

# **Actas de las Cuartas Jornadas Argentinas de Didáctica de las Ciencias de la Computación JADiCC 2024**

Octubre de 2024

Universidad Nacional de Río Cuarto

## Organizadores



**UNRC**



Facultad de Ciencias Exactas  
Físico-Químicas y Naturales  
UNRC



DEPARTAMENTO  
DE COMPUTACIÓN

<Program.AR/>  
DIEZ AÑOS



## **Chair**

Francisco Bavera - UNRC/UNViMe

## **Chair Track Artículos**

Claudia Queiruga - LINTI-FI-UNLP  
Marcela Daniele - UNRC

## **Chair Track Experiencias Docentes**

Natalia Colussi - FCEIA-UNR  
Claudia Banchoff Tzancoff - LINTI-FI-UNLP

## **Chair Track Talleres**

Cecilia Martinez - UNC  
Jaqueline Fernandez - UNSL

## **Chair Track Demostración Herramientas**

Maria Carmen Leonardi - INTIA/FCE-UNICEN  
María Cristina Werenitzky Curia - FACET-UNT

## **Comité Académico**

Gabriela Noemí Aranda - UNCo  
Marcelo Arroyo - UNRC  
Claudia Banchoff Tzancoff - UNLP  
Francisco Bavera - UNRC/UNViMe  
Verónica Bollati - CONICET/UTN-FRRe  
Fernando Bordignon - UNiPe/UBA  
Maria Virginia Brassesco - UNiPe/UBA  
Fernanda Carmona - UNdeC  
Ana Casali - FCEIyA-UNR/CIFASIS  
Claudia Casariego - UNLa  
Laura Cecchi - UNComa  
Natalia Colussi - FCEIA-UNR  
Marcela Daniele - UNRC  
Gladys Dapozo - UNNE  
Gustavo Del Dago - FS/UNPAZ  
Lucila Dughera - CONICET  
Maria Emilia Echeveste - UNC  
Jacqueline Fernandez - UNSL  
Gonzalo Fernández - UNQ/UBA  
Fernando Frati - UNdeC  
Laura Garbarini - UNLa  
María Fernanda Golobisky - UTN-FRSF

Marcos Gómez - UNC  
Renata Guatelli - UNLaM/UNAJ  
Marta Lasso - UNPA/UACO  
Carmen Leonardi - INTIA/UNICEN  
Matías López y Rosenfeld - UBA  
Cecilia Martinez - UNC/CONICET  
Pablo Martínez López - UNQ/UNaHur/FS  
Analía Méndez - UNSE  
Ilda Flavia Millán Tejada - UNSJ  
Natalia Monjelat - CONICET/UNR  
Noelia Pinto - UTN-FRRe  
Fernando Puricelli - UNaHur  
Claudia Queiruga - UNLP  
Teresa Quintero - UNRC  
Alice Rambo - UNaMisiones  
Nora Reyes - UNSL  
Jorge Rodríguez - UNComa  
Fabiana Rosso - ISFD Menendez Pidal  
Gladis Sequeira - UNAM  
Pablo Turjanski - UBA/CONICET  
María Cristina Werenitzky Curia - UNT  
Dante Zanarini - UNR

## **Comité Organizador**

Francisco Bavera - UNRC/UNViMe  
Mariano Benet - Fundación Sadosky  
Marcela Daniele - UNRC  
Teresa Quintero - UNRC  
Marcelo Arroyo - UNRC  
Cesar Cornejo - UNRC  
Cecilia Kilmurray - UNRC

Marcelo Uva - UNRC  
Romina Yppolito - UNRC  
Sonia Perdigiani - UNRC  
María Marta Novaira - UNRC  
Agustín Nolasco - UNRC  
Nicolás Streri - UNRC

# Índice

<b>1 - Introducción</b>	1
<b>2 - Conferencias</b>	3
<b>3 - Mesas de Discusión</b>	4
<b>4 - Talleres</b>	5
<b>5 - Demostraciones de Herramientas</b>	7
<b>6 - Artículos</b>	10
<b>Eliminar el ciclo de invisibilidad y reducir barreras para el aprendizaje y la participación (BAP): Hacia la construcción de entornos virtuales de enseñanza y aprendizaje inclusivos en la enseñanza de la Programación y Robótica.</b> Rocio Noelia Rodriguez.	11
<b>Mediación Tecnopedagógica en la formación de alumnos de Primer Año de Ingeniería implementada en un modelo Bimodal.</b> Artemisa Trigueros.	24
<b>Investigación en el profesorado universitario en computación: presente y futuro.</b> Julia A Zacharski, Agustín Gómez Díaz, Miguel Avalos.	37
<b>¿Es posible enseñar los fundamentos de la programación sólo con Arduino?: Análisis de un curso introductorio para estudiantes de secundaria usando bloques.</b> Gonzalo Pablo Fernández, Christian Cossio-Mercado.	44
<b>¿Cómo llevar lo que sabemos de Metacognición a las clases de programación?</b> Christian Cossio-Mercado, Gonzalo Pablo Fernández, Gastón Pérez.	58
<b>Implementación de Microlearning utilizando Susurros: Una Propuesta en el ámbito Universitario.</b> Claudia Gabriela Alderete.	71
<b>La Inteligencia Artificial en Educación Media de la República Argentina.</b> Valeria C González Angeletti.	80
<b>Explorando la enseñanza de conceptos de programación utilizando diferentes lenguajes de programación.</b> José M. Massa, Carmen Leonardí, Virginia Mauco.	95
<b>Recomendaciones acerca de los escenarios al momento de introducir en clases algoritmos de ordenamiento de secuencias de elementos.</b> Tobías Valdés Castro, Pablo Turjanski, Matías Lopez-Rosenfeld.	105
<b>Impacto de un módulo de formación docente en las opiniones sobre arquitectura de computadoras, inteligencia artificial y problemáticas asociadas.</b> Marcos J Gómez, Julián Dabbah, Tomas Caballero.	117
<b>Criterios de evaluación en las materias de programación de las carreras del Módulo de Informática de la FCEQyN UNaM.</b> Daniela A Martinez.	129
<b>Una aproximación a las actividades en la enseñanza superior de programación de videojuegos.</b> Gaston Caminiti.	137
<b>Uso de IA en programación inicial desde la perspectiva de los estudiantes.</b> Gladys Dapozzo, Cristina Greiner, Raquel Petris, Ana María Company, María Cecilia Espíndola.	144
<b>La construcción de estadísticas relativas al acceso a saberes digitales en Argentina.</b> Cecilia Martinez, Araceli Acosta.	155
<b>Reflexiones sobre la naturaleza de la computación.</b> Gonzalo Pablo Fernández, Pablo Martínez López, Alejandro Artopoulos, Alejandra Llitteras.	167
<b>Evaluación de Habilidades de Pensamiento Computacional: avances, tendencias y resultados.</b> Marcela Daniele, Teresa Quintero, Francisco Bavera.	181
<b>7 - Experiencias Docentes</b>	190
<b>Uso de Herramientas de Software y Estrategias Didácticas para Facilitar la Comprensión de los Alumnos en la Enseñanza de Compiladores.</b> Gladis Sequeira, Gabriela Del Rocio Gomez, Cristian Kornuta.	191
<b>El desarrollo de habilidades de programación en el tramo superior de formación de carreras informáticas en la era de ChatGPT.</b> Paola Budan, Margarita Alvarez.	196
<b>El pensamiento computacional como propuesta tecnopedagógica centrada en la resolución de problemas para el aprendizaje de la geometría en cuarto de primaria.</b> Marcos Ariel Conte, Fabiana Justina Gilardoni.	203
<b>Algoritmia y juego en la escuela primaria. Miradas sobre una experiencia desconectada.</b> Edith Lovos, Martín Goin, María de la Trinidad Quijano.	212
<b>Iniciación a la programación en contextos de encierro con Arduino.</b> Daniela Macario Cabral, Christian Cossio-Mercado, Gonzalo Pablo Fernández.	221

<b>Inteligencia Artificial Generativa como herramienta para el desarrollo de habilidades de investigación en la asignatura toxicología de los alimentos.</b> Maria Laura Vranic.	231
<b>Codificación a la carta.</b> Carolina A Longo.	237
<b>Cultivando el Futuro: Diseño y Aplicación de un Riego Automático en el Aula.</b> Yanina Swarynski.	242
<b>Integración de Herramientas de Desarrollo de Software y de Aprendizaje Automático en un Trabajo Integrador Final de Programación.</b> Diana Carolina Vertiz del Valle, Santiago Salinas Sosa, Jordán Francisco Insfrán, Javier Eduardo Diaz Zamboni.	251
<b>Aprender a aprender, aprendiendo: Experiencias para y con otros.</b> Mariana Frutos, Fabio Andres Zorzan, Sandra Angelí, Guillermo Fraschetti.	259
<b>Enfoque Didáctico para la Integración Transversal de Análisis y Diseño de Sistemas e Ingeniería de Software en la Carrera de Analista.</b> Ariel Arsaute, Marcelo Uva, Daniela Solivellas, Franco Brusatti, Marcela Daniele.	266
<b>Co-diseño de materiales educativos sobre computación en un Jardín de Infantes de la Provincia de Buenos Aires.</b> Lucila Dughera, Fernando Bordignon.	277
<b>La enseñanza de la informática en la Educación Semipresencial de Jóvenes y Adultos.</b> Bárbara Rocha Kermolj.	287
<b>Circuitos Creativos.</b> Marcela Rivero.	296
<b>El castor atareado como secuencia didáctica introductiva a las ciencias de la computación.</b> Guillaume Hoffmann.	306
<b>Talleres de acercamiento a la programación orientados a la escuela secundaria. Una experiencia desde un proyecto de extensión universitaria de la UNLP.</b> Isabel Kimura, Milagros Palacios, Simón Mc Govern, Candela Arias, Claudia Queiruga, Claudia Banchoff Tzancoff.	311
<b>Inteligencia Artificial en la Escuela Secundaria. Experiencias de un proyecto de extensión universitaria.</b> Isabel Kimura, Santiago Chaves, Pascual Coudannes, Valentin Domé, Claudia Queiruga, Soledad Gómez.	319
<b>Experiencia y aprendizajes del dictado del curso “La ciudadanía en un mundo atravesado por computadoras” para docentes del nivel primario.</b> Natalia Colussi, Ana Casali, Hernán Galardi, Claudia Deco, Cristina Bender.	329
<b>Modelización como propuesta pedagógica para enseñar contenidos de Ciencias de la Computación.</b> Emilia Echeveste, Jonathan Alonso, Claudio Di Paolo, Nicolas Balmaceda, Consuelo Moyano Budde.	339
<b>Pilas y Bloques, aliado como andamiaje hacia experiencia contrucciónista con robótica educativa.</b> Ana Lia Zamora.	346
<b>Programación en Grupo usando Python+ABP Una Experiencia Enriquecedora en el Aula Universitaria.</b> Natalia Colussi, Pamela Viale.	356

# Introducción

La Didáctica de las Ciencias de la Computación se ha convertido en un tema central en la sociedad contemporánea, estrechamente vinculado con las demandas de la sociedad del conocimiento. Formar personas capaces de comprender y dominar las características de esta nueva realidad no solo fomenta una ciudadanía crítica, sino también una capacidad activa para incidir en el desarrollo nacional. En lugar de ser simples usuarios o consumidores, es necesario preparar ciudadanos que participen activamente en la construcción del futuro.

En esta dirección, las Jornadas Argentinas en Didáctica de las Ciencias de la Computación (JADiCC) se suma a las iniciativas que promueven la investigación y la acción educativa en el área, reafirmando la importancia de espacios que fomenten el encuentro, la discusión y la producción de ideas relacionadas con la Didáctica de las Ciencias de la Computación. Las JADiCC tienen como principal objetivo reunir a investigadores, docentes y estudiantes interesados en el tema, promover el debate académico y construir una red de intercambio de experiencias y conocimientos que beneficie a toda la comunidad educativa.

Desde su origen en las Jornadas Argentinas de Didáctica de la Programación (JaDiPro) en 2018 (UNQui) y 2019 (UNC), estas jornadas han evolucionado para abarcar un espectro más amplio, consolidándose como un espacio clave en el ámbito educativo. En la edición de 2021 se extendió el área de interés a la didáctica de las Ciencias de la Computación en general. A partir de entonces las JADiCC se llevaron a cabo en 2021, en modalidad virtual organizadas por la Fundación Sadosky, en 2022 se retornó a la presencialidad en la Universidad Nacional del Nordeste y la Universidad Nacional del Comahue fue sede en 2023. Llegando a la edición 2024, organizada en formato virtual por el Departamento de Computación de la Facultad de Ciencias Exactas, Físico-Químicas y Naturales de la Universidad Nacional de Río Cuarto.

Este evento reunió el esfuerzo conjunto de más de 35 instituciones, incluyendo universidades, institutos de investigación y centros de formación docente, reflejando una verdadera construcción colectiva. Con una convocatoria que superó los 2000 participantes y la realización de dos conferencias, diez talleres, cuatro mesas de discusión y la presentación de dieciséis investigaciones, veintidós relatos de experiencias docentes y seis herramientas, las jornadas reafirmaron su importancia como un espacio de formación, reflexión y colaboración.

Además, la participación de representantes de todas las provincias argentinas y de varios países de América Latina subraya la relevancia de este encuentro para nuestra comunidad. Este intercambio de perspectivas e ideas no sólo enriqueció a los participantes, sino que sentó las bases para futuras colaboraciones y proyectos conjuntos.

Con la publicación de estas actas, queremos no solo documentar lo acontecido, sino también dejar testimonio del compromiso colectivo que anima a esta comunidad. Esto representan un punto de partida para nuevas investigaciones, prácticas y reflexiones que fortalecerán aún más el área de interés de las JADiCC. Sabemos que los desafíos son grandes, pero también tenemos la certeza de que, trabajando juntos, podremos construir una educación en Ciencias de la Computación que sea inclusiva, crítica y transformadora.

Estas actas están estructuradas en torno a las actividades desarrolladas en las jornadas. Primero se presenta una breve descripción de las conferencias dictadas. Luego se reseñan las mesas de discusión en las que interactuaron reconocidos especialistas. En las dos secciones siguientes se describen los talleres de formación docente que se llevaron a cabo en el marco de las JADiCC y las herramientas para la enseñanza de temas de las Ciencias de la Computación expuestas. Por último se presentan los trabajos presentados divididos en dos secciones artículos, que presentan investigaciones y sus resultados en el área de la Didáctica de las Ciencias de la Computación, y experiencias docentes de enseñanza y aprendizaje de las Ciencias de la Computación, en todos los niveles del sistema educativo.

Finalmente, queremos expresar nuestro profundo agradecimiento a todos los que hicieron posible esta edición de las JADiCC. Desde los organizadores hasta los asistentes, cada aporte y cada participación contribuyó al éxito de este evento. Los invitamos a sumarse nuevamente en 2025, cuando las jornadas tendrán lugar en la Facultad de Ciencias Físico-Matemáticas y Naturales de la Universidad Nacional de San Luis.

¡Muchas gracias y hasta el próximo año!

# Conferencias

## Conferencia de Apertura: "Supercómputo"

Nicolás Wolowick - UNC

Resumen: Esta conferencia gira en torno a los siguientes interrogantes: ¿Hay supercomputadoras en Argentina? ¿Hay en LATAM? ¿Cómo son? ¿Para qué se usan? ¿A quién pertenecen? ¿Cuál es su costo de usarlas, mantenerlas...? ¿Hay impacto ambiental? Entre otros aspectos relevantes del tema.

## Conferencia de Cierre: "Blockchain y el futuro descentralizado: implicaciones para la educación"

Laura Tardivo (UNRC - Bitcoin Computer)

Resumen: Esta conferencia explora la tecnología blockchain y su impacto en la educación. Se presenta blockchain como una cadena de bloques que almacena datos de forma segura, transparente y descentralizada, destacando sus componentes clave y características principales como la inmutabilidad y la seguridad. A continuación, se aborda la descentralización, mostrando cómo la distribución del control entre múltiples nodos reduce los costos y aumenta la transparencia y la seguridad. La charla ilustra el funcionamiento de blockchain, incluida la creación, verificación y adición de bloques a través de mecanismos de consenso como Prueba de trabajo y Prueba de participación. Se presentan aplicaciones generales de blockchain, que van desde criptomonedas hasta contratos inteligentes. En el ámbito educativo se destacan aplicaciones específicas como la gestión segura de la identidad, la emisión de certificados digitales verificables y la transparencia en las evaluaciones. Finalmente, se discuten las tendencias emergentes y el potencial futuro de blockchain en la educación, enfatizando cómo esta tecnología puede transformar la gestión de datos y personalizar el aprendizaje, abriendo nuevas oportunidades para la innovación educativa.

