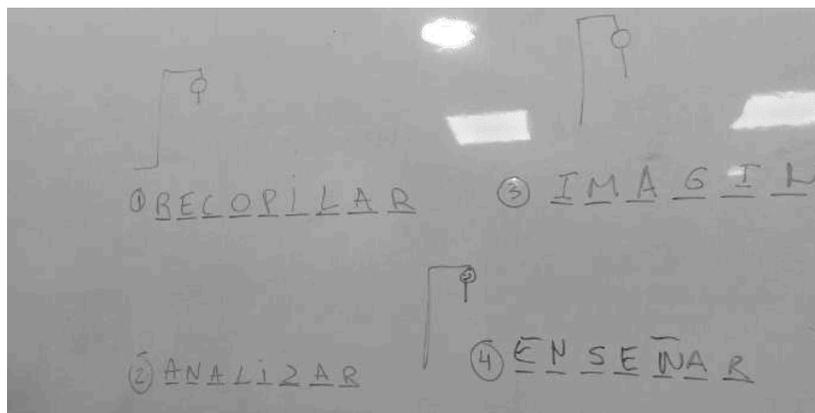


Figura 3: Palabras clave en juego de ahorcado  
Fuente: registro propio durante el desarrollo de la actividad.

El juego también fue aplicado para explicar la emocionalidad que los datos transmiten por un grupo que primero expuso y dio evidencias a través de una comparación de gráficos y videos sobre temas específicos, mostrando cómo la manera de comunicar la información, seleccionando estilos, gráficos, movimientos, etc. da cuenta del mensaje que se desea indicar, aún con la misma información base. Luego de la exposición trabajaron con un juego muy competitivo usando la plataforma kahoot.com.



Figura 4: Instancia de preguntas de la temática utilizando Kahoot  
Fuente: registro propio durante el desarrollo de la actividad.

La idea de la captura de datos de clasificación, tales como género o raza, por ejemplo, fue desarrollada por un grupo con una modalidad de debate. La instancia de puesta en común partió de referencias actuales acerca de la captación de datos al momento de registro en aplicaciones y redes sociales. Luego se consideraron las adaptaciones (en algunos casos, simplemente cosméticas) para tratar de abordar la captura de datos de género de manera no binaria. Asimismo, se incluyeron análisis sobre las consecuencias macro y micro de obtener los datos con detalles en cuanto a la clasificación. El debate fue sumamente participativo, donde el resto del curso sumó ejemplos y sus perspectivas, considerando en cada caso pros y contras de cada enfoque. Y como punto de mayor nivel en el debate, se dieron situaciones en las cuales no se tomaba una posición como única válida, se admitía que había elementos valiosos desde las diferentes perspectivas compartidas. El trabajo de la clase mencionada tuvo una extensión algo mayor, al tiempo de resultar la de mayor participación e interés por el grupo completo de alumnos.

Otro elemento de la experiencia que resulta valioso reportar tiene que ver con los materiales para registrar las producciones, así como para captar la valoración de los estudiantes del devenir de la actividad. En cada presentación grupal se tomaron fotos, se completó una grilla de observación semi estructurada con una serie de aspectos de interés, y se subieron elementos significativos en una cuenta en la red social Instagram. La cuenta era cerrada para los estudiantes y la docente, y se incluyeron los materiales producidos, las fotografías, así como algunos comentarios espontáneos que surgieron. El instrumento para relevar la evaluación de la experiencia fue un formulario tipo encuesta, autoadministrado que facilitó la sistematización de las opiniones de los estudiantes.

### **Conclusiones y reflexiones**

Siguiendo a Ochoa (2003), además del proyecto, se enfatiza en la experiencia de los alumnos, pero también en las vivencias como docente en el transcurso de la experiencia.

El desarrollo de la actividad en primer lugar aportó a la interacción áulica elementos distendidos, de intercambio y de acercamiento docente-alumnos en una temática relevante en la generación de ideas. Permitió fomentar la escucha de opiniones, así como un debate con ideas diferenciadas y fundamentadas, en un clima de respeto y valoración del disenso. El aprendizaje como docente fue notorio, especialmente en la manera de introducir un tema opinable y discutible en el marco de materias más técnicas, como también en cuanto al poder conocer a los estudiantes desde sus aportes creativos, sus convicciones, sus opiniones, sus formas de expresarse.

Entre los principales resultados observados se destacan:

- Interacción de los alumnos en temáticas relacionadas con el género, la inclusión y las diferencias.
- Posibilidad de expresarse a través de diferentes medios y soportes, manifestando una notoria capacidad de innovación y creatividad con respecto a los modelos de presentación de clases tradicionales y los modos encarados por los grupos que presentaron previamente.
- Incorporación de dinámicas como videos, juegos, entre otros elementos de notoria variedad entre grupos.
- Posicionamiento ante los diferentes conceptos con ánimo de justificar y adaptar a la propia realidad que observar a nivel personal y nacional.
- Compromiso con la tarea y con las consignas.

Algunos factores que se consideraron para futuras experiencias, se vinculan con las tensiones entre lo proyectado y el proceso. En este sentido se destaca que los tiempos previstos de dedicación de los estudiantes, resultaron extensos. Debían leer y comprender un capítulo del texto, idear una manera de compartir y presentar, y finalmente producir los materiales, contenidos, etc. La contraparte positiva y valorada fue el cumplimiento en tiempo y forma de las entregas y presentaciones de todos los grupos.

Resultó valioso haber realizado un diseño pormenorizado de la actividad y las consignas, y la experiencia pudo ser documentada gracias al registro intencionado de cada elemento de la experiencia implementada.

La evaluación por parte de los participantes de la experiencia realizada es el punto de cierre de valor de la actividad. En este caso, a partir de una encuesta semi estructurada, los participantes evaluaron la eficacia de la actividad para incorporar la temática y propusieron elementos a optimizar para futuras ediciones. El resultado fue positivo de manera generalizada y se obtuvieron datos de valor para la retroalimentación de la experiencia.