# Mesas de Discusión

## **Mesa 1 "Computación en el Nivel Inicial"**

Panelistas: Lucila Dughera (CONICET-e-TCS), Fernando Bordignon (UNIPE), Martín Torres UNC).  
Moderadora: María Cecilia Martinez (CONICET - UNC).

El objetivo principal será conocer a distintos docentes e investigadores que trabajen en temas relacionados con el desarrollo de la computación en el nivel inicial. La reunión forma parte de una agenda de investigación más amplia que consiste en mapear experiencias educativas, instituciones y personas que trabajen en temas relacionados con las ciencias de la computación y su didáctica en los distintos niveles educativos.

## **Mesa 2 "Datos y Género en TEC"**

Panelistas: Veronica Marino (Fundación Quantitas), Milagros Gimenez (Chicas en Tecnología), José Antonio Pozzer (UNNE-CONICET).

Moderadora: Ivana Feldfeber (Data Género).

La brecha de género en el ámbito de la formación y desarrollo profesional en Informática sigue siendo muy alarmante. Tanto en los espacios formativos como de producción de conocimiento o desarrollo de software las mujeres representan menos del 30 % del total y sobre las diversidades escasean los datos. Sin embargo, investigadoras e investigadores se ocupan de darle seguimiento a este tema y entender de manera más precisa el fenómeno planteando hipótesis y realizando pesquisas en Argentina y la región. Chicas en Tecnología, el IDES, la Fundación Sadosky y Data Género se reúnen en esta mesa para compartir los ejes de sus investigaciones más recientes en la materia, dar cuenta del tipo de datos que faltan, contar cómo sortearon esas vacancias para dar respuesta a sus hipótesis y pensar juntos soluciones a futuro para el relevamiento sistemático de datos que permita diseñar políticas que mitiguen el crecimiento de la brecha.

## **Mesa 3 "De la Alfabetización a la Informática y de la Informática a la alfabetización"**

Panelistas: Viviana Harari (Directora del proyecto de extensión "El barrio va a la universidad" UNLP), Beatriz Diuk (creadora del programa de alfabetización DALE!), Fernando Bordignon (UNIPE).

Moderadora: Claudia Queiruga (UNLP).

Se propone aportar una mirada desde el nivel inicial y la importancia del desarrollo del pensamiento computacional como un saber que colabora en mejorar otras alfabetizaciones en función de las últimas investigaciones y meta análisis realizados sobre el tema.

## **Mesa 4 "Competencias digitales para la enseñanza y la evaluación"**

Panelistas: Alejandra Santos (UNLa), Cecilia Sagol (UNLa-UNDAV), Clara Ingrassia (UNLa) y Walter Campi (UNQui).

Moderadoras: Claudia Casariego (UNLa) y Laura Garbarini (UNLa).

Se propone en esta mesa poner en el centro de la discusión las competencias digitales docentes para la enseñanza y la evaluación, abordando en forma transversal temáticas y problemáticas vinculadas con la ciudadanía digital, la IA y la ludopatía.

# Talleres

## **Taller 1 “Estrategias para la enseñanza de la Programación Lógica en la Escuela”**

Dictado por Jorge Rodríguez y Laura Cecchi (UNComa).

Prolog (derivado de “PROgrammation en LOGique”) es un lenguaje de Programación Lógica que nació a comienzos de la década de los 70, de la mano de Alain Colmerauer, un investigador de la Universidad de Marsella. En el marco de esta propuesta se le presenta al grupo de docentes una colección de problemas susceptibles a ser tratados con herramientas de la Programación Lógica. En general se abordan problemáticas presentes en las propuestas curriculares de la educación obligatoria. Trabajar en base a este tipo de problemas busca situar la propuesta en el contexto específico del nivel escolar, abriendo la posibilidad de explorar probables formatos para la inserción de programación lógica a la educación. En este contexto la propuesta busca ofrecer un contacto satisfactorio con prácticas y conceptos fundamentales sobre programación lógica y codificación en Prolog.

## **Taller 2 “La accesibilidad en la educación superior: aportes desde las Ciencias Informáticas”**

Dictado por Ivana Harari, Ana María Úngaro y Paola Amadeo (UNLP).

En abril de 2024, durante el Foro de la Juventud de la UNESCO, se destacó el derecho a la educación superior para estudiantes con discapacidad como un derecho humano esencial para combatir las desigualdades. A pesar de los avances, persisten barreras como la falta de accesibilidad y la discriminación. Los estudiantes con discapacidad enfrentan obstáculos en áreas como enseñanza, comunicación y accesibilidad digital en materiales y plataformas educativas. La discapacidad se entiende como una construcción social influida por factores ambientales que pueden limitar o empoderar. El taller propone desde las Ciencias Informáticas fomentar la accesibilidad en educación respetando estándares de diseño universal para garantizar una educación inclusiva y de calidad.

## **Taller 3 “Introducción al diseño de trayectos de formación en RT3D con Unity Engine”**

Dictado por Gastón Caminiti (UNL).

El taller ofrece a los participantes un espacio de aproximación inicial sobre el diseño de programas educativos focalizados en desarrollo o producción de realidad virtual (VR), aumentada (AR), extendida (XR) o videojuegos. En el contexto de la industria 4.0, donde las aplicaciones inmersivas cobran relevancia, interesa no solo consumirlas, sino también constituir o afianzar programas destinados a la formación de profesionales a desempeñarse en la industria RT3D. En el encuentro se exploran y problematizan configuraciones curriculares y su adecuación al contexto regional, bajo un enfoque práctico y colaborativo que permita a los asistentes dar los primeros pasos hacia la planificación de estos trayectos en la propia institución educativa.

## **Taller 4 “Desarrollo de proyectos de enseñanza de programación con p5.js”**

Dictado por Marcelo Arroyo, Valeria Bengolea y Gastón Scilingo (UNRC).

La enseñanza de programación en los diferentes niveles educativos puede potenciarse con el uso de entornos de programación que permitan a los estudiantes el desarrollo de aplicaciones gráficas con animaciones que permitan ver la evolución de las computaciones y visualizar fácilmente los resultados. El proyecto p5.js ofrece bibliotecas de desarrollo de programas gráficos y animaciones en la web basado en un enfoque que relaciona computación y arte. El único requisito es el uso de un navegador web y opcionalmente un editor de programación. Además el proyecto ofrece un editor en línea para el desarrollo de aplicaciones con acceso libre. Este taller tiene como objetivos, la introducción a entornos de programación basados en p5.js, el desarrollo de proyectos de enseñanza y entrenamiento de programación y su uso en entornos web educativos.

## **Taller 5 “Demostración de herramienta/entorno Tinkercad”**

Dictado por Carolina Wayar (Instituto Privado Mixto San Agustín).

Es innegable que hoy nuestro que hacer diario está mediado por dispositivos tecnológicos y los diversos softwares que los hacen funcionar. Nuestras aulas no están ajena a ello sea por la necesidad de conciliar estos dispositivos con el rol docente y las dinámicas que se quieren poner en práctica o bien por la intención concreta de hacer uso de los mismos como herramienta en propuestas de aprendizajes. Entre las diversas propuestas de enseñanza que involucran el manejo de software se encuentra la robótica educativa. Esta disciplina permite trabajar software y hardware específicos en proyectos concretos poniendo en juego saberes, habilidades y capacidades específicas: programación, diseño de prototipos, resolución de problemas y trabajo colaborativo entre otros. La propuesta de este taller es trabajar con el simulador Tinkercad para programar y simular circuitos con la placa Arduino.

## **Taller 6 “Un tipo de problemas que favorecen los procesos de generalización”**

Dictado por Flavia Buffarini (UNRC), Fabiana Rosso (ISFD Ramón Menéndez Pidal) y Francisco Bavera (UNRC).

En este Taller se pretende hacer visible el rol de la resolución de una clase de problemas en la construcción del pensamiento computacional, a partir de la reflexión didáctico-matemático en el marco de la Didáctica de la Matemática. Se presenta una clase de problemas cuya resolución posibilita la construcción del pensamiento computacional, en tanto favorece el desarrollo de habilidades de pensamiento de orden superior tales como: abstracción, generalización, reconocimiento de patrones, pensamiento lógico, exhaustividad, economía, verificación, validación, descomposición de problemas en problemas más sencillos y manejables, etc.

## **Taller 7 “¿El ChatGPT sabe lo que dice?”**

Dictado por Marcos Gómez, Julián Dabbah (Fundación Sadosky).

El taller propone explorar herramientas de IA y de IA generativa como una aproximación a su funcionamiento y para exhibir sus limitaciones en cuanto a sesgos y alucinaciones.

## **Taller 8 ¿Por qué cada vez existen más restricciones para crear contraseñas?**

Dictado por Débora Cingolani, Marcos Gómez (Fundación Sadosky).

¿A qué obedecen las restricciones que nos plantean las aplicaciones y sitios web para crear contraseñas? ¿Es posible que alguien descifre mi contraseña? ¿Existen contraseñas seguras? ¿Qué hace segura a una contraseña?

## **Taller 9 “Pilas Bloques Desenchufado”**

Dictado por Rocio Gonzalez, Diana Lopez Alvas, Daniel Ferro (Fundación Sadosky).

Plantear una secuencia didáctica para que estudiantes de primer ciclo de primaria aprendan los principios básicos de la programación con Pilas Bloques desenchufado.

## **Taller 10 "Análisis Crítico de Plataformas Educativas para Enseñar a Programar"**

Dictado por Angela Hahn, Yamila Ojeda, Virginia Brassesco (UniPe).

Elegir una plataforma para enseñar programar no puede depender de la moda del momento, como docentes debemos tener claros los ejes que ponemos en juego a la hora de elegir herramientas en base a nuestros estudiantes, sus gustos y sus motivaciones.

# Demostraciones de Herramientas

## Demostración 1 “Hacia una nueva experiencia de programación por medio de dibujos”

Presentadores Agustín Rafael Martínez y Agustín Alejo Fernández Ortuzar (UBA).

Descripción: Se trata de un entorno de programación donde niñas y niños crean programas por medio de sus propios dibujos. Esto permite programar por medio de símbolos que pertenecen al universo de los propios niños, ya que son ellos quienes los han creado previamente. Estos dibujos se arrastran y arrojan permitiendo una experiencia directa sobre lo que es considerado la esencia de la computabilidad: la manipulación de símbolos. Este nuevo lenguaje ha sido presentado en 2022 en conferencias internacionales de programación, entre ellas ACM Splash. En 2023 han sido testeado en un contexto de aula en un taller de tres jornadas de duración abierto a la comunidad. Desde 2024 está disponible online para ser descargado.

Link: <https://represent.ar/>

## Demostración 2 “Construye tu propio Tortuguino: programación gráfica con Arduino y Turtle”

Presentadora Agustina Paredes (Código STEM).

Descripción: Tortuguino es una variante de un proyecto Open Source muy popular en Arduino: un robot dibujante. Es fácil de construir, fácil de programar, económico y cumple funciones interesantes que se pueden aplicar dentro del aula. Desde enseñar comandos básicos para aprender a programar hasta trabajar de forma transversal con conceptos de otras áreas, por ejemplo figuras geométricas en matemática. Es compatible con plataformas como <https://code.org>, <https://turtleacademy.com/>, entre otras, que se enfocan en programación gráfica y en bloques.

Link: [Tortuguino](#).

## Demostración 3 “Arduino en la Escuela: Programación por bloques para Arduino”

Presentador Gonzalo Pablo Fernández (UBA).

Descripción: Arduino en la Escuela (AeLE) es un entorno de programación por bloques para Arduino. Se destaca por su versatilidad, en tanto puede ser utilizada en contextos variados y con públicos de características muy diferentes, no sólo en edad, sino también en contexto sociocultural y formación previa. Algunas de las principales características que la distinguen del resto de los entornos de programación por bloques para Arduino son su portabilidad (es gratuita, funciona en Windows y Linux está completamente en español y no requiere internet, registrarse ni contar con permisos de administrador para instalarlo), su robustez, su diseño del lenguaje (incluye parámetros, listas y eventos, además de contar con un extenso catálogo de bloques para comunicarse con sensores y actuadores varios) y su personalización (permitiendo al docente configurar gran parte de su comportamiento, como los bloques disponibles y su organización en categorías o la forma en que se manejan los errores y las advertencias, para adaptarla a sus clases particulares). También se destacan características que optimizan la producción y corrección de código, como la posibilidad de seleccionar varios bloques a la vez, un buscador de bloques, las ediciones rápidas en los menús contextuales (que incluyen autocompletar argumentos y refactorizar partes del código) y la navegación por teclado. De esta forma, Arduino en la Escuela resuelve parcial o totalmente las limitaciones de otras herramientas disponibles para programar placas Arduino por medio de bloques, por lo que puede ser de utilidad para docentes e instituciones que busquen una herramienta liviana, versátil y de uso libre.

Link: <http://aele.dc.uba.ar>

## **Demostración 4 “Pandalyze: una introducción al análisis de datos programando con bloques”**

Presentadora Sofía Martín (UNLP).

Descripción: Pandalyze es una herramienta que brinda un entorno de programación basado en bloques orientada a introducir aspectos básicos de análisis de datos. Esta herramienta presenta una interfaz similar a Scratch o Pilas Bloques, en donde los programas se construyen encastrando bloques con las instrucciones de forma más intuitiva que la escritura de código puro. Los bloques en Pandalyze representan funcionalidades específicas de la librería pandas de Python, permitiendo realizar operaciones comunes de manipulación y análisis de datos de manera visual. En la misma herramienta se muestra el código Python equivalente, con lo cual es una buena forma de introducir las funcionalidades básicas para luego, si es que se desea, seguir el trabajo con Python y otras herramientas muy utilizadas como Jupyter Notebook. Esta integración también facilita la transición hacia la programación en Python.

Link: <https://668fdd58badbd2000869cde0--pandalyze.netlify.app/>

## **Demostración 5 “Prolog Web App Creator”**

Presentador Santiago A. Villarroel (UNCoMa).

Descripción: Prolog Web App Creator es un ambiente de desarrollo integrado (IDE) que permite a principiantes y a personas que no son programadores, diseñar y crear aplicaciones Web con funcionalidades que incluyen la consulta a programas Prolog. Permite a niños y adolescentes crear software, basado sobre su propio código Prolog, o bien sobre algún código Prolog remixado, transformándose en creadores de tecnología, en vez de ser meros consumidores. Quienes usan Prolog Web App Creator para crear aplicaciones web, pueden desarrollar software utilizando su propio código Prolog o remezclando el código de sus compañeros. La interfaz consta de dos vistas: la vista de diseño y la vista del artefacto resultante. La vista de diseño permite al estudiante crear la aplicación, inicialmente se proporciona un programa Prolog como entrada e insertando botones e imágenes en un lienzo se avanza en el proceso de construcción del artefacto. Cada predicado en el código genera un botón susceptible a ser arrastrado en el lienzo como pieza de la aplicación que consulta sobre la base de conocimientos provista como entrada. En esta vista, los estudiantes producen su artefacto y prueban el programa lógico antes de generar la aplicación final. La vista del artefacto resultante muestra la interfaz diseñada y ejecuta el código presionando los botones colocados por el estudiante. A partir de la aplicación creada en Web generada con Prolog Web App Creator, los usuarios podrán consultar un programa Prolog en forma transparente, esto es, sin conocer la sintaxis de este lenguaje. La facilidad de uso, la colaboración y la publicación del resultado también son aspectos relevantes del entorno, permitiendo a los estudiantes compartir el diseño y el producto con sus círculos sociales. Los principales aportes de esta herramienta son la posibilidad de generar una aplicación Web sin conocimiento tecnológico sobre cómo realizarlo, e interactuar con un programa Prolog, aún desconociendo cómo se consulta una base de conocimiento. Prolog Web App Creator promueve el empoderamiento de las personas al diseñar, crear, compartir y usar soluciones de su propia autoría con tecnología Web, en su entorno cotidiano. Particularmente, en la enseñanza de la programación utilizando el lenguaje Prolog, colabora en la difusión y uso real de las soluciones que se desarrollen en este lenguaje de Programación Lógica. De este modo, los estudiantes pueden compartir con sus familias, compañeros y comunidad en general sus creaciones para que los prueben y utilicen.

Link: <https://prologforkids.fi.uncoma.edu.ar/>

## **Demostración 6 “ArguiBot IDE: entorno de desarrollo basado en la programación en bloque para la enseñanza de la robótica usando la plataforma Arduino”**

Presentador Nicolás Streri (UNRC).

Descripción: ArguiBot IDE es un entorno de desarrollo de código abierto diseñado para la enseñanza de la robótica y la programación a estudiantes de nivel secundario. Se basa en un enfoque de programación por bloques, que permite a los estudiantes sin experiencia previa en programación familiarizarse rápidamente con los conceptos fundamentales de la robótica y la codificación. Está construido sobre la plataforma Arduino, lo que le otorga una gran versatilidad y accesibilidad en términos de hardware compatible. ArguiBot ofrece un conjunto reducido de bloques en español que permiten a los estudiantes crear algoritmos y controlar dispositivos electrónicos sin necesidad de conocer en profundidad los lenguajes de programación tradicionales. Esta simplicidad no solo hace que el aprendizaje sea más accesible para los estudiantes, sino que también reduce las barreras técnicas para los docentes que desean incorporar la robótica en sus planes de estudio. El entorno está diseñado para trabajar con una amplia gama de módulos plug and play genéricos disponibles en el mercado. ArguiBot permite a las instituciones educativas y a los estudiantes utilizar componentes electrónicos económicos y fáciles de adquirir, adaptándose a diferentes contextos educativos. Además, uno de los aspectos destacados de ArguiBot es la incorporación de funciones para mejorar la experiencia de enseñanza y aprendizaje, como el sistema de calificación y feedback para el seguimiento del progreso de los estudiantes y la entrega de trabajos. Lo que permite a los docentes monitorizar y evaluar el desarrollo de competencias en sus alumnos de manera más estructurada y eficiente.

Link: <https://github.com/nicostreri/ArguiBot>

# **Artículos**

Versión Preliminar

# Eliminar el ciclo de invisibilidad y reducir barreras para el aprendizaje y la participación (BAP): Hacia la construcción de entornos virtuales de enseñanza y aprendizaje inclusivos en la enseñanza de la Programación y Robótica.

**Lic. Rocio Noelia Rodriguez<sup>1</sup>**

<sup>1</sup> Ministerio de Educación GCBA- Agencia de Habilidades para el futuro, CABA, Argentina  
[rocionoeliaaguero4486@gmail.com](mailto:rocionoeliaaguero4486@gmail.com)

## Resumen

Este estudio exploratorio analiza funcionalidades nativas y extensiones para diseñar entornos virtuales de enseñanza y aprendizaje (EVEA) inclusivos y accesibles para personas con diversidad funcional, utilizando el LMS del programa "Aprende Programando" de la Agencia de Habilidades para el futuro del Ministerio de Educación del GCBA como caso de estudio. Se examina herramientas y adaptaciones basadas Pautas de Accesibilidad Web (WCAG) 2.1, que establecen cuatro principios clave (perceptible, operable, comprensible y robusto) y tres niveles de conformidad (A, AA y AAA) con los principios DUA, la accesibilidad cognitiva, de la información y comunicación, tecnológica, con el fin de abordar diversas barreras (sensoriales, motrices, de aprendizaje, económicas e idiomáticas) y promover una educación en ciencias de la computación más equitativa. La investigación busca problematizar el ciclo de invisibilidad y las barreras para el aprendizaje y la participación (BAP) que se manifiesta en el diseño de propuestas educativas en los EVEA.

Palabras Clave: Inclusión Educativa; Accesibilidad Web; Moodle; Diseño Universal para el Aprendizaje; Barreras para el Aprendizaje y la Participación

## 1. Introducción

Este estudio exploratorio se propone contribuir a los diálogos actuales sobre la inclusión educativa en entornos virtuales de aprendizaje (EVA), considerando la diversidad funcional y cognitiva de las personas. Para ello, se fundamenta en el marco de los derechos humanos y la educación inclusiva, respaldado por la Convención sobre los Derechos de las Personas con Discapacidad (CDPD) de la ONU (2006)[1], que establece el derecho a la educación en igualdad de condiciones y promueve un sistema educativo inclusivo a todos los niveles. A nivel nacional, la Ley Nacional de Educación 26.206 (2006)[2] y la Ley Nacional 26.378 (2008)[3] garantizan el derecho a la educación sin discriminación y promueven la eliminación de barreras para el aprendizaje y la participación. Adicionalmente, compromisos internacionales como la Declaración de Nueva Delhi (2014)[4] y los Objetivos de Desarrollo Sostenible (2015)[5] refuerzan la necesidad de garantizar una educación de calidad, inclusiva y equitativa, y promover oportunidades de aprendizaje a lo largo de la vida para todas las personas.

Desde la perspectiva de derechos humanos e inclusión educativa, este trabajo adopta el modelo social de la discapacidad, que entiende la discapacidad no como un atributo individual, sino como resultado de la interacción entre las personas con deficiencias y las barreras impuestas por el entorno (Palacios, 2008)[6]. Este modelo contrasta con el modelo médico, que individualiza la discapacidad y la considera un problema a ser "corregido" o "superado". La CDPD refuerza esta perspectiva al definir la discapacidad como un concepto en evolución que resulta de la interacción entre las personas con deficiencias y las barreras del entorno que obstaculizan su participación plena y efectiva en la sociedad.

Una de estas barreras, profundamente arraigada en las actitudes y prácticas sociales, es el capacitismo, un sistema de creencias y prácticas discriminatorias hacia las personas con discapacidad que perpetúa el ciclo de invisibilidad, excluyendo a este colectivo de la participación plena en la sociedad (Mareño Sempertegui y Torrez, 2013)[7]. Este ciclo se manifiesta en la falta de acceso a oportunidades educativas y de desarrollo, lo que a su vez refuerza la discriminación y la exclusión. Romper este ciclo implica reconocer y valorar la diversidad, así como eliminar las barreras que impiden la participación plena de las personas con discapacidad. El capacitismo se entrelaza con

otras formas de discriminación, como la basada en la pobreza, el género, la etnia o el origen migratorio, creando situaciones de discriminación múltiple o interseccional que agravan la exclusión (Gómez & Tavarez, 2018; Calvo, 2001; Díaz Velázquez et al., 2011) [8] [9] [10].

En línea con estos marcos teóricos y conceptuales, el concepto de inclusión educativa se refiere a garantizar que todos los estudiantes, incluyendo aquellos con discapacidad, tengan acceso igualitario a oportunidades de aprendizaje y participación en entornos educativos diversos (Mareño Sempertegui y Torrez, 2013) [7]. Para lograr la inclusión, es fundamental abordar las barreras para el aprendizaje y la participación (BAP), que son obstáculos que impiden o dificultan el acceso, la participación y el logro de los estudiantes en el proceso de aprendizaje. Estas barreras pueden ser físicas, actitudinales, comunicacionales, metodológicas, instrumentales y programáticas (Werneck, 2005) [11].

En el ámbito de los entornos virtuales de enseñanza y aprendizaje (EVEA), entendidos como plataformas con potencial ubicuo que facilitan la enseñanza y el aprendizaje por medio del uso de herramientas tanto tecnológicas digitales como de la Internet (Brioli, C., Amaro, R., & García, I. 2011) [12], las BAP pueden manifestarse como barreras de comunicación e información (Holst, 2006) [13], dificultando el acceso, la comprensión y el uso de la información por parte de las personas con discapacidad. Estas barreras pueden ser físicas, sensoriales, cognitivas o tecnológicas. Para garantizar la inclusión educativa en los EVEA, es crucial considerar la accesibilidad en todas sus dimensiones, incluyendo la accesibilidad a la información y la comunicación, que incluye todas las acciones que facilitan el intercambio de información a nivel interpersonal y el acceso de las personas a la producción y uso de mensajes (visuales, audibles, táctiles, entre otros), promoviendo una imagen real y positiva de la persona, y la accesibilidad tecnológica, que se refiere a la tecnología derivada de productos, sistemas o equipos técnicos accesibles, como configuraciones específicas en plataformas de aprendizaje, extensiones para la enseñanza de la programación, compatibilidad con tecnologías de asistencia, herramientas de accesibilidad web, entre otras.

Además, la accesibilidad cognitiva, definida por Brusilovsky (2016) [14] como la facilidad con que entornos, productos y servicios pueden ser comprendidos y utilizados por todas las personas, independientemente de sus capacidades cognitivas, es un aspecto clave para garantizar la inclusión en los EVEA. Para lograrlo, se propone un modelo de diseño basado en la "Secuencia vertebradora de la accesibilidad", que busca asegurar la Seguridad Espacial Cognitiva (SEC) mediante tres pasos: un Origen fácilmente identificable, una Continuidad clara y libre de obstáculos en el camino hacia el Destino, el cual también debe ser fácilmente reconocible. Para garantizar esta continuidad y evitar la desorientación, Brusilovsky sugiere implementar un sistema de apoyos tanto físicos como virtuales, que incluyan referencias, señalización e información claras y comprensibles para todos.

En el contexto de los entornos virtuales de aprendizaje, esto se traduce en el diseño de interfaces intuitivas, contenidos claros, lenguaje sencillo y múltiples formas de representación y participación.

Complementariamente, se adopta un enfoque metodológico-analítico basado en las Pautas de Accesibilidad al Contenido Web (WCAG) 2.1[15], que establecen cuatro principios clave (perceptible, operable, comprensible y robusto) y tres niveles de conformidad (A, AA y AAA) para garantizar que el contenido web sea accesible para la mayor cantidad de personas posible.

El Nivel A, considerado el mínimo de accesibilidad, se centra en la perceptibilidad del contenido, requiriendo elementos básicos como subtítulos y alternativas de texto para videos y audios. Esto asegura que la información sea accesible para la mayoría de los usuarios, incluyendo aquellos con discapacidades visuales o auditivas.

El Nivel AA, exigido habitualmente en sitios web, amplía los requisitos del Nivel A y se enfoca en la operabilidad y comprensibilidad del contenido. Esto implica proporcionar subtítulos en vivo, descripciones de audio y una navegación clara y consistente, entre otros, para mejorar la experiencia de usuarios con discapacidades motoras o cognitivas.

El Nivel AAA, aunque es el más completo en términos de accesibilidad, no se recomienda para sitios web generalistas debido a la dificultad de cumplir con todos sus criterios en diversos tipos de contenido. Este nivel se centra en la robustez del contenido, incluyendo requisitos más exigentes como el lenguaje de signos y audio descripciones extendidas, que pueden resultar costosos y complejos de implementar. Por lo tanto, si bien es el ideal en términos de accesibilidad, su aplicación debe evaluarse cuidadosamente en función de la naturaleza y el propósito del sitio web.

Finalmente, el Diseño Universal para el Aprendizaje (DUA) se considera un marco pedagógico fundamental para promover la flexibilidad y la diversidad en la presentación de la información, la interacción con el contenido y la evaluación del aprendizaje en los EVEA (Pastor et al., 2014)[16]. El DUA propone tres principios clave:

proporcionar múltiples formas de representación, acción y expresión (ofrecer opciones para que los estudiantes interactúen con el material y demuestren su aprendizaje de diversas maneras), e implicación (Utilizar diversas estrategias para captar el interés de los estudiantes, fomentar su motivación y promover su participación activa), que guiarán el diseño e implementación de estrategias para mejorar la accesibilidad y usabilidad de los EVEA en la enseñanza de la programación.

En síntesis, este marco teórico multidimensional, que abarca desde los derechos humanos y la inclusión educativa hasta la accesibilidad cognitiva y el Diseño Universal para el Aprendizaje, proporciona una base sólida para comprender y abordar las barreras que enfrentan las personas con discapacidad en los entornos virtuales de aprendizaje. Al aplicar estos principios, se espera contribuir a la creación de EVEA más inclusivos, equitativos y accesibles para todos los estudiantes, independientemente de sus capacidades.

Respecto el caso de estudio, se seleccionó "Aprendé Programando" debido a su relevancia en el contexto actual, donde la programación y la robótica se consideran habilidades fundamentales para el futuro y por su amplia propuesta de formación en modalidad virtual. Diversos estudios y experiencias previas respaldan esta elección.

El documento "Programación y Robótica: Objetivos de Aprendizaje para la Educación Obligatoria" del Ministerio de Educación de la Nación, Argentina (2017)[17] destaca que estas disciplinas permiten a los estudiantes "desarrollar habilidades fundamentales para solucionar diversas problemáticas sociales, crear oportunidades y prepararse para su integración en el mundo del trabajo" (p. 8).

Además, investigaciones como la de Costa et al. (2023)[18], en "Computational Thinking, Visual Impairment and Melodic" enfatizan la importancia de desarrollar habilidades de pensamiento computacional desde temprana edad, y proyectos como "Scratch Jr Tactile" (de Miguel, 2023)[19] demuestran cómo la programación puede ser accesible e inclusiva para todos los estudiantes, incluidos aquellos con discapacidades.

El artículo "Scratch y Necesidades Educativas Especiales: Programación para todos" (López-Escribano & Sánchez-Montoya, 2012)[20] refuerza el valor de la programación como herramienta de aprendizaje inclusiva y motivadora, resaltando su potencial para el desarrollo de habilidades cognitivas y creativas en estudiantes con necesidades educativas especiales.

En este contexto, "Aprendé Programando" se presenta como un caso de estudio ideal para analizar cómo la inclusión y la accesibilidad pueden ser integradas en la enseñanza de la programación y la robótica, contribuyendo así a romper el ciclo de invisibilidad y exclusión que enfrentan las personas con discapacidad y promoviendo una educación en ciencias de la computación más equitativa y democrática.

## **Exploración de Acciones y Herramientas para la Inclusión y Accesibilidad en un Entorno Virtual de Enseñanza y Aprendizaje (EVEA) basado en Moodle**

En el trabajo exploratorio sobre el LMS Moodle del programa "Aprende Programando", la inclusión educativa fue abordada mediante diversas acciones que aprovechan las tecnologías digitales y funcionalidades nativas del entorno. Tomando como referencia los marcos de la accesibilidad cognitiva y el Diseño Universal de Aprendizaje (DUA), se buscó garantizar la accesibilidad y usabilidad del entorno virtual de enseñanza y aprendizaje (EVEA) para todos los participantes, considerando especialmente las personas con discapacidades sensoriales, motrices y dificultades de aprendizaje.

La revisión de la literatura sobre dificultades de aprendizaje y discapacidades (Fresneda y Mendoza, 2005; Mareño Sempertegui y Torrez, 2013; Mendoza, 2016; Juárez, A., & Pérez, J. 2004; Orjales Villar, 2000; Parra y Luque-Rojas, 2013; Puente González, 2012; Ramírez Morera, 2010) [21] [7] [22] [23] [24] [25] [26] [27] permitió aproximarse a la identificación de las barreras que pueden enfrentar personas con discapacidades sensoriales (ceguera total, baja visión, daltonismo, fotofobia, disfemia, hipoacusia, sordera), motrices (dificultades en extremidades superiores) y de aprendizaje (dislexia, disgrafía, discalculia, dispraxia, Trastorno del Procesamiento Auditivo Central, Trastorno del Lenguaje Oral, TDAH). Se tuvo en cuenta también aquellos casos de diversidad lingüística y condiciones materiales diversas.

Tabla 1. Identificación apoyos y barreras para el aprendizaje y participación.