### Referencias bibliográficas

- Álvarez Castillo, J. L., García-Cano Torrico, M. (2022). Diversidad e inclusión en la universidad: La vía de la institucionalización. España: Narcea Ediciones.
- CEPAL (2017) Brechas, ejes y desafíos en el vínculo entre lo social y lo productivo. Naciones Unidas. Santiago.
- D'Ignazio, C.; Klein (2020). Data Feminism. MIT Press (Traducción al español en: <https://data-feminism.mitpress.mit.edu/pub/v874jd7x/release/1>)
- D'Ignazio, C.; Klein, L., Livio, M., Niederer, S., & Colombo, G. (2021). La práctica del feminismo de datos: Conversaciones con Catherine D'Ignazio, Lauren Klein y Maya Livio. Diseña, (19), Interview.1. <https://doi.org/10.7764/disena.19.Interview.1>
- Degli Esposti, S. (2021) El rol del análisis de género en la reducción de los sesgos algorítmicos. Información Comercial Española, Revista de Economía 921: 127-143. Recuperado de: <https://digital.csic.es/handle/10261/306692>
- José Manuel Juárez Núñez, Sonia Comboni Salinas (2020) Interculturalidad y diversidad en la educación: concepciones, políticas y prácticas. México: Universidad Autónoma Metropolitana, Unidad Xochimilco, División de Ciencias Sociales y Humanidades.
- Mato, (2018) Educación superior, diversidad cultural e interculturalidad en América Latina. Conferencia Regional de Educación Superior. Córdoba: UNC.
- Ochoa, L. (2003). Módulo 2: La documentación narrativa de experiencias escolares. En Manual de capacitación sobre registro y sistematización de experiencias pedagógicas, Ministerio de Educación, Ciencia y Tecnología.

Seligman, M. (2011). *Florecer. La nueva psicología positiva y la búsqueda de bienestar* (Flourish: A Visionary New Understanding of Happiness and Well-being). New York: Free Press, ISBN 978-1-4391-9075-3.

UNESCO (2020) *Informe de seguimiento de la educación en el mundo: Inclusión y educación*. (2020). Paris (Francia): UNESCO Publishing.

## Acercamiento a la programación visual. Una experiencia virtual con docentes de la región patagónica

Edith Lovos, Martín Goin  
{elovos,mgoin}@unrn.edu.ar

CIEDIS  
Universidad Nacional de Río Negro

### Resumen

En este trabajo se presentan y discuten algunos resultados alcanzados en dos implementaciones de una propuesta pedagógica en formato taller y modalidad virtual, que tiene por objetivo acercar a las y los docentes de nivel medio de la región geográfica donde se inserta la Universidad Nacional de Río Negro a la programación visual usando bloques. Si bien, ambas implementaciones tuvieron un índice de abandono alto, las y los docentes que completaron el trayecto propuesto, lograron apropiarse de los conocimientos y valoraron la propuesta positivamente. Así mismo, se conformó un espacio de encuentro entre docentes de nivel medio y superior que permitió reflexionar y discutir sobre estrategias didácticas, así como también sobre otras tecnologías digitales que puedan ser un aporte a mejorar las prácticas de enseñanza. A futuro, se espera avanzar en una propuesta de extensión universitaria destinada a docentes y estudiantes de los últimos años de nivel primario y ciclo básico de nivel medio, enfocada en la programación desconectada a través del uso de un juego de mesa de producción propia.

**Palabras Claves:** programación visual, formación docente, nivel medio

### Introducción

En la vida cotidiana de las y los ciudadanos en general, las tecnologías de la información y la comunicación (TIC) adquieren cada vez más presencia. Así la posibilidad de acceso a determinados dispositivos tecnológicos (celulares inteligentes por ejemplo) y servicios (internet principalmente), pueden ser un paso en el camino hacia la democratización del acceso, que necesariamente debe ser acompañado de habilidades y saberes que posibiliten a las y los ciudadanos comprender la tecnología, reconocer situaciones problemáticas que puedan solucionarse/mejorarse a través de las mismas y poder avanzar en ello de manera soberana (Llambi et al., 2023). En Argentina, a través de diferentes políticas públicas (CFE Res. N° 263/15, CFE 423/22, PLANIED) e iniciativas como la que lleva adelante la Fundación Sadosky a través de Program.AR, se busca trabajar en la formación docente en Ciencias de la

Computación en general y en programación en particular. Por otra parte, en la provincia de Río Negro, donde se ubica la Universidad Nacional de Río Negro (UNRN), desde el año 2017, se implementó para el universo de nivel medio, una nueva forma de trabajo en el turno diurno, conocida como escuela secundaria rionegrina (ESRN). Esta política educativa como señalan Tobeña y Nobile (2021), modificó: la estructura curricular, la organización institucional y el espacio de trabajo docente. Así, la ESRN propone designar a los profesores por cargo buscando concentrar su trabajo y promoviendo el sentido de pertenencia con la institución, así un docente tiene como máximo 1 cargo por turno (en 1 o 2 escuelas máximo). El diseño curricular propuesto, apunta al trabajo por áreas de conocimiento integradas por unidades curriculares, donde hay un ciclo básico (de 1º a 2º año) y un ciclo orientado (de 3º a 5º año), que demanda poner en práctica habilidades de trabajo disciplinar, interdisciplinar y colaborativo, como también la evaluación conjunta por parte de las y los docentes. En relación al proceso de selección de las orientaciones de las escuelas, las mismas se llevaron adelante con la participación de la comunidad educativa, y de los 93 establecimientos educativos que hay en la provincia, 14 seleccionaron la orientación informática.<sup>1</sup> Sin embargo, la implementación de esta política, como señalan Tobeña y Nobile (op.cit) muestra una vacancia de perfiles docentes afines a la propuesta, así como también la necesidad de cubrir cargos con profesores idóneos y avanzar en el diseño e implementación de diferentes propuestas de formación inicial y continua.

En este trabajo, se presenta y discute una propuesta pedagógica destinada a la formación docente, que busca ser un aporte para el aprendizaje de la programación siguiendo el formato taller, y promoviendo un espacio de encuentro para pensar y discutir sobre estrategias didácticas y otras tecnologías (juegos serios, aplicaciones móviles entre otras) que permitan su inclusión e integración en las actividades áulicas.

### **Descripción de la experiencia**

La experiencia que se presenta a continuación se trata de curso de programación, al que denominaremos de aquí en adelante como Taller Scratch (TS), surge a partir de una convocatoria para la presentación de propuestas de formación docente en modalidad virtual destinadas a los docentes de nivel medio y, enmarcadas en el Programa Nacional de Formación Permanente “Nuestra Escuela” que el Ministerio de Educación lleva adelante a través del Instituto Nacional de Formación Docente (INFoD). La convocatoria estableció entre otros objetivos pedagógicos, que las propuestas deberían ajustarse a los Núcleos de Aprendizaje Prioritarios vigentes según nivel y modalidad, y abordar las temáticas que

---

<sup>1</sup> <https://educacion.rionegro.gov.ar/nota/4871>

establece el Plan Anual Educativo 2022 (Res.Ministerial N° 840/22). Asimismo, se estableció en 60 el número de participantes por aulas destinadas a la implementación de las propuestas pedagógicas.

Así el TS, formó parte de las 12 propuestas presentadas por la Universidad Nacional de Río Negro (UNRN) y tuvo como destinatarios a las y los docentes, docentes en formación, referentes TIC y directivos de nivel secundario y primario, interesados en acercarse a la programación y que habitan en la provincia de Río Negro. Se optó por un formato que propicia el enfoque de “aprender haciendo” y pudiera ser seguido de acuerdo a los tiempos de las y los docentes participantes. Asimismo, y a partir de los aportes encontrados en experiencias formativas realizadas en el país (Dapozo et al., 2017, Monjelat et al., 2019, Fusero y Chiarani, 2018) y otras recientes en el ámbito de la UNRN (Goin et al., 2021) se optó por utilizar el lenguaje de programación Scratch.