	Barreras para el Aprendizaje y la Participación (BAP)	Apoyos en EVEA
<b>Sensoriales</b>		
Ceguera total	Dificultad para acceder a contenido visual, dependencia de lectores de pantalla y necesidad de descripciones textuales detalladas.	Proporcionar material en formatos de audio (narraciones, descripciones de audio), utilizar herramientas de texto a voz (como Natural Reader), garantizar compatibilidad con lectores de pantalla, y ofrecer transcripciones de videos y otros materiales visuales.
Baja visión	Dificultad para leer texto pequeño, bajo contraste y distinguir colores en la pantalla.	Utilizar temas de Moodle con alto contraste y fuentes personalizables, permitir el ajuste del tamaño del texto y el zoom, evitar el uso excesivo de colores y proporcionar materiales en formatos alternativos (como texto grande o audio).
Daltonismo	Dificultad para distinguir colores, especialmente rojo y verde, en elementos de la interfaz y contenidos del curso.	Evitar el uso de rojo y verde para transmitir información importante, utilizar patrones y texturas en lugar de colores para resaltar información, ofrecer descripciones textuales de gráficos y diagramas. Utilizar herramientas como el Accessibility Block de Moodle para permitir a los estudiantes personalizar los colores y el contraste de la interfaz.
Fotofobia	Sensibilidad a la luz intensa de la pantalla, puede causar dolor de cabeza, fatiga visual y dificultad para concentrarse.	Permitir a los estudiantes ajustar el brillo de la pantalla, utilizar temas de Moodle con fondos oscuros y fuentes claras, recomendar el uso de filtros de pantalla o gafas especiales, y ofrecer alternativas a los materiales visuales (como audio o texto).
Sordera/Hipoacusia	Dificultad para acceder a contenido auditivo (videos, grabaciones de clases) y participar en debates o videoconferencias en tiempo real.	Proporcionar subtítulos y transcripciones de videos y grabaciones, utilizar herramientas de transcripción automática en videoconferencias, ofrecer foros de discusión asincrónicos para facilitar la participación de estudiantes con dificultades auditivas.
Disfemia/Tartamudez	Dificultad para participar en actividades orales en tiempo real, como debates o presentaciones, debido a la ansiedad y el nerviosismo.	Ofrecer alternativas a la participación oral en tiempo real, como foros de discusión asincrónicos, tareas escritas o presentaciones grabadas. Proporcionar un entorno de apoyo y comprensión en las actividades orales, permitiendo tiempo adicional para hablar y evitando interrupciones.
<b>Motrices</b>		
Dificultades motoras finas	Dificultad para usar el teclado y el ratón en actividades que requieren precisión y rapidez, como escribir código o manipular objetos en simulaciones.	Permitir el uso de teclados alternativos (como teclados en pantalla o teclados virtuales), ofrecer opciones de navegación por voz o joystick, proporcionar herramientas de edición de código con funciones de autocompletado y sugerencias, adaptar las actividades para que requieran menos interacción física con el teclado y el ratón.

De Aprendizaje		
Dislexia	Dificultad para leer y comprender textos en la pantalla, confusión de letras y sonidos, problemas de ortografía en actividades de programación que requieren lectura y escritura de código.	Utilizar fuentes de fácil lectura, espaciado adecuado y alto contraste en los textos, ofrecer opciones de audio para los materiales de lectura, utilizar herramientas de revisión ortográfica y gramatical en editores de código y proporcionar instrucciones claras y concisas, con apoyo visual (diagramas, imágenes) para facilitar la comprensión de conceptos y tareas de programación.
Disgrafía	Dificultad para escribir de forma legible y coherente en entornos virtuales, problemas de ortografía y gramática al escribir código o participar en foros de discusión.	Permitir el uso de editores de código con funciones de autocompletado y revisión de sintaxis, ofrecer opciones de dictado por voz para la escritura de texto, utilizar correctores ortográficos y gramaticales en foros y otras herramientas de comunicación, proporcionar plantillas y ejemplos de código para facilitar la escritura y reducir la carga cognitiva.
Discalculia	Dificultad para comprender conceptos matemáticos y realizar cálculos en actividades de programación que involucran lógica y operaciones numéricas.	Proporcionar materiales de apoyo visual (diagramas, gráficos) para ilustrar conceptos matemáticos, ofrecer tutoriales y ejemplos paso a paso para la resolución de problemas numéricos, utilizar herramientas de cálculo en línea o integradas en el entorno virtual, permitir el uso de calculadoras y adaptar las actividades para que se centren en la comprensión de la lógica y los procesos matemáticos en lugar de la memorización de fórmulas o procedimientos.
Dispraxia	Dificultad para planificar y coordinar movimientos en entornos virtuales, como arrastrar y soltar elementos o manipular objetos en simulaciones.	Ofrecer alternativas a las actividades que requieren una coordinación motora fina precisa, como opciones de navegación por teclado o voz, proporcionar instrucciones claras y paso a paso para las tareas que involucran manipulación de objetos virtuales, permitir el uso de herramientas de asistencia como joysticks o trackballs, y adaptar las actividades para que se centren en la comprensión de los conceptos y la lógica en lugar de la habilidad motora.
TDAH	Dificultad para mantener la atención y concentración en actividades en línea, distracción por estímulos visuales y auditivos, impulsividad al responder a preguntas o participar en foros.	Dividir el contenido en módulos más pequeños y manejables, utilizar herramientas de seguimiento del progreso para ayudar a los estudiantes a mantener el enfoque, ofrecer descansos frecuentes durante las actividades en línea, utilizar recordatorios visuales y auditivos para
Diversidad		
Vulnerabilidad Socioeconómica	Acceso limitado a dispositivos y conectividad. Falta de familiaridad con la tecnología.	Materiales educativos de acceso offline y descargables en formatos como PDF. Adaptación de materiales a dispositivos móviles. Capacitación en habilidades digitales:

		Ofrecer tutoriales sobre el uso de Moodle y herramientas tecnológicas básicas.
Diversidad Lingüística	Dificultad para comprender instrucciones y contenidos en español. Diferencias culturales en la interpretación de la información y la interacción entre pares.	Utilizar la función de paquetes de idiomas de Moodle para traducir la interfaz y los contenidos. Integrar herramientas de traducción en la plataforma Promover actividades que valoren la diversidad cultural y lingüística en el aula virtual. Ofrecer alternativas a la participación oral en tiempo real, como foros de discusión asincrónicos, tareas escritas o presentaciones grabadas. Proporcionar un entorno de apoyo y comprensión en las actividades orales, permitiendo tiempo adicional para hablar y evitando interrupciones.

(Tabla 1. Tabla 1. Identificación apoyos y barreras para el aprendizaje y participación. Elaboración propia)

## Herramientas, extensiones y acciones para la Accesibilidad en Moodle

La exploración en la implementación de la accesibilidad en Moodle[28] ha revelado una serie de herramientas, extensiones y acciones clave que se alinean con los principios del Diseño Universal para el Aprendizaje (DUA), la accesibilidad web (WCAG) y la inclusión educativa. Al abordar la accesibilidad desde múltiples dimensiones y utilizar una variedad de herramientas y acciones, se colabora con que cada estudiante pueda alcanzar su máximo potencial en un entorno de aprendizaje inclusivo y motivador.

Tabla 2. Herramientas, extensiones y acciones para la Accesibilidad en Moodle.

Dimensión	Descripción
Diseño Intuitivo y Navegación Fluida	Implementación de "Secuencia Vertebradora de la Accesibilidad" para una navegación clara y fácil de seguir en Moodle, esto es configuración de "Redirecciónamiento al Área Personal" donde se encuentran accesos directos a cursos, bloque de calendario, notificaciones y mensajes. Al interior del aula la navegación es a través de etiquetas con pictogramas como hipervínculos que disminuyen la sobrecarga cognitiva al quitar exceso de elementos y redujo el "scroll". Configuración avanzada de la apariencia con css de Moodle y personalizaron temas Snap, Fordson y Moove para garantizar un contraste adecuado, tipografías legibles y diseños simples, facilitando la navegación y el acceso.
Información y Comunicación Accesibles	Incorporación de extensión "Enrol Notification" y "Resource Notification" para automatizar el envío de correos electrónicos a los estudiantes, informándoles sobre su matriculación en cursos, actualizaciones y cambios en los recursos.  Integración embebida de Google Calendar dentro del aula virtual y la integración del bloque calendario en el área personal que permite a los estudiantes recibir recordatorios de actividades y visibilizar eventos importantes.

	<p>Función insignias (badges) de Moodle para comunicar resultados de aprendizaje y competencias alcanzadas.</p>
	<p>Integración y diferenciación de foros: uno para avisos generales y otro para consultas específicas, optimizando así la comunicación e interacción.</p>
	<p>Recurso etiqueta con modal para presentar las grabaciones de las clases, cuyos videos fueron embebidos desde YouTube. Este modal, diseñado con estilos CSS, permite una visualización optimizada de recursos externos sin salir del entorno.</p>
	<p>Implementación de función Paquete de idioma de Moodle para permitir que estudiantes de diferentes orígenes lingüísticos accedan al contenido en su idioma preferido.</p>
Herramientas Tecnológicas para la Enseñanza	<p>El plugin Atto Syntax Highlighter permite resaltar la sintaxis del código directamente en el editor Atto de Moodle. Este resultado hace que el código sea más legible y comprensible, ayudando a los estudiantes a identificar rápidamente la estructura y los componentes del código.</p> <p>El Virtual Programming Lab (VPL) es un módulo de actividad en Moodle que gestiona las asignaciones de programación y facilita la evaluación automática. En los cursos de programación y robótica, esta herramienta permite a los estudiantes escribir y probar código en un entorno controlado, obteniendo retroalimentación instantánea sobre su desempeño. VPL no solo ahorra tiempo a los instructores en la corrección manual, sino que también proporciona a los estudiantes una experiencia de aprendizaje práctica y eficiente.</p> <p>CodeRunner es otro plugin esencial para la enseñanza de programación, permitiendo la ejecución y evaluación de programas directamente en Moodle. Los estudiantes pueden responder a preguntas de programación escribiendo y ejecutando código, lo que es evaluado automáticamente y facilitando el aprendizaje iterativo y el desarrollo de habilidades prácticas.</p> <p>El Editor Atto ha sido personalizado para facilitar la inserción y embebido de contenido multimedia accesible, siguiendo los principios del Diseño Universal para el Aprendizaje (DUA). Esta personalización permite crear contenido más inclusivo y accesible, utilizando múltiples formas de representación tanto para la presentación de contenido como resolución de actividades por parte de los estudiantes.</p> <p>Los cuestionarios interactivos y las actividades H5P son herramientas valiosas que permiten la creación de ejercicios autocorrectivos con retroalimentación inmediata y mejorar sus habilidades en tiempo real, con, por ejemplo, tipo de pregunta llenar espacio en blanco y arrastar y soltar para la práctica de escritura de código o programación con bloques sin salir del entorno.</p> <p>El plugin Tablero (Board) proporciona un espacio de colaboración virtual en forma de tablero de notas adhesivas, donde los estudiantes pueden compartir ideas y discutir proyectos. Esta herramienta fomenta la interacción y el trabajo en equipo, elementos cruciales en la enseñanza de programación y robótica. Al colaborar en un entorno virtual, los estudiantes pueden intercambiar ideas, resolver problemas conjuntamente y aprender unos de otros, emulando un entorno real de desarrollo de software.</p> <p>El módulo Taller (Workshop) en Moodle facilita la evaluación entre pares, permitiendo a los estudiantes enviar sus trabajos y evaluarlos mutuamente. Esta metodología de evaluación promueve la crítica constructiva y el aprendizaje colaborativo. En un curso de programación y robótica, este enfoque es particularmente útil para enseñar a los estudiantes a revisar y mejorar su trabajo en base a la retroalimentación recibida, mejorando así sus habilidades tanto en programación como en la evaluación crítica de código.</p>

Verificación de la Accesibilidad	<p>El Brickfield Accessibility Toolkit mejora la accesibilidad del contenido en Moodle. Evalúa el contenido del curso, identifica problemas de accesibilidad y ofrece soluciones para corregirlos. Esto asegura que todos los materiales sean accesibles para todos los estudiantes, cumpliendo con los estándares de accesibilidad y proporcionando un entorno inclusivo.</p>
	<p>El Accessibility Block permite a los usuarios personalizar la visualización de Moodle según sus necesidades visuales. Permite cambiar el tamaño del texto, ajustar el contraste y los colores, y guarda estas configuraciones para cada usuario. Esto mejora la legibilidad y hace que Moodle sea más accesible para todos.</p>
Múltiples Formatos de Presentación de Contenidos	<p>Recurso Libro (Book): Permite crear materiales estructurados en capítulos y subcapítulos, facilitando la navegación y la incorporación de multimedia accesible, como videos con subtítulos y transcripciones.</p>
	<p>Recurso Lección (Lesson): Ofrece un aprendizaje interactivo y personalizado a través de páginas con contenido, preguntas y rutas adaptativas, permitiendo a los estudiantes avanzar a su propio ritmo y recibir retroalimentación individualizada.</p>
	<p>Página de Contenido (Content Page): Permite segmentar el contenido en secciones manejables y crear rutas de aprendizaje específicas, lo cual puede ser útil para estudiantes con dificultades de atención o que necesitan un enfoque más estructurado.</p>
	<p>En LMS soporta diversos soportes multimedia y permite accesibilizarlos:</p>
	<p>Audio: Se proporcionaron transcripciones completas, subtítulos precisos.</p>
	<p>Video: Se incluyen subtítulos, descripciones auditivas, se aseguró un contraste y tamaño de texto adecuados, y se proporcionaron transcripciones para aquellos que no pueden acceder al contenido visual o auditivo.</p>
	<p>Imágenes: Se agrega texto alternativo descriptivo y se utilizaron colores y contrastes adecuados para asegurar que las imágenes sean comprensibles para personas con daltonismo o baja visión.</p>
	<p>Slides: Se utilizan fuentes claras, tamaños de letra adecuados, estructuras coherentes y descripciones verbales para contenido visual complejo.</p>
	<p>Documentos hipertextuales: Se incluye hipervínculos descriptivos, estructuras claras y encabezados jerarquizados para facilitar la navegación y la comprensión del contenido.</p>
	<p>Los contenidos tienen una versión descargable y portable en pdf para quienes no disponen de dispositivos y conectividad de forma permanente.</p>
	<p>eXeLearning herramienta de código abierto para crear contenidos accesibles, adaptables y personalizados, siguiendo los principios del DUA. Su capacidad de acceso sin conexión y portabilidad la hacen ideal para contextos con limitaciones económicas.</p>
Extensiones Recomendadas de Chrome	<p>Hiper Bold: Esta extensión mejora la legibilidad del texto resaltando en negrita la primera mitad de las palabras, lo cual puede ser beneficioso para estudiantes con dislexia.</p>
	<p>Natural Reader: Esta herramienta convierte texto en línea en audio con voz natural, ofreciendo subtítulos y modos de lectura adaptables, lo cual puede ser útil para estudiantes con dislexia, baja visión o dificultades de aprendizaje.</p>

	Mercury Reader: Simplifica el diseño de páginas web, eliminando distracciones y mejorando la legibilidad, lo cual puede ser beneficioso para estudiantes con TDAH o dificultades de atención.
	Conversor PDF en línea con Reconocimiento Óptico de Caracteres (OCR): Este conversor en línea permite convertir documentos PDF en formatos accesibles, asegurando que el contenido sea legible por dispositivos de asistencia y compatible con tecnologías de lectura de pantalla.

(Tabla 2. Herramientas, extensiones y acciones para la Accesibilidad en Moodle. Elaboración propia.)

## Análisis de compatibilidad de Herramientas, extensiones y acciones con DUA y Accesibilidad.

Las herramientas, extensiones y acciones posibles en LMS encuentran compatibilidad con las Pautas de Accesibilidad Web (WCAG) 2.1, que establecen cuatro principios clave (perceptible, operable, comprensible y robusto) y tres niveles de conformidad (A, AA y AAA) con los principios DUA, la accesibilidad cognitiva y accesibilidad en la información y comunicación.

- Redirecciónamiento al Área Personal: Esta acción cumple con el principio de operabilidad de las WCAG 2.1 (nivel A) y con el principio de acción y expresión del DUA al facilitar la navegación y el acceso directo a los recursos principales del curso.
- Etiquetas con Pictogramas: El uso de etiquetas con pictogramas como hipervínculos cumple con el principio de perceptibilidad de las WCAG 2.1 (nivel A) y con el principio de representación múltiple del DUA al proporcionar información visual clara y comprensible, especialmente para estudiantes con dificultades de lectura o comprensión.
- Configuración Avanzada de Apariencia: La personalización de temas (Snap, Fordson y Moove) para garantizar un contraste adecuado, tipografías legibles y diseños simples cumple con el principio de perceptibilidad de las WCAG 2.1 (nivel AA) y con el principio de representación múltiple del DUA, facilitando la navegación y el acceso a los estudiantes con diversas capacidades visuales.
- Extensiones "Enrol Notification" y "Resource Notification": Estas extensiones cumplen con el principio de perceptibilidad de las WCAG 2.1 (nivel A) y con el principio de implicación del DUA al proporcionar notificaciones claras y oportunas sobre actualizaciones y cambios en el curso, siendo especialmente útil para estudiantes con dificultades de atención o memoria.
- Integración de Google Calendar y Bloque Calendario: La integración de estas herramientas cumple con el principio de comprensibilidad de las WCAG 2.1 (nivel A) y con el principio de representación múltiple del DUA, proporcionando una visualización clara y estructurada de los eventos y actividades del curso.
- Insignias (Badges): El uso de insignias cumple con el principio de operabilidad de las WCAG 2.1 (nivel A) y con el principio de implicación del DUA, al proporcionar una forma interactiva y visual de reconocer los logros de los estudiantes, fomentando la motivación y el compromiso.
- Foros Diferenciados: La diferenciación de foros cumple con el principio de comprensibilidad de las WCAG 2.1 (nivel A) y con el principio de compromiso del DUA, al organizar la comunicación de manera clara y estructurada, facilitando la participación y el intercambio de información.
- Etiqueta con Modal para Grabaciones: Esta función cumple con el principio de operabilidad de las WCAG 2.1 (nivel A) y con el principio de acción y expresión del DUA al proporcionar una forma accesible de visualizar videos sin salir del entorno de aprendizaje.
- Paquete de Idioma: La implementación de esta función cumple con el principio de comprensibilidad de las WCAG 2.1 (nivel AA) y con el principio de representación múltiple del DUA al permitir que los estudiantes accedan al contenido en su idioma preferido, eliminando barreras lingüísticas.
- Atto Syntax Highlighter: Este plugin cumple con el principio de perceptibilidad de las WCAG 2.1 (nivel A) y con el principio de representación múltiple del DUA al resaltar la sintaxis del código, facilitando su lectura y comprensión, especialmente para estudiantes con dificultades visuales o de aprendizaje.

- VPL: El Virtual Programming Lab cumple con los principios de operabilidad y comprensibilidad de las WCAG 2.1 (nivel A) y con los principios de acción y expresión, y de compromiso del DUA al proporcionar un entorno controlado para escribir y probar código, con retroalimentación inmediata.
- CodeRunner: Este plugin cumple con los principios de operabilidad y comprensibilidad de las WCAG 2.1 (nivel A) y con los principios de acción y expresión, y de compromiso del DUA al permitir la ejecución y evaluación automática de código, fomentando la práctica y el aprendizaje autónomo.
- Editor Atto Personalizado: La personalización del Editor Atto cumple con el principio de operabilidad de las WCAG 2.1 (nivel A) y los principios de acción y expresión, y de representación múltiple del DUA al facilitar la inserción y embebido de contenido multimedia accesible.
- Cuestionarios Interactivos y Actividades H5P: Estas herramientas cumplen con los principios de operabilidad y comprensibilidad de las WCAG 2.1 (nivel A) y los principios de acción y expresión y de compromiso del DUA al proporcionar ejercicios autocorrectivos con retroalimentación inmediata, fomentando la práctica y el aprendizaje autónomo.
- Plugin Tablero (Board): Este plugin cumple con el principio de interacción de las WCAG 2.1 (nivel AA) y el principio de compromiso del DUA al fomentar la colaboración y el trabajo en equipo en un entorno virtual.
- Módulo Taller (Workshop): Este módulo cumple con el principio de implicación de las WCAG 2.1 (nivel AA) y el principio de compromiso del DUA al promover la evaluación entre pares y el aprendizaje colaborativo.
- Brickfield Accessibility Toolkit: Esta herramienta cumple con el principio de robustez de las WCAG 2.1 (nivel AA) y el principio de acción y expresión del DUA al evaluar y mejorar la accesibilidad del contenido en Moodle.
- Accessibility Block: Este bloque cumple con el principio de adaptabilidad de las WCAG 2.1 (nivel AA) y el principio de representación múltiple del DUA al permitir a los usuarios personalizar la visualización de Moodle según sus necesidades visuales.
- Recursos Libro, Lección y Página de Contenido: Estos recursos cumplen con el principio de representación múltiple de las WCAG 2.1 (nivel A) y el DUA al ofrecer diferentes formas de presentar la información.
- Soporte para Diversos Soportes Multimedia: El soporte para diversos formatos multimedia cumple con el principio de representación múltiple de las WCAG 2.1 (nivel A) y el DUA al permitir la inclusión de audio, video, imágenes, etc., enriqueciendo la experiencia de aprendizaje.
- eXeLearning: Esta herramienta cumple con el principio de accesibilidad de las WCAG 2.1 (nivel A) y el principio de representación múltiple del DUA al permitir la creación de contenidos accesibles, adaptables y personalizados, incluso en contextos con limitaciones económicas o de conectividad.
- Hiper Bold, Natural Reader y Mercury Reader: Estas extensiones cumplen con el principio de perceptibilidad de las WCAG 2.1 (nivel AA) y el principio de representación múltiple del DUA al mejorar la legibilidad y la comprensión del contenido, especialmente para estudiantes con dislexia, baja visión o dificultades de atención.
- Conversor PDF en línea con OCR: Este conversor cumple con el principio de robustez de las WCAG 2.1 (nivel AA) y el principio de acción y expresión del DUA al permitir la conversión de documentos PDF en formatos accesibles.

## Conclusiones

Este estudio exploratorio, centrado en la inclusión educativa en entornos virtuales de aprendizaje (EVEA) para personas con diversidad funcional, ha desvelado un panorama complejo de desafíos y oportunidades en el campo de la enseñanza de la programación y la robótica. A pesar de los avances en la implementación de herramientas y estrategias basadas en el Diseño Universal para el Aprendizaje (DUA) y la accesibilidad web, cognitiva, tecnológica y de la información y comunicación, persisten barreras significativas. Es crucial reconocer que la diversidad funcional es amplia y compleja, abarcando más allá de las discapacidades sensoriales, motrices y algunas dificultades de aprendizaje abordadas en este estudio. Investigaciones futuras deberían incluir discapacidades intelectuales, psicosociales y otras condiciones que pueden afectar la experiencia de aprendizaje en línea, reconociendo la interseccionalidad de la discapacidad con factores socioeconómicos, género, edad y

origen étnico. La discriminación múltiple o interseccional puede agravar las barreras, por lo que comprender estas interacciones es esencial para diseñar soluciones inclusivas.

Otra consideración es que evaluar rigurosamente el impacto de las estrategias de accesibilidad e inclusión implementadas es otro desafío clave. Si bien este estudio identificó herramientas y adaptaciones prometedoras, se necesitan investigaciones más exhaustivas para medir su efectividad real en participación, motivación, rendimiento académico y satisfacción de los estudiantes con diversidad funcional. La evaluación continua y la retroalimentación de los usuarios son fundamentales para refinar y mejorar estas estrategias a lo largo del tiempo.

A pesar de estos desafíos, este estudio también ha revelado oportunidades significativas. Las tecnologías digitales ofrecen un enorme potencial para personalizar el aprendizaje, proporcionar herramientas de apoyo y crear contenidos accesibles en múltiples formatos, rompiendo barreras y abriendo nuevas posibilidades para estudiantes con diversidad funcional. La enseñanza de la programación y la robótica en edades tempranas es un área de creciente importancia en el mundo actual, ya que estas habilidades son cada vez más demandadas en el mercado laboral y son fundamentales para el desarrollo de la creatividad, el pensamiento crítico y la resolución de problemas. Garantizar que esta educación sea inclusiva desde el principio es esencial para construir un futuro equitativo y próspero para todos.

La formación tanto de agentes decisores como de docentes en DUA, inclusión educativa, accesibilidad cognitiva, de la información y comunicación es esencial para crear entornos de aprendizaje inclusivos en este campo. La capacitación continua y el apoyo institucional son cruciales para fomentar una cultura de inclusión en la educación en línea. Es imperativo que las instituciones educativas inviertan en la formación y desarrollo profesional de su personal docente y administrativo, asegurando que estén capacitados para implementar prácticas inclusivas y utilizar las herramientas tecnológicas de manera efectiva. Esto incluye tanto la formación técnica como la sensibilización sobre la diversidad funcional y la creación de un ambiente de aprendizaje acogedor y respetuoso.

En conclusión, este estudio sienta las bases para futuros trabajos de investigaciones y acciones en inclusión educativa en entornos virtuales que se espera poder realizar en adelante, particularmente en el campo de la programación y la robótica, que es cada vez más relevante en la educación y en el mundo laboral. Se advierte que el ciclo de invisibilidad y las barreras del aprendizaje y participación se encuentran muy presentes. Abordar los desafíos identificados y aprovechar las oportunidades emergentes nos permitirá avanzar hacia EVEA verdaderamente inclusivos, equitativos y accesibles para todos los estudiantes, independientemente de sus capacidades. Este es un compromiso continuo que requiere la colaboración de investigadores, educadores, diseñadores instruccionales, desarrolladores de tecnología y, sobre todo, de las personas con diversidad funcional, cuyas voces y experiencias deben ser centrales en este proceso. La inclusión educativa no es solo un derecho humano fundamental, sino también un imperativo ético y social para construir un futuro más justo y equitativo para todos, donde todas las personas tengan la oportunidad de desarrollar su potencial y contribuir plenamente a la sociedad.

## Referencias bibliográficas

- [1] Organización de las Naciones Unidas. (2006). *Convención sobre los Derechos de las Personas con Discapacidad*.
- [2] Ley Nacional de Educación 26.206 (2006).
- [3] Ley Nacional 26.378 (2008).
- [4] UNESCO. (2015). *Declaración de Nueva Delhi: Educación 2030*.
- [5] Organización de las Naciones Unidas. (2015). *Transformar nuestro mundo: la Agenda 2030 para el Desarrollo Sostenible*.
- [6] Palacios, A. (2008). *El modelo social de discapacidad: orígenes, caracterización y plasmación en la Convención Internacional sobre los Derechos de las Personas con Discapacidad*. Madrid: CINCA.

- [7] Mareño Sempertegui, N. Y., & Torrez, M. P. C. (2013). Educación inclusiva y barreras para el aprendizaje y la participación. *Revista Latinoamericana de Educación Inclusiva*, 7(2), 115-130.
- [8] Gómez, N. A., & Tavarez, G. E. O. (2018). La interculturalidad y la discapacidad en entornos universitarios. *Reencuentro. Análisis de Problemas Universitarios*, 29(75), 133-154.
- [9] Calvo Buezas, T. (2001). Género y discapacidad. Dos realidades, una discriminación múltiple. *Siglo Cero*, 32(3), 39-49.
- [10] Díaz Velázquez, P., García Leos, J. L., & Hernández Leos, G. (2011). Interseccionalidad: un enfoque para el estudio de las desigualdades sociales. *Convergencia. Revista de Ciencias Sociales*, 18(55), 189-214.
- [11] Werneck, C. (2005). *Manual sobre Desarrollo Inclusivo para los Medios y Profesionales de la Comunicación*. Escola Da Gente- WVA Editora, Río de Janeiro
- [12] Brioli, C., Amaro, R., & García, I. (2011). Referente teórico y metodológico para el diseño instruccional de entornos virtuales de enseñanza y aprendizaje (EVEA). *Docencia universitaria*, 12(2).
- [13] Holst, P. (2006). La accesibilidad a la información y a la comunicación de las personas con discapacidad en el marco de los derechos humanos. *Revista Española de Discapacidad*, 4(1), 11-34.
- [14] Brusilovsky, P. (2016). Accesibilidad cognitiva en entornos físicos y virtuales. En P. A. Flores (Ed.), *Tecnologías para la inclusión* (pp. 21-58). Universidad de Salamanca.
- [15] Initiative for Web Accessibility (WAI). (2019, 8 de abril). Novedades en las Pautas de Accesibilidad para el Contenido Web (WCAG) 2.1. W3C. <https://www.w3.org/WAI/standards-guidelines/wcag/new-in-21/es>
- [16] Pastor, C. A., Sánchez, J. M., & Zubillaga, A. (2014). Diseño Universal para el aprendizaje (DUA). Recuperado de [https://www.educadua.es/doc/dua/dua\\_pautas\\_intro\\_cv.pdf](https://www.educadua.es/doc/dua/dua_pautas_intro_cv.pdf)
- [17] Ministerio de Educación de la Nación. (2017). Competencias de Educación Digital (1a ed.). Ministerio de Educación de la Nación.
- [18] Costa, R., Araújo, C., Neto, A. C., Cunha, A., & Henriques, P. R. (2023). Computational Thinking, Visual Impairment and Melodic. IE Comunicaciones: Revista Iberoamericana de Informática Educativa, (37), 1-13.
- [19] De Miguel, R. (s. f.). Scratch Jr Tactile: así es el proyecto que enseña pensamiento computacional al alumnado con discapacidad. Educación 3.0. <https://www.educaciontrespuntocero.com/tecnologia/scratch-jr-tactile/>
- [20] Escribano, C. L., & Sánchez-Montoya, R. (2012). Scratch y necesidades educativas especiales: Programación para todos. *Revista de educación a distancia (RED)*, (34).
- [21] Fresneda, V. & Mendoza, E. (2005). *Dificultades de aprendizaje e intervención psicopedagógica*. Vol. I: Conceptos, evaluación y tratamiento.
- [22] Mendoza, E. (2016). *Dificultades de aprendizaje e intervención psicopedagógica: Vol. II: Lectura, escritura y matemáticas*.
- [23] Juárez, A., & Pérez, J. (2004). *Dislexia, disgrafía, disortografía. Evaluación y tratamiento*.
- [24] Orjales Villar (2000). Déficit de atención con hiperactividad: el modelo híbrido de las funciones ejecutivas de Barkley. *Revista complutense de educación*, 11(1), 71-84.
- [25] Parra, D., & Luque-Rojas, M. J. (2013). Discalculia. *Revista de Humanidades y Ciencias Sociales*, (18), 181-202.

- [26] Puente González, A. (2012). *Dificultades de aprendizaje escolar*. Psicología Pirámide.
- [27] Ramírez Morera, M. (2010). Dificultades de aprendizaje: unificación de criterios diagnósticos. *Revista de neurología*, 50(Supl 3), S25-S30.
- [28] Moodle Docs. (s.f.). *Accessibility*. Recuperado de <https://docs.moodle.org/dev/Accessibility>

Versión Preliminar

# **Mediación Tecnopedagógica en la formación de alumnos de Primer Año de Ingeniería implementada en un modelo Bimodal**

Artemisa Trigueros<sup>1</sup>, Mabel Compagnoni<sup>1</sup>, Larisa Toro<sup>1</sup>, Sabrina Gómez<sup>1</sup>

<sup>1</sup>Universidad Nacional de La Matanza, Florencio Varela 1903,

B1754 San Justo, Pcia. De Buenos Aires, Argentina

[artemisa , mcompagnoni, larisa, sgomez @unlam.edu.ar](mailto:artemisa , mcompagnoni, larisa, sgomez @unlam.edu.ar)

## **Resumen**

Las Tecnologías de la Información y la Comunicación avanzan en forma vertiginosa, incorporando e involucrando a la mayoría de las actividades humanas, desde las sociales, culturales, políticas, económicas, deportivas y educativas. En referencia a la educación, dicho avance proporciona una excelente y necesaria oportunidad para los sistemas educativos y sus instituciones, de evaluar, reformular, actualizar, revisar e innovar los diseños del proceso de enseñanza – aprendizaje, tendientes a una nueva planificación que incluya mediaciones tecnológicas dentro de su diseño. El Aislamiento Social Obligatorio, implementado durante la pandemia de COVID, durante 2020 y 2021, permitió comenzar a implementar distintas herramientas tecnológicas en las cátedras universitarias. El presente trabajo aborda las innovaciones de mediación tecnopedagógicas Bimodales (Presencial y Virtual) en los procesos de enseñanza, incorporadas por la Universidad Nacional de La Matanza aplicadas durante el primer cuatrimestre del año 2022, en la formación de alumnos de Primer Año implementadas en un modelo Bimodal de la cátedra Fundamentos de TIC's<sup>1</sup>, perteneciente al plan de estudio de las cinco carreras de Ingeniería de la UNLaM<sup>2</sup>. Este trabajo fue presentado en CONAIISI 2022.