Por otra parte, el TS se implementó en dos instancias, una de Noviembre 2022 a Mayo 2023 (con un receso a partir del 17 de Diciembre del 2022 y hasta el 15 de Marzo del 2023) y otra de Abril a Junio de 2023, con una duración de 12 semanas. La propuesta estuvo a cargo de dos docentes-investigadores de la UNRN, vinculados a carreras de ciencia y tecnología de las Sedes Andina y Atlántica, a la vez que desarrollan investigación sobre tecnologías aplicadas en Educación. Asimismo, el equipo docente viene trabajando en la promoción de actividades lúdicas para el desarrollo de habilidades de razonamiento algorítmico (Lovos et al., 2021).

La convocatoria e inscripción, se llevó a través de los canales de comunicación del INFoD, siendo para la primera implementación del TS, desde el 30 de septiembre al 06 de Octubre de 2022, y para la segunda del 20 de febrero al 13 de marzo de 2023.

En cuanto a los contenidos del TS, los mismos se estructuraron en 4 unidades temáticas: Acercamiento a la Programación, Introducción al mundo Scratch, Estructuras de control y por último Juegos y animaciones. En la primera unidad se abordaron conceptos como: Informática, Computación, Datos e Información, Algoritmos y Programas, y Enfoques para la enseñanza, tomando como base los recursos didácticos disponibles en Program.ar.

El TS se configuró para ser trabajado durante 12 semanas, dónde cada unidad temática estuvo compuesta por: Vídeos de presentación de contenido, Materiales de lectura, Actividad Práctica con Scratch (AP) y Actividad de auto-evaluación conceptual (cuestionario). En la figura 1 se muestra una captura de pantalla para el caso de la unidad 1.

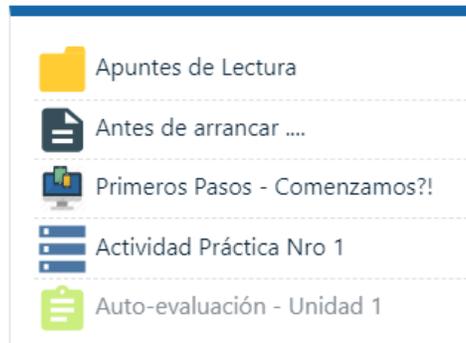


Figura 1. Captura de pantalla disposición contenidos por unidad

Sobre la forma de trabajo, el TS propone a las y los docentes avanzar a su ritmo, teniendo en cuenta el recorrido propuesto, donde las actividades didácticas están articuladas, de manera tal que para poder realizar las AP de cada unidad, primero es necesario haber asistido a los vídeos explicativos y en forma opcional al material de lectura. En el caso de la autoevaluación, para poder realizarla es necesario haber entregado la actividad práctica (AP). Asimismo, al comienzo de la Unidad 1, se aplicó una encuesta diagnóstica que permitió recuperar información demográfica y conocimientos previos.

Aunque el TS se compone mayormente de actividades asincrónicas, se estableció un espacio de encuentro sincrónico de participación opcional (a través de la plataforma Zoom), que se fijó a partir de las 19hs los días martes. Durante estos encuentros, se trabajan consultas realizadas en situ o las que se realizan a través del correo o foros del aula virtual, asimismo se presentan y discuten las producciones realizadas en las actividades prácticas.

Para la acreditación del curso, las y los participantes debían desarrollar y entregar todas las actividades prácticas propuestas y responder a los cuestionarios de autoevaluación de cada unidad temática. En la tabla 1, se presentan los requerimientos de las actividades requeridas en cada unidad, los mismos se han establecido a partir de experiencias previas (Monjelat et al., 2019).

Actividad Práctica	Requerimientos
Dialogo - Presentación	Duración (entre 1 minuto y 1 minuto y medio). El fondo debe ser un paisaje representativo del lugar de residencia Incluir una pequeña animación en los personajes, por ejemplo el pestañeo.

	En el dialogo se deben visualizar las viñetas, así como el audio (texto a voz)
Manejo de las Estructuras de Control	3 personajes del mundo animal, que deben moverse desde un borde a otro (uno por vez), realizando los movimientos asociados a su especie (ej. ave vuela, pez nada, etc.) Duración total : entre 1 y 3 minutos Agregar sonido a cada animal El fondo debe cambiar cada vez que aparece un animal Indicar los créditos con cambio de fondo y agregando sonidos propios del Scratch
Juegos y animaciones	Duración 3 minutos mínimo Control: mínimo 4 herramientas distintas Sensores: mínimo 2 herramientas distintas Movimiento: mínimo 4 herramientas distintas Apariencia: mínimo 4 herramientas distintas Eventos: mínimo 4 herramientas distintas Sonido: mínimo 1 herramienta Operadores: mínimo 4 herramientas distintas Variables: mínimo una variable y una herramienta Algún bloque de extensión: música, lápiz, texto a voz o traducir.

Tabla 1. Requerimientos para las actividades prácticas

En relación a las herramientas, se utilizó el campus bimodal de la UNRN (<https://campusbimodal.unrn.edu.ar/>) y plataforma Scratch. Específicamente para la entrega de las AP se utilizó la funcionalidad base de datos provista por el campus, de esta manera las producciones de las y los participantes quedaron accesibles a toda la comisión. Y para la actividad de cierre, se utilizó la plataforma *flip*<sup>2</sup>, a través de la cual se les invitó a elaborar un resumen (90 segundos de vídeo) de lo más significativo que han aprendido en el curso y cómo lo podrían utilizar en su práctica docente.

Por último, es importante destacar, que la convocatoria, establecía que la participación de los docentes, sería valorada con puntaje por la Junta de Clasificación Secundaria, según el artículo 8 de la Resolución 642/17 del Ministerio de Educación de la provincia de Río Negro

## Resultados

A continuación se presentan en la Tabla 2, algunos datos sobre cantidad de docentes inscriptos, cantidad de docentes que iniciaron el curso y finalmente cantidad de participantes que lograron finalizar y acreditar.

<sup>2</sup> <https://flip.com/>

TS -Implementación	Aulas	Inscriptos	Iniciaron	Participaron	Acreditaron
Primera	4	241	108	31	18
Segunda	2	113	74	10	6

Tabla 2. Datos de la implementación del Taller Scratch

Si tenemos en cuenta, el periodo de inscripción, los números dan cuenta del interés por la propuesta, aun cuando en la primera implementación del TS, ésta coincidía con el final del año escolar, donde se intensifica la actividad laboral de las y los docentes (cierre de notas, mesas de exámenes, actos escolares, etc.).

A continuación, se presentan algunos datos recuperados de la encuesta diagnóstica a la que respondieron 134 docentes de los 182 que iniciaron el TS en sus dos implementaciones. En relación a la región geográfica a la cual pertenecen, en la Gráfico 1, se observa qué provienen mayoritariamente de la zona Alto Valle y Valle Medio de la provincia de Río Negro. Según datos oficiales<sup>3</sup>, en esa región, se concentra la oferta de nivel medio ESRN con la orientación informática.

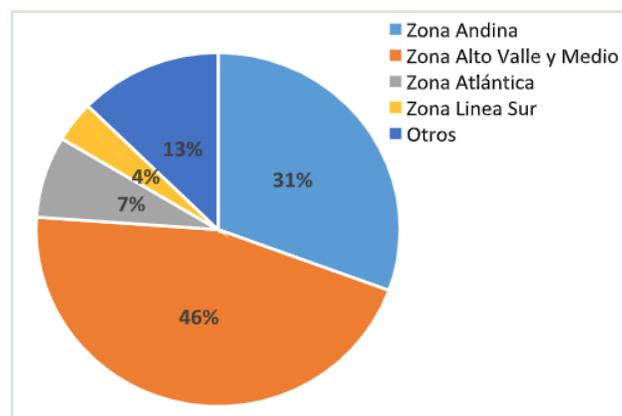


Gráfico 1. Participantes por zona geográfica

El 65% del grupo participante, son mujeres con predominio en el rango etario de 36 y 45 años. Del total del grupo, el 43% indicó tener conocimientos previos de programación, así el grupo se conforma por participantes mayormente noveles en relación a los temas que aborda el TS. Consultados respecto a si en la/s escuela/s donde llevan adelante la práctica docente se

---

<sup>3</sup>

<https://rionegro.gov.ar/articulo/18015/la-escuela-secundaria-de-rio-negro-eligio-sus-orientaciones-en-cada-lo-calidad-de-la-provincia>

trabajan contenidos vinculados a la programación, el 50,75% indicó que no, del resto, el 14,18% indico no saberlo y el 35,07% que sí. En el Gráfico 2 se presenta la distribución del grupo en relación al nivel educativo donde llevan adelante la práctica docente.

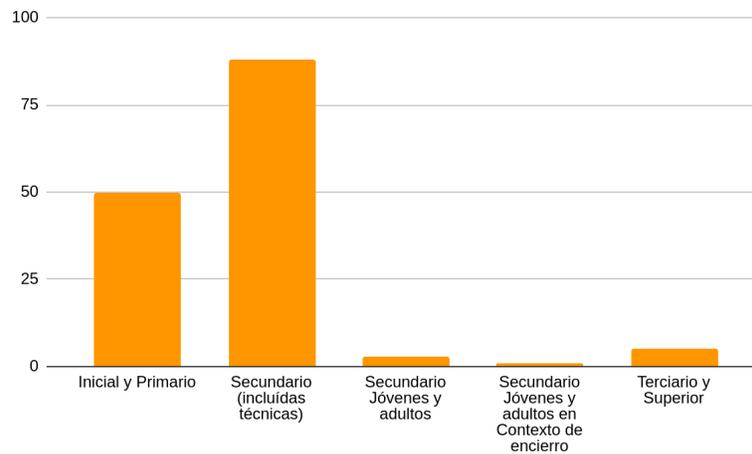


Gráfico 2. Participantes por nivel educativo de su práctica profesional

Como parte del cierre del TS, se solicitó a las y los docentes compartir a través de la plataforma *flip*, su valoración respecto de lo aprendido y las posibilidades de incluirlo en el aula. Un factor común de las expresiones docentes, es lo divertido que les resultó avanzar con las consignas, aun cuando fueran complejas, y los conceptos y herramientas fueran novedosas. Por otra parte, la forma de organización del aula y en particular el paso a paso propuesto en los vídeos fueron valorados positivamente. Asimismo, rescataron el valor de los encuentros sincrónicos, como espacio que les permitió encontrar respuestas (y otras preguntas) e intercambiar conocimientos con otros docentes, así como también para formar equipos de trabajo. Estos encuentros se grababan y quedaban disponibles a todos los participantes. Sobre la posibilidad de llevar lo aprendido al aula, para quienes el TS fue su primer acercamiento a la programación, coincidieron en la necesidad de profundizar en el uso de la plataforma y de invitar/recomendarla a otros colegas.

Sobre los factores que les dificultaron el recorrido propuesto, indicaron principalmente: la sobrecarga en su actividad laboral, la demanda de habilidades para el aprendizaje autónomo que se pusieron en juego en el TS y en el caso de la primera implementación el retomar el trabajo luego del receso. En cuanto al recurso base de datos, que se utilizó para la entrega de las actividades prácticas, aunque su uso no resultó intuitivo, el mismo permitió poner a

disposición del grupo las producciones desarrolladas, comentarlas y reconocer otras formas de resolver problemas similares.

Sobre los cuestionarios, las preguntas con mejores resultados son aquellas referidas a la funcionalidad de los bloques. Siendo las que presentaron mayor dificultad, aquellas que implicaban realizar el seguimiento de un algoritmo, como las que se presentan en las Figuras 2 y 3.

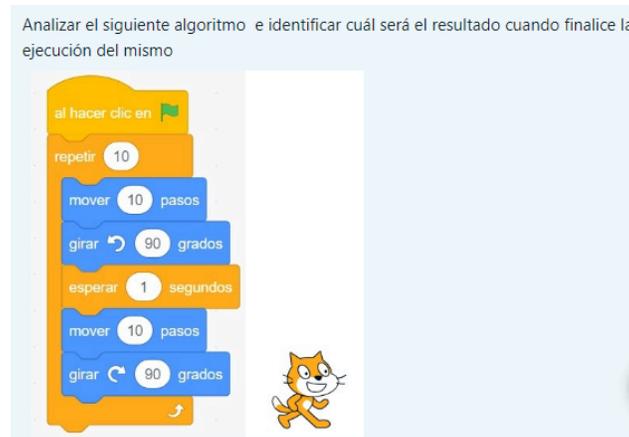


Figura 2. Pregunta Autoevaluación Unidad 2 - Repetición



Figura 3. Pregunta Autoevaluación Unidad 3 – Estructuras de control, variables y operaciones

A partir de estos resultados, aun cuando ambas implementaciones del TS, hayan tenido un índice de abandono alto, las y los docentes que llegaron a completar el trayecto propuesto, han logrado apropiarse los conocimientos y aplicarlos en producciones que cumplen con los objetivos establecidos. Respecto al desarrollo de la última actividad práctica, que proponía avanzar en la implementación de un juego o animación, permitió abordar temas como el diseño de juegos educativos, el potencial del recurso para la autoevaluación entre otros. Así también la complejidad de la tarea, propicio la conformación de grupos de trabajo entre docentes de diferentes disciplinas y residencia, que no estuvo exento de dificultades pero que entendemos pueden ser un paso para animarse a otras prácticas educativas

### **Conclusiones**

La enseñanza de programación y/o la robótica, permiten acercar las CC a las aulas de nivel medio, sin embargo el desafío de trabajarlas por áreas curriculares integradas siguiendo la propuesta de la ESRN, requiere que las y los docentes se sientan motivados, acompañados y empoderados a incluirla en sus prácticas pedagógicas, desde una postura crítica y reflexiva. En este sentido, construir lazos con instituciones y organizaciones vinculadas al tema, que permitan acercarse a diferentes enfoques de enseñanza y aprendizaje (así como también recursos) puede ser un aporte para atender a la diversidad que presentan los diferentes espacios de práctica docente.