**Palabras clave:** Informática, Tecnopédagogía, Presencialidad, Virtualidad, Bimodalidad.

## **1. Introducción**

Existe consenso al afirmar que las TIC en educación ofrecen una oportunidad para enriquecer los aprendizajes de calidad. De esto da cuenta el Marco de Acción para la Educación 2030, el plan acordado a nivel internacional para implementar el Objetivo de Desarrollo Sostenible Número 4, cuando solicita a los países que aprovechen las Tecnologías de la Información y la Comunicación (TIC) para “reforzar el aprendizaje efectivo y de calidad”. En este sentido, las instituciones educativas también afrontan la oportunidad de constituirse en un espacio que favorezca la reducción de brechas digitales y promuevan la democratización del conocimiento, haciendo accesible el conocimiento a todos y todas y poniendo un énfasis importante en los procesos de inclusión. [1]

Autores como [2, 3] señalan que “innovación” es un concepto que implica cambio, pero aclaran que no todo cambio significa una innovación. El cambio en sí mismo puede ser espontáneo, mientras que la innovación tiene una intencionalidad intrínseca. Así también, en la innovación, el cambio es deseado y producto de un proceso. Por lo tanto, la innovación consiste en un cambio consciente, producto de un proceso y que se realiza dentro de un contexto determinado.

Para hablar de innovación en educación es necesario atreverse a repensar la tarea educativa. El cómo se enseña no es solo qué procedimientos novedosos se pueden incorporar. Es repensar, reestructurar y redefinir el triángulo didáctico para dar lugar a otros modos de organizar la enseñanza. [1]

<sup>1</sup> Fundamentos de TICs es el nombre de la materia

<sup>2</sup> Universidad Nacional de La Matanza

La innovación tecnopedagógica en la educación universitaria consiste en un cambio consciente de las prácticas pedagógicas por parte de los docentes, producto de estrategias emergentes con apropiación de la tecnología dentro de un contexto y tiempo determinado. [4]

Es deseable entonces, desarrollar un enfoque sistémico [5] que tome en cuenta una diversidad de variables contempladas en el proceso de enseñanza aprendizaje, generando una integración entre lo tecnológico, pedagógico y el contexto en el que se realiza. En este punto, es que cabe hablar de tecnopedagogía, dado que ésta busca entender la integración de la tecnología digital en las prácticas pedagógicas dentro de los entornos de enseñanza-aprendizaje. [6]

El desarrollo de procesos colectivos de aprendizaje, la extensión de las experiencias educativas más allá de las aulas, la creación de comunidades educativas diversas, el fortalecimiento de las modalidades colaborativas y horizontales para la construcción del conocimiento, la accesibilidad a materiales didácticos en diversos formatos, la posibilidad de renovar las formas de evaluación, entre otros, son elementos que el denominado aprendizaje móvil aporta al diseño de nuevos modelos educativos. [7, 8]

Asimismo, el mercado laboral exige competencias –denominadas Competencias del Siglo XXI, Competencias digitales, Competencias de orden superior– que, al ser soportadas por la tecnología, sólo pueden ser desarrolladas apropiadamente con ella. [1]

Cabe destacar, que las estrategias de integración de las TIC en educación apuntan a promover procesos de enseñanza y de aprendizaje orientados al objetivo de “aprender a aprender” a lo largo de la vida. [1]

## 2. Contexto

La Resolución Ministerial N° 2641/2017 regula la educación a distancia en las instituciones de educación superior del país. En dicha resolución se aprueba el documento sobre la opción pedagógica y didáctica de educación a distancia propuesto por el consejo de universidades y también se establece la organización, exigida por la Comisión Nacional de Evaluación y Acreditación Universitaria (CONEAU) a los establecimientos universitarios, para impartir carreras asociadas a los Sistemas Institucionales de Educación a Distancia (SIED).

Asimismo, en el documento de consideraciones sobre las estrategias de hibridación frente al inicio del ciclo lectivo 2022 de la CONEAU [9] consta que después de las prácticas de emergencia experimentadas en los últimos tiempos una amplia mayoría de los docentes y de los estudiantes han naturalizado el tránsito de la educación presencial a la virtual.

Con base en lo anteriormente dicho es que, luego de los años 2020 y 2021 en los cuales las instituciones universitarias debieron adoptar la Educación Remota de Emergencia en el marco de las medidas sanitarias que impuso la pandemia, muchas universidades tomaron la decisión de incorporar o incrementar su oferta de educación a distancia. En la Universidad Nacional de La Matanza se plantearon estrategias que combinen tanto la educación virtual con la presencial.

## 3. Materiales y Métodos

El ingreso de las nuevas Tecnologías de la Información y Comunicación (TIC) en la Educación Superior y, paulatinamente, en la universitaria, genera transformaciones que impactan tanto en los procesos de enseñanza como en las formas organizacionales y en la conformación política de las universidades [10].

El modelo bimodal es el que se desarrolla en una universidad convencional, presencial, como cualquiera de las que tenemos en nuestro país, y que decide enseñar, también, a distancia. Para eso, tiene que adquirir una configuración especial, que permita en su interior el desarrollo armónico de ambas modalidades. Eso es, en esencia, lo que significa la Bimodalidad. [11]

Ante la nueva perspectiva de confeccionar un diseño tecnopedagógico de enseñanza mediada, como es el caso del diseño de ambientes y situaciones de aprendizaje en las plataformas MiEL (Campus Virtual propio de la UNLaM)<sup>3</sup> y Microsoft Teams<sup>4</sup> brindadas por la Universidad Nacional de La Matanza, se consideraron la naturaleza y el tipo de actividades que los alumnos realizarían de forma no solo individual sino colectiva; las características de las herramientas que se pusieron a disposición del alumnado; los usos previstos de dichas herramientas; y las reglas y prácticas educativo-culturales en las

<sup>3</sup> MiEL: Materias Interactivas en Línea.

<sup>4</sup> Microsoft Teams es una plataforma unificada que permite la colaboración entre personas de un mismo equipo, compartiendo recursos y cuya función principal es la comunicación constante entre los miembros del equipo.

que se insertarían las actividades propuestas. Asimismo, se reflexionó respecto a las potencialidades de los instrumentos pedagógicos sincrónicos y asincrónicos de ambas plataformas.

De esta forma, al trabajar la integración de TIC en la educación se hace imprescindible redefinir los objetivos educativos en tanto búsqueda de sentido, desarrollar nuevas formas de enseñar y aprender, modificar contenidos y prácticas, reformular las propuestas de formación docente y cambiar los estilos de enseñanza, potenciar las redes y abrir nuevos espacios y modalidades de interacción e intercambio para lograr aprendizajes significativos y de calidad para todos. [1]

Esto supone la coexistencia de diferentes modelos de integración TIC que apunten al potencial y riqueza pedagógica de una ecología de dispositivos donde las tecnologías se utilicen no sólo como herramientas sino y sobre todo como un nuevo ecosistema para el aprendizaje. [1]

Proponer actividades con distintos formatos y lenguajes, mejorar la comunicación, desarrollar la autonomía y la creatividad, son parte del nuevo diseño educativo, que no se sostiene sin la entrada del entorno digital. Para ello, es imprescindible integrar en el currículum nuevos componentes y mejorar la relevancia de los contenidos que se enseñan con información de calidad y mejores conexiones con lo contemporáneo. [1]

Es decir, el desarrollo de la innovación tecnopedagógica en un ámbito universitario y desde una mirada sistémica requiere entender el rol docente más allá de la transmisión de contenidos científicos, considerando estrategias actuales que faciliten la acción formativa, la reflexión, la creación, el debate, el aprendizaje colaborativo, la metacognición, etc. [4]

De acuerdo a estas estrategias y teniendo en cuenta lo descrito por [12] en su propuesta de buenas prácticas de educación mediada en el contexto universitario, Fundamentos de TICs desarrolló la planificación del diseño curricular de la materia.

En el 1º Cuatrimestre 2022, se implementó la Bimodalidad en Fundamentos de TICs a través de una oferta, optativa para el alumno, de siete cursos presenciales y un curso virtual.

A continuación, se detallan las herramientas de mediación tecnopedagógica que se aplican en la propuesta didáctica del modelo bimodal abordado en el presente trabajo, implementadas en la materia Fundamentos de TICs, correspondientes al Primer Año de las cinco carreras de Ingeniería de la Universidad Nacional de la Matanza.

## 4 Innovaciones tecnopedagógicas implementadas

Las herramientas aplicadas en el desarrollo del Curso Bimodal se organizaron en 4 momentos de acuerdo a las prácticas pedagógicas identificadas en el trabajo de [11]

- Antes de iniciar el Curso
- Al inicio del Curso
- Durante la Ejecución del Curso
- Al finalizar el Curso

### 4.1 Antes de iniciar el Curso

Se desarrolló un currículum en el que se incorporó el diseño didáctico de ambas modalidades, teniendo en cuenta los objetivos y contenidos de la materia, tanto para el dictado presencial como para el virtual.

En la propuesta pedagógica se utiliza MIEl como campus virtual para ambas modalidades y Microsoft Teams como plataforma para los encuentros sincrónicos por videoconferencia de los cursos virtuales.

Entre las funcionalidades de ambas plataformas se destaca que en la plataforma MIEl es donde se gestionan los foros y se establece el primer contacto con los alumnos de los cursos virtuales ya que, luego de inscribirse, tienen acceso en forma automática a las materias que cursarán; dicha plataforma también se utiliza como repositorio de contenidos para las dos modalidades.

En tanto que Microsoft Teams, además de proveer un espacio para los encuentros sincrónicos por videoconferencia de los cursos virtuales, permite la creación de equipos y de canales privados que propician el trabajo grupal colaborativo con supervisión del docente.

La Figura 1 muestra la identificación del equipo correspondiente al Curso 02 Virtual, en la Plataforma Microsoft Teams.



Figura 1- Identificación del curso virtual en Microsoft Teams.

Debido a que los cursos presenciales y el curso virtual comparten los mismos contenidos, material teórico y trabajos prácticos, así como, el mismo espacio en el repositorio de materiales curriculares dentro del Campus Virtual MieL, se incluyeron módulos comunes a ambas modalidades en los apartados de PRESENTACIÓN DE LA MATERIA y materiales de estudio de cada Unidad Temática, y módulos diferenciados para cada modalidad en cuanto a su METODOLOGÍA Y CRONOGRAMA.

La Figura 2, permite observar los contenidos comunes y los contenidos diferenciados para los cursos presenciales y el curso virtual.

Figura 2. Módulos para Cursos Presenciales y Curso Virtual.

Asimismo, se implementaron Foros en MieL para llevar a cabo las Clases de Consulta. En estos foros pueden participar los alumnos de ambas modalidades, en forma Virtual y Asincrónica. La Figura 3 permite conocer los horarios y los docentes a cargo de las Clases de Consulta (se han borrado los nombres de los docentes para proteger su privacidad). Estos Foros de Consulta, permite a los estudiantes de todos los cursos, realizar sus preguntas y formular sus dudas a los docentes designados para atender los Foros. Están organizados en 6 Foros, uno por Unidad, y han resultado muy útiles para los alumnos, que, no sólo realizan sus consultas, sino que tienen acceso a las consultas de sus compañeros.

Figura 3. Foros de clases de consulta.

Asimismo, se seleccionaron, diseñaron y organizaron en forma Virtual y Asincrónica a través de MieL, los materiales o recursos didácticos interactivos, así como instrumentos de evaluación, con un propósito formativo y/o sumativo. En la Figura 4, se muestra unos de los recursos didácticos interactivos utilizados.

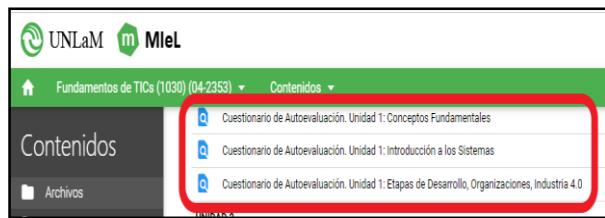


Figura 4. Cuestionarios de Autoevaluación

A través de estos cuestionarios interactivos de autoevaluación, se proporciona una retroalimentación a los alumnos, con el propósito de orientarlos en su trayectoria educativa. Dichos Cuestionarios, se implementaron para cada Unidad de la Materia.

Se implementó una estrategia innovadora de seguimiento tutorial para los alumnos de los cursos virtuales. Para facilitar su realización se aplicó la herramienta Insights que brinda Microsoft Teams, como se muestra en la Figura 5.

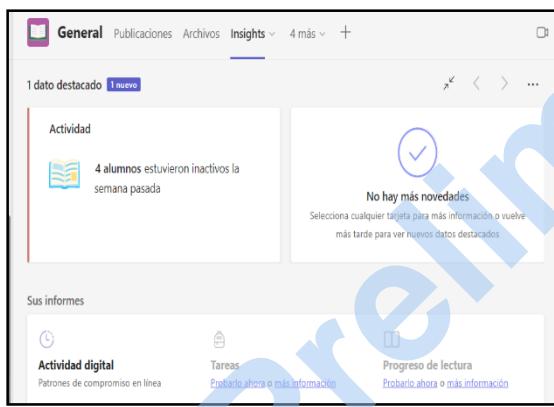


Figura 5. Insights de Microsoft Teams

La herramienta Insights, como se observa en la Figura 5, informa la cantidad de alumnos que no han estado activos durante un determinado tiempo, lo cual permitió llevar a cabo un seguimiento personalizado de cada estudiante.

## 4.2 Al inicio del Curso

Se promovió el interés de los estudiantes sobre los contenidos curriculares poniendo a su disposición material didáctico interactivo que, además de facilitar el aprendizaje de los temas abordados, proporcionan vínculos hipermediales que fomenten la búsqueda autónoma de información.

La Figura 6 muestra el material multimedia correspondiente a la Unidad 1.



Figura 6. Material interactivo utilizado en la unidad 1.

Como parte de los materiales didácticos de la catedra, se incluyeron enlaces a contenidos hipermedia de la Web con el propósito de fomentar búsquedas que expandan el alcance de los ejes temáticos que constituyen el currículum de la materia, tal como puede verse en la Figura 7.

**Lectura requerida: Unidad 4 \_ Teoría**  
Desde la página 1 a la 20 están estos temas explicados.

**Ejercitación requerida: Unidad 4 \_ Práctica**  
Esta Unidad contiene conceptos teóricos. La parte práctica, la comenzaremos a realizar a partir de la Segunda Clase. Necesitamos que comprendas más conceptos, para poder interrelacionarlos y simular el PROCESAMIENTO de una computadora.

**Material multimedia sugerido:**

- En este video podés encontrar algunas [curiosidades históricas](#) mostradas por una profesora de la cátedra.
- Acá podés ver un [ejemplo de PLC](#).

Figura 7. Links a información adicional en la Web

Se confeccionaron documentos de orientación llamados “Orientadores” para cada unidad temática con el propósito de promover en los alumnos su autonomía en la gestión del conocimiento.

La Figura 7, muestra un fragmento del Orientador correspondiente a la unidad 4, donde se incluye un enlace a un simulador de procesador (SIMUPROC) y otros vínculos que expanden los temas abordados en la unidad temática.

La Figura 8, muestra los Orientadores para cada Unidad temática, preparados especialmente para que se organicen en su trayectoria educativa los estudiantes del Curso Virtual.

**CURSO VIRTUAL**

- METODOLOGÍA DE CURSADA VIRTUAL
- CRONOGRAMA DE ACTIVIDADES VIRTUAL
- ORIENTADOR UNIDAD 1
- ORIENTADOR UNIDAD 2
- ORIENTADOR UNIDAD 3
- ORIENTADOR UNIDAD 4
- ORIENTADOR UNIDAD 5
- ORIENTADOR UNIDAD 6

Figura 8. Orientadores para el Curso Virtual

**2.2. Representación de números reales**

Al imaginar un número escrito en notación científica por ejemplo  $4,5 \times 10^3$  este número puede almacenarse de esta forma o haciendo desplazar su coma que separa la parte entera de la fraccionaria generando números equivalentes por ejemplo  $0,45 \times 10^4$ ; este efecto de mover la coma decimal o el punto que separaría la parte entera de la parte fraccionaria (por ejemplo los ingleses utilizan el punto para separador de parte entera y parte fraccionaria) se denomina en la literatura punto flotante. Actualmente para almacenar este tipo de números en notación de punto flotante se utiliza la norma IEEE 754 en sus formatos de simple y doble precisión.