A futuro, se espera avanzar en la implementación de una propuesta de extensión universitaria, recientemente aprobada en la convocatoria “Universidad, Cultura y territorio” de la Secretaria de Políticas Universitarias (SPU), y destinada a docentes y estudiantes de los últimos años de nivel primario y ciclo básico de nivel medio, enfocada en la programación desconectada a través del uso de un juego de mesa de producción propia denominado JAM (Juego Algorítmico de mesa).

### **Referencias**

- Dapozo, G. N., Greiner, C. L., & Petris, R. H. (2017). Programación en las escuelas. Experiencia de formación docente en el Nordeste Argentino. *Memorias del Encuentro Internacional de Educación a Distancia*, 5(5).
- Fussero, G. B., & Chiarani, M. C. (2018). Programación visual en la formación de docentes de Biología. In *Memorias de las Jornadas Nacionales y Congreso Internacional en Enseñanza de la Biología* (Vol. 1, No. Extraordinario, pp. 143-148).
- Goin, M., Lovos, E. N., & Sanz, C. (2021). Aprender a Programar en Tiempos de Pandemia. Una experiencia con Docentes de Nivel Primario.

- Lovos, E. N., Goin, M. M. J., & Gibelli, T. (2021). Juegos analógicos para el aprendizaje de algoritmos en el Nivel Primario.
- Llambi, C., Borchardt, M., Klinkovich, V., Locca, N., Martínez, C., & Scasso, M. (2023). Aprendizajes y desafíos para la enseñanza de las Ciencias de la Computación en las escuelas.
- Monjelat, N. (2019). Enseñanza de la programación en la formación docente: pautas pedagógicas desde una experiencia con maestros de primaria. In I Simposio Argentino de Educación en Informática (SAEI 2019)-JAIIO 48 (Salta).
- PLANIED: Plan Integral de Educación Digital del Ministerio de Educación y Deportes de la Nación Argentina (s.f).
- Tobeña, V., & Nobile, M. (2021). ¿Hacia dónde va el trabajo docente en secundaria? Análisis de su reconfiguración en dos políticas provinciales recientes. *Revista de Educación*, (22), 43-66.
- Consejo Federal de Educación Res. 263/15. Programación y robótica. Disponible en: [https://cfe.educacion.gob.ar/resoluciones/res15/263-15\\_01.pdf](https://cfe.educacion.gob.ar/resoluciones/res15/263-15_01.pdf)
- Consejo Federal de Educación Res. 423/22. Disponible en: [https://www.argentina.gob.ar/sites/default/files/2022/09/rcfe\\_432-22.pdf](https://www.argentina.gob.ar/sites/default/files/2022/09/rcfe_432-22.pdf)
- Consejo Provincial de Educación Resolución 642/2017. Disponible en: [https://educacion.rionegro.gov.ar/files/Direcciones\\_2020/Superior/00642-17%20declaraci%C3%B3n%20de%20inter%C3%A9s.pdf](https://educacion.rionegro.gov.ar/files/Direcciones_2020/Superior/00642-17%20declaraci%C3%B3n%20de%20inter%C3%A9s.pdf)

## Acercamiento a la programación visual. Una experiencia virtual con docentes de la región patagónica

Edith Lovos, Martín Goin  
{elovos,mgoin}@unrn.edu.ar

CIEDIS  
Universidad Nacional de Río Negro

### Resumen

En este trabajo se presentan y discuten algunos resultados alcanzados en dos implementaciones de una propuesta pedagógica en formato taller y modalidad virtual, que tiene por objetivo acercar a las y los docentes de nivel medio de la región geográfica donde se inserta la Universidad Nacional de Río Negro a la programación visual usando bloques. Si bien, ambas implementaciones tuvieron un índice de abandono alto, las y los docentes que completaron el trayecto propuesto, lograron apropiarse de los conocimientos y valoraron la propuesta positivamente. Así mismo, se conformó un espacio de encuentro entre docentes de nivel medio y superior que permitió reflexionar y discutir sobre estrategias didácticas, así como también sobre otras tecnologías digitales que puedan ser un aporte a mejorar las prácticas de enseñanza. A futuro, se espera avanzar en una propuesta de extensión universitaria destinada a docentes y estudiantes de los últimos años de nivel primario y ciclo básico de nivel medio, enfocada en la programación desconectada a través del uso de un juego de mesa de producción propia.

**Palabras Claves:** programación visual, formación docente, nivel medio

### Introducción

En la vida cotidiana de las y los ciudadanos en general, las tecnologías de la información y la comunicación (TIC) adquieren cada vez más presencia. Así la posibilidad de acceso a determinados dispositivos tecnológicos (celulares inteligentes por ejemplo) y servicios (internet principalmente), pueden ser un paso en el camino hacia la democratización del acceso, que necesariamente debe ser acompañado de habilidades y saberes que posibiliten a las y los ciudadanos comprender la tecnología, reconocer situaciones problemáticas que puedan solucionarse/mejorarse a través de las mismas y poder avanzar en ello de manera soberana (Llambi et al., 2023). En Argentina, a través de diferentes políticas públicas (CFE Res. N° 263/15, CFE 423/22, PLANIED) e iniciativas como la que lleva adelante la Fundación Sadosky a través de Program.AR, se busca trabajar en la formación docente en Ciencias de la

Computación en general y en programación en particular. Por otra parte, en la provincia de Río Negro, donde se ubica la Universidad Nacional de Río Negro (UNRN), desde el año 2017, se implementó para el universo de nivel medio, una nueva forma de trabajo en el turno diurno, conocida como escuela secundaria rionegrina (ESRN). Esta política educativa como señalan Tobeña y Nobile (2021), modificó: la estructura curricular, la organización institucional y el espacio de trabajo docente. Así, la ESRN propone designar a los profesores por cargo buscando concentrar su trabajo y promoviendo el sentido de pertenencia con la institución, así un docente tiene como máximo 1 cargo por turno (en 1 o 2 escuelas máximo). El diseño curricular propuesto, apunta al trabajo por áreas de conocimiento integradas por unidades curriculares, donde hay un ciclo básico (de 1º a 2º año) y un ciclo orientado (de 3º a 5º año), que demanda poner en práctica habilidades de trabajo disciplinar, interdisciplinar y colaborativo, como también la evaluación conjunta por parte de las y los docentes. En relación al proceso de selección de las orientaciones de las escuelas, las mismas se llevaron adelante con la participación de la comunidad educativa, y de los 93 establecimientos educativos que hay en la provincia, 14 seleccionaron la orientación informática.<sup>1</sup> Sin embargo, la implementación de esta política, como señalan Tobeña y Nobile (op.cit) muestra una vacancia de perfiles docentes afines a la propuesta, así como también la necesidad de cubrir cargos con profesores idóneos y avanzar en el diseño e implementación de diferentes propuestas de formación inicial y continua.

En este trabajo, se presenta y discute una propuesta pedagógica destinada a la formación docente, que busca ser un aporte para el aprendizaje de la programación siguiendo el formato taller, y promoviendo un espacio de encuentro para pensar y discutir sobre estrategias didácticas y otras tecnologías (juegos serios, aplicaciones móviles entre otras) que permitan su inclusión e integración en las actividades áulicas.

### **Descripción de la experiencia**

La experiencia que se presenta a continuación se trata de curso de programación, al que denominaremos de aquí en adelante como Taller Scratch (TS), surge a partir de una convocatoria para la presentación de propuestas de formación docente en modalidad virtual destinadas a los docentes de nivel medio y, enmarcadas en el Programa Nacional de Formación Permanente “Nuestra Escuela” que el Ministerio de Educación lleva adelante a través del Instituto Nacional de Formación Docente (INFoD). La convocatoria estableció entre otros objetivos pedagógicos, que las propuestas deberían ajustarse a los Núcleos de Aprendizaje Prioritarios vigentes según nivel y modalidad, y abordar las temáticas que

---

<sup>1</sup> <https://educacion.rionegro.gov.ar/nota/4871>

establece el Plan Anual Educativo 2022 (Res.Ministerial N° 840/22). Asimismo, se estableció en 60 el número de participantes por aulas destinadas a la implementación de las propuestas pedagógicas.

Así el TS, formó parte de las 12 propuestas presentadas por la Universidad Nacional de Río Negro (UNRN) y tuvo como destinatarios a las y los docentes, docentes en formación, referentes TIC y directivos de nivel secundario y primario, interesados en acercarse a la programación y que habitan en la provincia de Río Negro. Se optó por un formato que propicia el enfoque de “aprender haciendo” y pudiera ser seguido de acuerdo a los tiempos de las y los docentes participantes. Asimismo, y a partir de los aportes encontrados en experiencias formativas realizadas en el país (Dapozo et al., 2017, Monjelat et al., 2019, Fusero y Chiarani, 2018) y otras recientes en el ámbito de la UNRN (Goin et al., 2021) se optó por utilizar el lenguaje de programación Scratch.

Por otra parte, el TS se implementó en dos instancias, una de Noviembre 2022 a Mayo 2023 (con un receso a partir del 17 de Diciembre del 2022 y hasta el 15 de Marzo del 2023) y otra de Abril a Junio de 2023, con una duración de 12 semanas. La propuesta estuvo a cargo de dos docentes-investigadores de la UNRN, vinculados a carreras de ciencia y tecnología de las Sedes Andina y Atlántica, a la vez que desarrollan investigación sobre tecnologías aplicadas en Educación. Asimismo, el equipo docente viene trabajando en la promoción de actividades lúdicas para el desarrollo de habilidades de razonamiento algorítmico (Lovos et al., 2021).

La convocatoria e inscripción, se llevó a través de los canales de comunicación del INFoD, siendo para la primera implementación del TS, desde el 30 de septiembre al 06 de Octubre de 2022, y para la segunda del 20 de febrero al 13 de marzo de 2023.

En cuanto a los contenidos del TS, los mismos se estructuraron en 4 unidades temáticas: Acercamiento a la Programación, Introducción al mundo Scratch, Estructuras de control y por último Juegos y animaciones. En la primera unidad se abordaron conceptos como: Informática, Computación, Datos e Información, Algoritmos y Programas, y Enfoques para la enseñanza, tomando como base los recursos didácticos disponibles en Program.ar.

El TS se configuró para ser trabajado durante 12 semanas, dónde cada unidad temática estuvo compuesta por: Vídeos de presentación de contenido, Materiales de lectura, Actividad Práctica con Scratch (AP) y Actividad de auto-evaluación conceptual (cuestionario). En la figura 1 se muestra una captura de pantalla para el caso de la unidad 1.

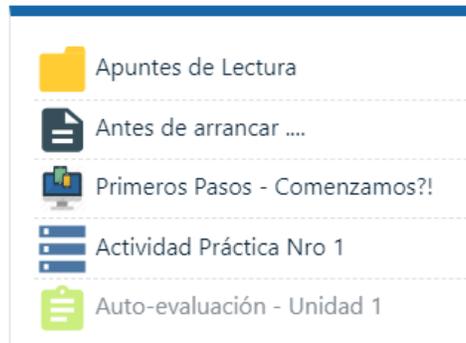


Figura 1. Captura de pantalla disposición contenidos por unidad

Sobre la forma de trabajo, el TS propone a las y los docentes avanzar a su ritmo, teniendo en cuenta el recorrido propuesto, donde las actividades didácticas están articuladas, de manera tal que para poder realizar las AP de cada unidad, primero es necesario haber asistido a los vídeos explicativos y en forma opcional al material de lectura. En el caso de la autoevaluación, para poder realizarla es necesario haber entregado la actividad práctica (AP). Asimismo, al comienzo de la Unidad 1, se aplicó una encuesta diagnóstica que permitió recuperar información demográfica y conocimientos previos.

Aunque el TS se compone mayormente de actividades asincrónicas, se estableció un espacio de encuentro sincrónico de participación opcional (a través de la plataforma Zoom), que se fijó a partir de las 19hs los días martes. Durante estos encuentros, se trabajan consultas realizadas en situ o las que se realizan a través del correo o foros del aula virtual, asimismo se presentan y discuten las producciones realizadas en las actividades prácticas.

Para la acreditación del curso, las y los participantes debían desarrollar y entregar todas las actividades prácticas propuestas y responder a los cuestionarios de autoevaluación de cada unidad temática. En la tabla 1, se presentan los requerimientos de las actividades requeridas en cada unidad, los mismos se han establecido a partir de experiencias previas (Monjelat et al., 2019).

Actividad Práctica	Requerimientos
Dialogo - Presentación	Duración (entre 1 minuto y 1 minuto y medio). El fondo debe ser un paisaje representativo del lugar de residencia Incluir una pequeña animación en los personajes, por ejemplo el pestañeo.