**Material multimedia sugerido:**

- NotacionCientifica.ppsx
- PuntoFlotante.ppsx

**Lectura requerida: Unidad 2 – Teoría**

Desde la página 43 a la 49 estarán estos temas explicados. Allí también hay 1 ejercicio desarrollado paso a paso para aprender a utilizar la norma IEEE 754 para almacenar números y 1 ejercicio sugerido con su resolución incluida

**Ejercitación requerida: Unidad 2 \_ Práctica**

Primeramente, se recomienda realizar el Ejercicio 13 para practicar mover la coma tal como se realiza en el colegio en notación científica (Página 22), luego los ejercicios a resolver son: Los ejercicios a resolver sobre operaciones aritméticas son desde el 14 al 21 (Página 22 a 23)

Figura 9. Fragmento de un orientador

La Figura 9 expone un fragmento de uno de los Orientadores donde pueden observarse los lineamientos que facilitan la autonomía en la gestión del conocimiento.

### 4.3 Durante la Ejecución del Curso

La Figura 10 muestra parte del Orientador de la Unidad 1, confeccionado especialmente para el alumnado del Curso Virtual. En él, se pueden ver las actividades, los ejercicios y otros materiales propuestos para cada clase.

Figura 10. Organizador de la Unidad 1.

Dentro de las actividades interactivas creadas para facilitar el aprendizaje, se encuentran los mapas conceptuales de cada unidad temática, creados por los docentes de la cátedra como se ve en la Figura 11. Los mismos fueron resueltos en forma colaborativa por los alumnos, organizados en grupos dentro del aula de los cursos presenciales y mediante videoconferencia dentro de canales privados en la plataforma Microsoft Teams para el curso virtual.

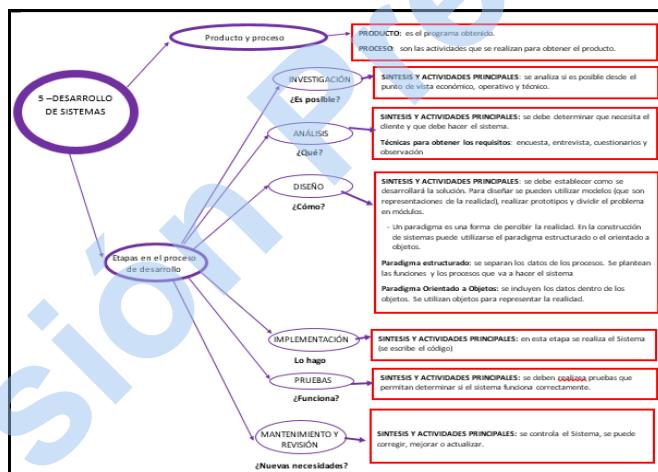


Figura 11. Fragmento del mapa conceptual propuesto.

En la Figura 12 se muestra un fragmento del enunciado de una actividad para la resolución de un mapa conceptual.

Figura 12. Fragmento para la resolución del mapa conceptual.

En la Figura 12 se visualizan las pautas para la realización del mapa conceptual brindada a los alumnos del curso virtual. Se detallan tanto los pasos para la resolución como también, las indicaciones y los tiempos para la entrega.

Durante el desarrollo del cursado de la materia, también se crearon foros en la plataforma MieL, tanto en los cursos presenciales como en el virtual, destinados a gestionar la información de cada unidad temática y foros de consultas a disposición de los alumnos para ser utilizados fuera de los horarios de clases (ver Figuras 13 y 14).

Los alumnos de los cursos virtuales debían ingresar primero a la plataforma MieL, y acceder al foro de Planificación Diaria, como muestra la Figura 13, donde se les informaba cómo se desarrollaría la clase.

The screenshot shows a forum post from a tutor named 'Tutor/a (com 02-3959)' dated 05/04/2022 at 08:28:19. The post contains the following text:

**Clase 5-4**

iBuenas tardes!

Les recordamos que, por favor, **NO ESCRIBAN EN ESTE FORO**. De esta forma, sus compañeros podrán encontrar con facilidad las actividades diarias.

Hoy nos juntaremos por **Teams a las 19hs**, para responder dudas sobre lo visto la clase anterior y realizar la parte práctica de la Unidad 1.

Luego trabajarán en forma colaborativa en grupos privados de Teams (en forma sincrónica), para resolver el primer trabajo práctico obligatorio. Las consignas de este están explicadas en el foro "Unidad 1".

Tengan siempre presente el **Orientador de la Unidad 1**, donde encontrarán con mayor detalle todos los temas vistos en cada clase.

Les dejamos el vínculo a la reunión de Teams: **CLIC ACÁ PARA LA CLASE POR TEAMS**

Saludos

Nos vemos!

Responder · Me gusta (1) · Ocultar

Figura 13. Foro de planificación diaria para curso virtual.

La Figura 14, permite observar parte de las Consultas, efectuadas en forma Virtual, correspondientes a la Unidad 3, que fue implementado antes de comenzar el cuatrimestre.

The screenshot shows a forum post from a student named 'Alumno/a (com 63-1333)' dated 06/05/2022 at 21:16:10. The post contains the following text:

Perdón por la larga pregunta:

16 En el ejercicio 2 de la unidad 3, deduce que el circuito representado en el MULTIPLE XNOR pero me dijeron que es un DECODIFICADOR. A pesar de ese error, el resultado que me dio fue el B igual que a (ya comprendí que en la parte que estoy haciendo mal)

27 eve el 31-4 cuando teníamos el XNOR, podemos deducir directamente como se escribiría o debemos hacer todos los pasos desde el 31-4

desde el 31-4 se explica algebraica para que sea válido?

Responder · Me gusta · Ocultar

Below this, another post from the same student is visible:

Hola Ezequiel

Un decodificador es un circuito combinacional de varias salidas (multifunción) que convierte la entrada binaria (de uno o más bits) en salidas correspondientes a productos cartesianos (múltiples).

Un multiplicador es un sistema que permite seleccionar una de varias entradas de datos, mediante un conjunto de líneas de control.

Un multiplexor, puede tener 2 entradas de datos para 1 terminal de control.

Pero el MAX SIEMPRE TIENE UNA SOLA SALIDA.

Creo que lo que te falló es no darte cuenta de que la figura que te muestran tiene en este caso 4 salidas.

Te recomiendo que vuelvas a mirar los videos y te apoyes en el apunte técnico para reforzar.

Cualquier otra duda podés preguntar.

Figura 14. Foros de consultas.

En la Figura 14 es posible observar la actividad de los alumnos y las respuestas de los docentes. Los alumnos de ambas modalidades tenían la posibilidad de ingresar a los foros de consultas habilitados en la plataforma MieL para todos los cursantes de la materia. En estos foros, como muestra la Figura 14, los alumnos expresaron sus dudas con las temáticas de la cátedra y los docentes respondieron a las mismas en forma detallada.

Para el alumnado de los cursos pertenecientes a la modalidad presencial además se habilitaron Foro para cada Unidad, para explicitar los materiales didácticos y las actividades propuestas que correspondían a los contenidos temáticos abordados en cada una de las clases, como puede verse en la Figura 15.

The screenshot shows a forum post from a tutor named 'Tutor/a (com 64-2656)' dated 26/05/2022 at 17:20:26. The post contains the following text:

**Tarea para el 31-05-22**

UNIDAD N°4  
INTRODUCCIÓN AL HARDWARE DE LOS SISTEMAS DE COMPUTACIÓN

- ✓ Leer la teoría de la UNIDAD n°4 hasta la página n°27.
- ✓ Resolver del TRABAJO PRACTICO N°34 -EXERCICIOS\_1 A 7.

Materiales multimedia sugeridos:

- En este video puedes encontrar algunas curiosidades históricas mostradas por una profesora de la cátedra.
- Aquí puedes ver un ejemplo de PLC.

Además se adjunta un Resumen Conceptual de la Unidad 4 para completar.

Archivos adjuntos: Resumen-Conceptual-U4.docx

Responder · Me gusta (4) · Ocultar

Below this, another post from the same tutor is visible:

Se comparte presentación utilizada durante la clase.

Archivos adjuntos: Unidad 4 PARTE 1,2.pptx

Figura 15. Foro creado para informar las actividades a realizar en un curso presencial

En ambas modalidades se diseñaron actividades con retroalimentación entre pares, propiciando espacios para que los alumnos puedan adquirir nuevos conocimientos, demostrar lo aprendido a sus compañeros y recibir la retroalimentación de ellos.

En relación con lo anterior, la Figura 16 muestra la Lista de Cotejo correspondiente al Trabajo Práctico Grupal de la Unidad 4, mediante la cual los alumnos fueron guiados en la actividad propuesta en donde se solicita aplicar el Método de retroalimentación en escalera, siendo publicada la producción de cada grupo en una pizarra virtual.

**FUNDAMENTOS DE TIC'S**

Consideraciones sobre la retroalimentación entre pares:

- ✓ Realizar una lectura comprensiva del material correspondiente al [Método de retroalimentación en escalera](#)
- ✓ Analizar el video del grupo asignado
- ✓ Confeccionar un breve párrafo incluyendo la retroalimentación a brindar, aplicando el método indicado (en al menos dos de sus aspectos como mínimo)
- ✓ Agregar en la pizarra virtual, como comentario de la publicación del video del grupo a brindar la retroalimentación, el párrafo confeccionado en el punto anterior. O bien podrán agregar una nueva publicación cuyo título sea "Retroalimentación para el Grupo X brindada por el Grupo Y"

A continuación, se observa el espacio para agregar la retroalimentación:

Figura 16. Trabajo con retroalimentación

En la Figura 17, podemos observar el muro virtual en donde es compartida la retroalimentación del trabajo práctico grupal interactivo N° 4. Los alumnos publicaron allí el video y sus compañeros de otro grupo brindaron una retroalimentación del mismo en un posteo debajo de éste.



Figura 17. Retroalimentación del Trabajo Práctico Grupal Interactivo N°4

La Figura 18 muestra la Asignación de Tareas publicadas en la plataforma Microsoft Teams, las mismas se implementaron en el curso virtual para posibilitar que los alumnos realicen autoevaluaciones de contenidos temáticos abordados en la materia, mayormente constituidas por actividades asincrónicas.

Microsoft Teams

Asignado Devuelto Borradores Todas las clases

Para ver las tareas anteriores, vaya a un equipo de clase individual.

Unidad 1 - Parte 1	1030 F de TICs VIRTUAL Curso 02 - 1C - 2022 • Vence el 1 de abril de 2022 23:59
Foro 1: Resolución de mapa conceptual	1030 F de TICs VIRTUAL Curso 02 - 1C - 2022 • Vence el 5 de abril de 2022 23:00
Foro 3: Unidad 2 -Parte B	1030 F de TICs VIRTUAL Curso 02 - 1C - 2022 • Vence el 26 de abril de 2022 23:59
Actividad 6 - Unidad 3	1030 F de TICs VIRTUAL Curso 02 - 1C - 2022 • Vence el 17 de mayo de 2022 23:59
Cierre Unidad 4	1030 F de TICs VIRTUAL Curso 02 - 1C - 2022 • Vence el 17 de junio de 2022 23:59

Figura 18. Tareas en Microsoft Teams

Algunas de las autoevaluaciones realizadas por los estudiantes, fueron implementadas a través de la herramienta formularios de Microsoft Teams, como muestra la Figura 19. Esto permitió que al finalizar la misma, los alumnos pudieran ver la resolución correcta. Además, permitió identificar cuáles eran las dificultades de los alumnos que habían realizado la actividad, con el propósito de intensificar la enseñanza de esos contenidos temáticos y promover su aprendizaje.

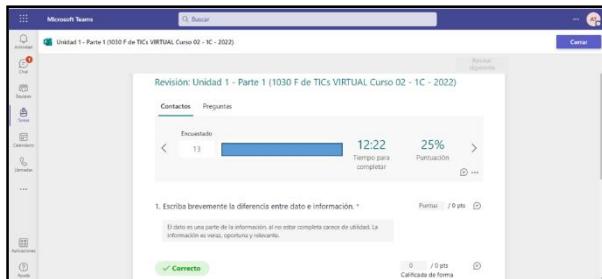


Figura 19. Tareas utilizando los formularios de Microsoft Teams.

De acuerdo con la modalidad de los cursos, los Trabajos Prácticos Grupales se realizaron en las aulas o en los canales privados de Microsoft Teams.

La Figura 20 permite observar la resolución de un Trabajo Práctico Grupal en modalidad Presencial.



Figura 20. Resolución de Trabajo Práctico Grupal Presencial.

La Figura 21 muestra en la parte izquierda, los canales privados generados en la plataforma Microsoft Teams destinados a realizar actividades prácticas grupales interactivas y sobre la derecha, parte de la actividad del Grupo 6 (para proteger su privacidad, se ocultaron los nombres de los alumnos).

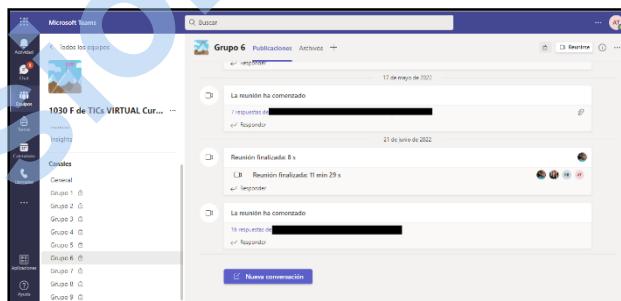


Figura 21. Intercambio de alumnos del grupo 6, en el canal privado.

En concordancia con lo manifestado en el apartado “Antes de iniciar el curso”, se realizó el seguimiento del alumnado tanto en los cursos presenciales como en aquel de cursado virtual respecto a la asistencia, entrega de trabajos, participación en clase, etc. Cabe destacar, que en el Curso con modalidad Virtual se utilizó, entre otras, la herramienta Insights, que permite realizar un seguimiento completo del curso.

En ambas modalidades, en caso de identificar alumnos con inasistencias reiteradas, se los contactó mediante mensajes de correo utilizando la herramienta de Mensajería de MIEl. Asimismo, los alumnos tenían la posibilidad de manifestar sus inquietudes por medio de la misma herramienta como se observa en la Figura 22.

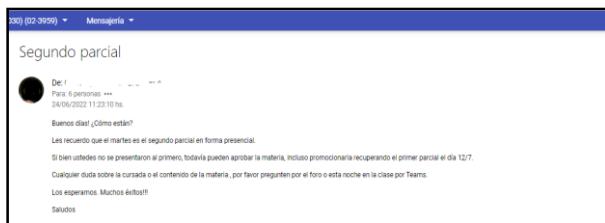


Figura 22. Mensaje de seguimiento por la plataforma MieL.

La Figura 23, muestra el seguimiento de los alumnos del Curso Virtual por medio de Insights. (Se han borrado los nombres de los alumnos para proteger su privacidad).

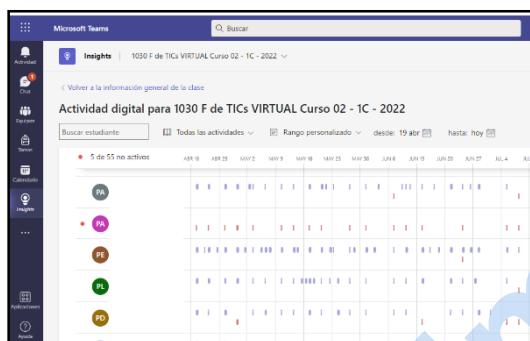


Figura 23. Seguimiento por medio de Insights.

Como estrategia innovadora de mediación pedagógica se implementó la posibilidad de acceder a espacios de Tutoría Académica a cargo de un docente de la cátedra, en forma virtual a través de videoconferencia en la plataforma Microsoft Teams, y disponibles para todo el alumnado de ambas modalidades de cursado previo a las instancias de evaluación. Estas instancias de tutoría se difundieron en los foros de cada curso en MieL, como puede verse en la Figura 24.

## Tutorías Académicas Fundamentos de TIC's



**La idea de la actividad, es fortalecer los temas de las unidades 4 y 5, como también aquellos correspondientes al recuperatorio.**

**Los talleres se realizan a través de la plataforma Teams en los días y horarios detallados a continuación, la duración de los mismos es de una hora.**

- ✓ **Miércoles 15 de junio 17 hs. taller repasando unidad 4.**
- ✓ **Miércoles 22 de junio 17 hs. taller repasando unidad 5.**
- ✓ **Jueves 7 de julio 11 hs. repaso para Recuperatorio 1er parcial.**
- ✓ **Jueves 7 de julio 12:15 hs. repaso Recuperatorio 2do parcial.**

**Aquellos que se quieran anotar deben enviar un mail a la siguiente dirección**

Figura 24. Difusión de la Tutoría Académica en la plataforma MieL.

El propósito de la Tutoría Académica brindada por la cátedra fue fortalecer las trayectorias pedagógicas del alumnado y promover el aprendizaje de las temáticas curriculares abordadas.