	En el dialogo se deben visualizar las viñetas, así como el audio (texto a voz)
Manejo de las Estructuras de Control	3 personajes del mundo animal, que deben moverse desde un borde a otro (uno por vez), realizando los movimientos asociados a su especie (ej. ave vuela, pez nada, etc.) Duración total : entre 1 y 3 minutos Agregar sonido a cada animal El fondo debe cambiar cada vez que aparece un animal Indicar los créditos con cambio de fondo y agregando sonidos propios del Scratch
Juegos y animaciones	Duración 3 minutos mínimo Control: mínimo 4 herramientas distintas Sensores: mínimo 2 herramientas distintas Movimiento: mínimo 4 herramientas distintas Apariencia: mínimo 4 herramientas distintas Eventos: mínimo 4 herramientas distintas Sonido: mínimo 1 herramienta Operadores: mínimo 4 herramientas distintas Variables: mínimo una variable y una herramienta Algún bloque de extensión: música, lápiz, texto a voz o traducir.

Tabla 1. Requerimientos para las actividades prácticas

En relación a las herramientas, se utilizó el campus bimodal de la UNRN (<https://campusbimodal.unrn.edu.ar/>) y plataforma Scratch. Específicamente para la entrega de las AP se utilizó la funcionalidad base de datos provista por el campus, de esta manera las producciones de las y los participantes quedaron accesibles a toda la comisión. Y para la actividad de cierre, se utilizó la plataforma *flip*<sup>2</sup>, a través de la cual se les invitó a elaborar un resumen (90 segundos de vídeo) de lo más significativo que han aprendido en el curso y cómo lo podrían utilizar en su práctica docente.

Por último, es importante destacar, que la convocatoria, establecía que la participación de los docentes, sería valorada con puntaje por la Junta de Clasificación Secundaria, según el artículo 8 de la Resolución 642/17 del Ministerio de Educación de la provincia de Río Negro

## Resultados

A continuación se presentan en la Tabla 2, algunos datos sobre cantidad de docentes inscriptos, cantidad de docentes que iniciaron el curso y finalmente cantidad de participantes que lograron finalizar y acreditar.

<sup>2</sup> <https://flip.com/>

TS -Implementación	Aulas	Inscriptos	Iniciaron	Participaron	Acreditaron
Primera	4	241	108	31	18
Segunda	2	113	74	10	6

Tabla 2. Datos de la implementación del Taller Scratch

Si tenemos en cuenta, el periodo de inscripción, los números dan cuenta del interés por la propuesta, aun cuando en la primera implementación del TS, ésta coincidía con el final del año escolar, donde se intensifica la actividad laboral de las y los docentes (cierre de notas, mesas de exámenes, actos escolares, etc.).

A continuación, se presentan algunos datos recuperados de la encuesta diagnóstica a la que respondieron 134 docentes de los 182 que iniciaron el TS en sus dos implementaciones. En relación a la región geográfica a la cual pertenecen, en la Gráfico 1, se observa qué provienen mayoritariamente de la zona Alto Valle y Valle Medio de la provincia de Río Negro. Según datos oficiales<sup>3</sup>, en esa región, se concentra la oferta de nivel medio ESRN con la orientación informática.

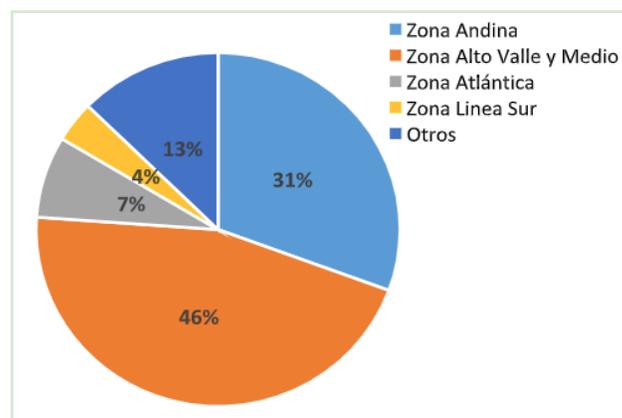


Gráfico 1. Participantes por zona geográfica

El 65% del grupo participante, son mujeres con predominio en el rango etario de 36 y 45 años. Del total del grupo, el 43% indicó tener conocimientos previos de programación, así el grupo se conforma por participantes mayormente noveles en relación a los temas que aborda el TS. Consultados respecto a si en la/s escuela/s donde llevan adelante la práctica docente se

<sup>3</sup>

<https://rionegro.gov.ar/articulo/18015/la-escuela-secundaria-de-rio-negro-eligio-sus-orientaciones-en-cada-lo-calidad-de-la-provincia>

trabajan contenidos vinculados a la programación, el 50,75% indicó que no, del resto, el 14,18% indico no saberlo y el 35,07% que sí. En el Gráfico 2 se presenta la distribución del grupo en relación al nivel educativo donde llevan adelante la práctica docente.

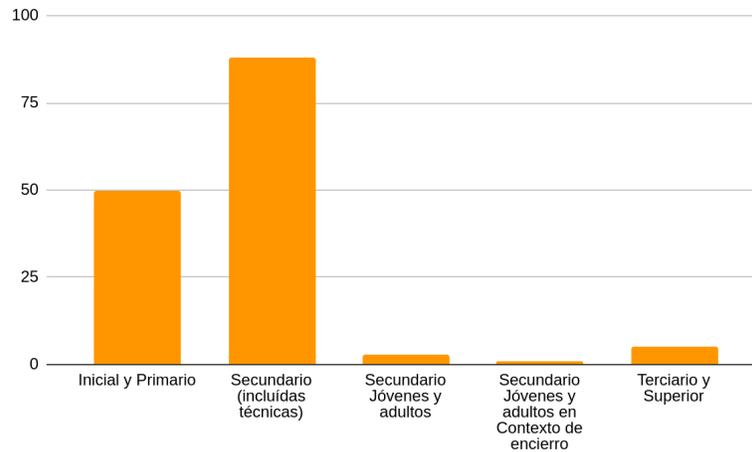


Gráfico 2. Participantes por nivel educativo de su práctica profesional

Como parte del cierre del TS, se solicitó a las y los docentes compartir a través de la plataforma *flip*, su valoración respecto de lo aprendido y las posibilidades de incluirlo en el aula. Un factor común de las expresiones docentes, es lo divertido que les resultó avanzar con las consignas, aun cuando fueran complejas, y los conceptos y herramientas fueran novedosas. Por otra parte, la forma de organización del aula y en particular el paso a paso propuesto en los vídeos fueron valorados positivamente. Asimismo, rescataron el valor de los encuentros sincrónicos, como espacio que les permitió encontrar respuestas (y otras preguntas) e intercambiar conocimientos con otros docentes, así como también para formar equipos de trabajo. Estos encuentros se grababan y quedaban disponibles a todos los participantes. Sobre la posibilidad de llevar lo aprendido al aula, para quienes el TS fue su primer acercamiento a la programación, coincidieron en la necesidad de profundizar en el uso de la plataforma y de invitar/recomendarla a otros colegas.

Sobre los factores que les dificultaron el recorrido propuesto, indicaron principalmente: la sobrecarga en su actividad laboral, la demanda de habilidades para el aprendizaje autónomo que se pusieron en juego en el TS y en el caso de la primera implementación el retomar el trabajo luego del receso. En cuanto al recurso base de datos, que se utilizó para la entrega de las actividades prácticas, aunque su uso no resultó intuitivo, el mismo permitió poner a

disposición del grupo las producciones desarrolladas, comentarlas y reconocer otras formas de resolver problemas similares.

Sobre los cuestionarios, las preguntas con mejores resultados son aquellas referidas a la funcionalidad de los bloques. Siendo las que presentaron mayor dificultad, aquellas que implicaban realizar el seguimiento de un algoritmo, como las que se presentan en las Figuras 2 y 3.

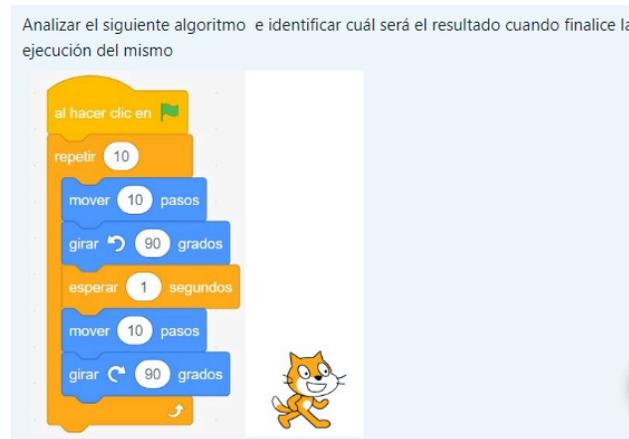


Figura 2. Pregunta Autoevaluación Unidad 2 - Repetición



Figura 3. Pregunta Autoevaluación Unidad 3 – Estructuras de control, variables y operaciones

A partir de estos resultados, aun cuando ambas implementaciones del TS, hayan tenido un índice de abandono alto, las y los docentes que llegaron a completar el trayecto propuesto, han logrado apropiarse los conocimientos y aplicarlos en producciones que cumplen con los objetivos establecidos. Respecto al desarrollo de la última actividad práctica, que proponía avanzar en la implementación de un juego o animación, permitió abordar temas como el diseño de juegos educativos, el potencial del recurso para la autoevaluación entre otros. Así también la complejidad de la tarea, propicio la conformación de grupos de trabajo entre docentes de diferentes disciplinas y residencia, que no estuvo exento de dificultades pero que entendemos pueden ser un paso para animarse a otras prácticas educativas

### **Conclusiones**

La enseñanza de programación y/o la robótica, permiten acercar las CC a las aulas de nivel medio, sin embargo el desafío de trabajarlas por áreas curriculares integradas siguiendo la propuesta de la ESRN, requiere que las y los docentes se sientan motivados, acompañados y empoderados a incluirla en sus prácticas pedagógicas, desde una postura crítica y reflexiva. En este sentido, construir lazos con instituciones y organizaciones vinculadas al tema, que permitan acercarse a diferentes enfoques de enseñanza y aprendizaje (así como también recursos) puede ser un aporte para atender a la diversidad que presentan los diferentes espacios de práctica docente.

A futuro, se espera avanzar en la implementación de una propuesta de extensión universitaria, recientemente aprobada en la convocatoria “Universidad, Cultura y territorio” de la Secretaria de Políticas Universitarias (SPU), y destinada a docentes y estudiantes de los últimos años de nivel primario y ciclo básico de nivel medio, enfocada en la programación desconectada a través del uso de un juego de mesa de producción propia denominado JAM (Juego Algorítmico de mesa).

### **Referencias**

- Dapozo, G. N., Greiner, C. L., & Petris, R. H. (2017). Programación en las escuelas. Experiencia de formación docente en el Nordeste Argentino. *Memorias del Encuentro Internacional de Educación a Distancia*, 5(5).
- Fussero, G. B., & Chiarani, M. C. (2018). Programación visual en la formación de docentes de Biología. In *Memorias de las Jornadas Nacionales y Congreso Internacional en Enseñanza de la Biología* (Vol. 1, No. Extraordinario, pp. 143-148).
- Goin, M., Lovos, E. N., & Sanz, C. (2021). Aprender a Programar en Tiempos de Pandemia. Una experiencia con Docentes de Nivel Primario.

- Lovos, E. N., Goin, M. M. J., & Gibelli, T. (2021). Juegos analógicos para el aprendizaje de algoritmos en el Nivel Primario.
- Llambi, C., Borchardt, M., Klinkovich, V., Locca, N., Martínez, C., & Scasso, M. (2023). Aprendizajes y desafíos para la enseñanza de las Ciencias de la Computación en las escuelas.
- Monjelat, N. (2019). Enseñanza de la programación en la formación docente: pautas pedagógicas desde una experiencia con maestros de primaria. In I Simposio Argentino de Educación en Informática (SAEI 2019)-JAIIO 48 (Salta).
- PLANIED: Plan Integral de Educación Digital del Ministerio de Educación y Deportes de la Nación Argentina (s.f).
- Tobeña, V., & Nobile, M. (2021). ¿Hacia dónde va el trabajo docente en secundaria? Análisis de su reconfiguración en dos políticas provinciales recientes. *Revista de Educación*, (22), 43-66.
- Consejo Federal de Educación Res. 263/15. Programación y robótica. Disponible en: [https://cfe.educacion.gob.ar/resoluciones/res15/263-15\\_01.pdf](https://cfe.educacion.gob.ar/resoluciones/res15/263-15_01.pdf)
- Consejo Federal de Educación Res. 423/22. Disponible en: [https://www.argentina.gob.ar/sites/default/files/2022/09/rcfe\\_432-22.pdf](https://www.argentina.gob.ar/sites/default/files/2022/09/rcfe_432-22.pdf)
- Consejo Provincial de Educación Resolución 642/2017. Disponible en: [https://educacion.rionegro.gov.ar/files/Direcciones\\_2020/Superior/00642-17%20declaraci%C3%B3n%20de%20inter%C3%A9s.pdf](https://educacion.rionegro.gov.ar/files/Direcciones_2020/Superior/00642-17%20declaraci%C3%B3n%20de%20inter%C3%A9s.pdf)

# JADiCC

Jornadas Argentinas de Didáctica  
de las Ciencias de la Computación

1 y 2 de diciembre de 2023  
NEUQUÉN  
PATAGONIA - ARGENTINA

---

ISBN 978-987-604-652-7