#### 4.4 Al finalizar el Curso

Para conocer la percepción de los alumnos sobre la materia, se realizó una encuesta de cátedra, particular de la materia Fundamentos de TICs, donde participaron alumnos de los Cursos Presenciales y el Curso Virtual en forma anónima.

La Figura 25 muestra las respuestas correspondientes a alumnos del Curso Virtual ante la pregunta: “Escribí qué mejoras o cambios proponés para mejorar la materia Fundamentos de TICs.” Referida a su cursado.

Escribí qué mejoras o cambios proponés para mejorar la materia Fundamentos de TICs.
De momento no se me ocurre nada, tuve una cursada muy agradable
Para mí no hay nada que mejorar. Fueron clases muy completas, donde en el modo Online se pudo aprender hasta mejor que en la facultad. A veces por el trabajo lo ideal es ahorrar ese tiempo de viaje y poder disfrutar de una clase como se hizo en este curso con las profesoras a cargo de la materia.
Por ahora nada. Me parece un los contenidos y las ordenes que se dan. Se adecua bien para lo que es la materia.
No cambiaría nada
Mejoras: por ahí la mejora sería a nivel de cámaras o nitidez de cámaras al pizarrón, por lo demás, estoy conforme.
Para mí está bien como me enseñaron online. No le haría cambios
Nada, excelente como está
YO QUE CURSE EN LA VIRTUALIDAD ME SENTÍ BIEN GUIADA. LOS TEMAS FUERON EXPLICADOS DESDE CERO Y LAS VECES QUE FUE NECESARIO, POR ESTO SIGAN ASÍ, TRANSMIRIENDO ESA PASIÓN POR EL CONOCIMIENTO!!

Figura 25. Encuesta de Cátedra

Como es posible observar los alumnos se manifestaron muy conformes con el dictado de la materia en su modalidad Virtual.

Asimismo, se compararon los resultados del Curso Virtual (Barras Oscuras) y los Cursos Presenciales (Barras Claras).

La Figura 26 muestra los porcentajes de Aprobados, Cursados, Reprobados y Ausentes.

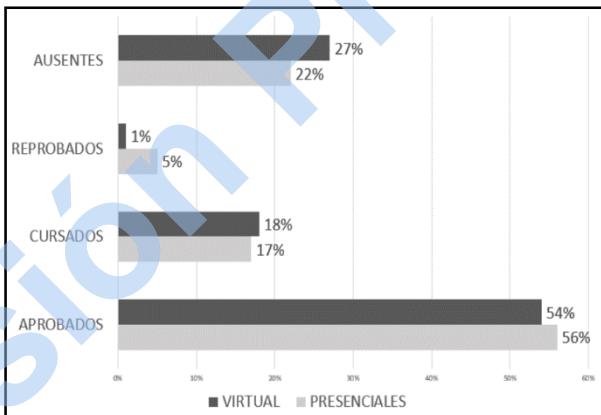


Figura 26. Comparación de resultados.

Es posible observar que los porcentajes en Aprobados y Cursados son muy parejos, presentándose la mayor disparidad en Ausentes (por cantidad de inasistencias y/o no presentarse a las evaluaciones) donde el Curso Virtual presentó mayor porcentaje y en Reprobados donde el Curso Virtual sólo obtuvo un 1%, mientras que el porcentaje de Reprobados en Cursos Presenciales, alcanzó el 5%.

### 5. Conclusiones

Las mediaciones tecnopedagógicas que se implementaron en el dictado de la materia, mediante la incorporación de herramientas de las TIC adaptadas a un diseño curricular, vigorizaron la posibilidad de generar una propuesta conjunta de formación superior, llevada a cabo a través de la plataforma Microsoft Teams y la plataforma MieL, en el sentido de que por su propia complementariedad se potenciaron mutuamente, generó de este modo un entorno abierto con múltiples intercambios comunicativos entre profesores y alumnos que suscitó una mayor profundización y apropiación del conocimiento. Los porcentajes de Aprobación y Cursada, se verificaron semejantes a los correspondientes a los cursos Presenciales. Se concluye, que las mediaciones tecnopedagógicas posibilitaron en el marco de un modelo bimodal, una trayectoria educativa exitosa y promovieron una educación de calidad en el nivel superior.

## Referencias

- [1] Lugo M. (2017) Capítulo I. La escuela y la cultura digital: el desafío por la calidad educativa con justicia social. En: Innovaciones didácticas en contexto / Walter Campi ... [et al.]; compilado por Adriana Imperatore; Marina Gergich. - 1a ed. (pp. 21-30). Bernal: Universidad Virtual de Quilmes.
- [2] Sanjinés, V. (2012). Efectos del programa innovación educativa “Escuelas de avanzada” y mejoramiento de la calidad en instituciones educativas del Callao. [Tesis para optar el grado académico de Maestro en Educación]. Universidad San Ignacio de Loyola
- [3] Mateus, J. (2014). Educación y TIC ¿quién innova a quién? Análisis de 5 experiencias en el Perú. <http://recursos.portaleducoas.org/publicaciones/educaci-n-y-tic-qui-ninnova-qui-n-lisis-de-5-experiencias-en-el>
- [4] Rivero, C. y Suárez, C. (2020). Conceptualización de la innovación Tecnopedagógica en la docencia universitaria. Recuperado en: <https://files.pucp.education/departamento/educacion/2020/11/13224634/Carol-Rivero-Conceptualizacion-de-la-innovacion-tecnopedagogica-en-la-docencia-universitaria.pdf>
- [5] Ramírez, L., y Ramírez, M. S (2018). El papel de las estrategias innovadoras en educación superior: retos en la Sociedad del conocimiento. Revista de Pedagogía 39(104), 147-170.
- [6] Shanks, J., y Young, S. (2019). Applying Attributes of Contemplative Technopедагогogy to a Social Media Assignment. Frontiers in Education, 4. <https://doi.org/10.3389/feduc.2019.00048>
- [7] Lugo, M. T. (2012), Encender el aprendizaje móvil en América Latina. Buenos Aires: UNESCO. Recuperado a partir de: <[unesdoc.unesco.org/](http://unesdoc.unesco.org/)>
- [8] UNESCO. (2013). Directrices de la UNESCO para las Políticas de Aprendizaje Móvil 2013. París: Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura.
- [9] CONEAU [Comisión Nacional de Evaluación y Acreditación Universitaria] (20 de diciembre de 2021). Consideraciones sobre las estrategias de hibridación. Disponible en: <https://www.coneau.gob.ar/coneau/wp-content/uploads/2021/12/IF-2021-123533751-APN-CONEAUME.pdf>
- [10] Villar A. (2016) Educación Superior y Entornos Virtuales. En: Bimodalidad: Articulación y Convergencia en la Educación Superior /Guillermo Tamarit ... [et al.]; compilado por Alejandro Villar. - 1a ed. -(pp. 29-32). Bernal: Universidad Virtual de Quilmes.
- [11] Mena M. (2016). Hacia la institucionalización de la Bimodalidad. En: Bimodalidad: Articulación y Convergencia en la Educación Superior /Guillermo Tamarit ... [et al.]; compilado por Alejandro Villar. - 1a ed. -(pp. 29-32). Bernal: Universidad Virtual de Quilmes.
- [12] Ruiz Bolívar, C., & Antonio Dávila, A. (2016). Propuesta de buenas prácticas de educación virtual en el contexto universitario. Revista de Educación a Distancia (RED), (49). Recuperado a partir de <https://revistas.um.es/red/article/view/257681>

# Investigación en el profesorado universitario en computación: presente y futuro.

Julia Zacharski<sup>1</sup>, Agustín Gómez Díaz<sup>2</sup>, Miguel Avalos<sup>3</sup>

<sup>1</sup> FCEQyN UNaM, Apóstoles, Argentina

<sup>2</sup> FCEQyN UNaM, Apóstoles, Argentina

<sup>3</sup> FCEQyN UNaM, Apóstoles, Argentina

[zacharskijulia@gmail.com](mailto:zacharskijulia@gmail.com) , [agustingd18@gmail.com](mailto:agustingd18@gmail.com) , [miguel.avalos@fceqyn.unam.edu.ar](mailto:miguel.avalos@fceqyn.unam.edu.ar)

## Resumen

En esta ponencia analizamos las experiencias que vinculan investigación-formación docente dentro del Profesorado Universitario en Computación (PUC) (FCEQyN - UNaM), desde un enfoque cualitativo, con entrevistas semi-estructuradas a estudiantes avanzados entre los años 2022-2024. Nuestros interrogantes a desarrollar son: ¿Qué sentidos adquiere la investigación en la formación docente de los estudiantes del PUC? ¿Cuál es su participación en espacios formales de investigación? Los registros mostraron una amplia gama de experiencias sobre la investigación, asociándolas con tareas específicas de sus cursos y materias; a pesar de que, la mayoría no estaba al tanto de los proyectos de investigación activos en la universidad. Consideramos que la investigación es un pilar de la vida universitaria, sabemos que a partir de ella no solo se puede lograr innovación y desarrollo del conocimiento, sino que también se constituye en un espacio fructífero para repensar las propias prácticas docentes tendientes a mejorarlas. En este sentido, resulta fundamental que desde la FCEQyN se tracen líneas de acción que fomenten la participación de los estudiantes del PUC en espacios de investigación, fortaleciendo así su formación docente y enriqueciendo a los equipos con sus capacidades.

**Palabras clave:** Investigación educativa; Formación docente; Experiencias; Profesorado en computación; Desarrollo profesional.

## 1. Introducción

Desde hace algunos años venimos indagando en torno a la temática de la investigación educativa, cuestión que se formalizó en enero del 2023 con el proyecto de investigación acreditado dentro de la Facultad de Ciencias Exactas Químicas y Naturales de la Universidad Nacional de Misiones (FCEQyN-UNaM) denominado: “Investigación y formación docente: experiencias de alumnos y alumnas en un profesorado universitario de la FCEQyN-UNaM” (16Q1976TI). En él nos abocamos al registro y análisis de las experiencias que vinculan investigación-formación docente dentro del profesorado universitario en computación, recuperando las voces de los estudiantes. Desde allí pudimos elaborar un diagnóstico de situación que nos permitió reconocer el campo en la actualidad, al mismo tiempo que esbozamos posibles perspectivas a futuro, como espacios de trabajo compartido entre docentes, alumnos y graduados del profesorado [1], [2].

Para nosotros como investigadores noveles fue importante transitar las distintas etapas de un proceso investigativo. Comenzamos con las lecturas bibliográficas para comprender los diferentes enfoques actuales en la temática, incluyendo la búsqueda de antecedentes y el desarrollo de un marco teórico, lo que nos permitió definir un problema. Posteriormente, avanzamos con las definiciones metodológicas, y la realización del trabajo de campo. Allí, con el análisis de bases de datos, la generación de gráficos e interpretaciones, el diseño de preguntas, la realización y análisis de entrevistas, pudimos formar un vínculo entre la complejidad del desarrollo de la investigación y lo que conocíamos por investigación dentro del campo socioeducativo.

En esta ponencia decidimos compartir algunos de los resultados obtenidos en el marco de dicha indagación cualitativa. Las principales interrogantes que responderemos en esta presentación son: ¿Qué sentidos adquiere la investigación para los estudiantes avanzados del PUC? ¿Cuál es su participación en espacios formales de investigación?

Primero, caracterizaremos a la población estudiantil del profesorado universitario en computación para facilitar un acercamiento a través de entrevistas. Posteriormente, analizaremos los sentidos y las experiencias en torno a la investigación, lo que permitirá comprender los matices que adquiere esta actividad dentro de su formación como futuros profesores. Indagar más de cerca permitirá obtener resultados precisos sobre la intervención en trabajos de investigación de los estudiantes avanzados, tratando de construir un “dibujo holístico y complejo con descripciones detalladas desde las perspectivas de los informantes” [1, p. 44].

Finalmente queremos exponer en este trabajo que, en el campo educativo la investigación es un motor de desarrollo profesional para estudiantes avanzados, graduados y educadores. Además, demostrar que es esencial llevar a cabo investigaciones en educación con el fin de profundizar conocimientos, perfeccionar estrategias y adoptar métodos innovadores que posteriormente se aplicarán en el aula. Esta práctica no solo enriquece la experiencia docente, sino que también contribuye significativamente al avance de la pedagogía.

## 2. Materiales y métodos

La investigación se basó en un enfoque de tipo cualitativo [3], con una modalidad etnográfica por medio de entrevistas semi estructuradas realizadas a ocho estudiantes avanzados del Profesorado Universitario en Computación (FCEQyN-UNaM) entre los años 2022-2024. Elegimos este enfoque etnográfico porque permite descubrir las interpretaciones de nuestro grupo de estudio mediante el trabajo de campo, destacando sus comentarios, apreciaciones y experiencias en su propio contexto. En esta investigación entendemos por “estudiantes avanzados” a aquellos alumnos que dispongan del 50% o más de las materias de la carrera aprobadas, mientras que, definimos como “espacios formales de investigación” a los proyectos y trabajos acreditados en el Sistema de Acreditación y Seguimiento de Proyectos de Investigación “SASPI”, dependiente de la Secretaría General de Ciencia y Tecnología, de la Universidad Nacional de Misiones. Los estudiantes entrevistados representan el 18% de la población total de 45 estudiantes avanzados registrados hasta el año 2022, que a pesar de no representar la totalidad de la población de estudio, sí nos brinda una aproximación inicial a sus concepciones sobre investigación, experiencias en el desarrollo de la misma y su participación en proyectos formales. Recordemos que las investigaciones de tipo cualitativas no persiguen la generación de una inferencia en términos estadísticos, sino la comprensión de sentidos sociales en el marco de experiencias contextuadas. En ese sentido, la selección de los sujetos participantes de la investigación y el tipo de muestreo, responden a una lógica más intencional que probabilística [4].

Las entrevistas realizadas se diseñaron previamente de manera colaborativa dentro del equipo. En distintas instancias pudimos consensuar aquellos tópicos que guiarían las conversaciones con nuestros interlocutores. Desde dicho instrumento indagamos, por ejemplo, si habían escuchado hablar de la investigación educativa, si conocían docentes o compañeros que estuvieran participando de esos espacios. Allí también buscábamos conocer qué sentidos adquiría la investigación en sus trayectorias educativas, qué vínculos podrían establecerse con la formación docente, y cuáles podrían ser las temáticas de su interés. Los participantes fueron seleccionados de acuerdo con los criterios mencionados anteriormente, y en la mayoría de los casos se realizó una única entrevista por persona. Todas las entrevistas fueron registradas a través de grabaciones, con el consentimiento de los entrevistados. Esto nos permitió capturar la diversidad y profundidad de sus experiencias en torno a la investigación, así como explorar las dinámicas sociales, culturales y contextuales que afectan los fenómenos analizados.

El Profesorado Universitario en Computación fue la última propuesta formativa creada en el ámbito de la carrera docente dentro de la FCEQyN, establecida en 2015. Para el año 2022, este programa contaba con un total de 411 alumnos inscritos y 26 egresados. Entre sus estudiantes, el 10,9% podría ser considerado “avanzado”, ya que había aprobado el 50% o más de las 34 materias requeridas. Más de la mitad de los estudiantes había ingresado en 2015 o 2016, lo que significa que llevaban entre 6 y 7 años en la carrera, cuyo plan de estudios tiene una duración de 4 años. Cabe destacar que casi el 80% de estos estudiantes se dedica exclusivamente a sus estudios, aunque muchos se encuentran en la mitad o un poco más del camino, con el 70% de sus materias aprobadas (es decir hasta 20 materias).

La mayoría de los estudiantes del PUC reside en Apóstoles, donde se encuentra el Módulo de Computación, que ofrece no sólo el profesorado, sino también la tecnicatura en analista de sistemas y la licenciatura en sistemas. Este contexto es relevante, ya que muchos estudiantes del PUC cursan simultáneamente las otras dos carreras. De hecho, todos nuestros entrevistados estaban inscritos en las tres propuestas al momento de las entrevistas. Aunque existen varias cátedras que forman un “tronco disciplinar común” entre las tres carreras, cada una presenta sus propias especificidades, lo que añade un nivel adicional de complejidad al analizar las trayectorias educativas de los estudiantes.

Asimismo, es importante señalar que, entre los estudiantes que trabajan, el 70% dedica más de 20 horas semanales a sus empleos, lo que limita el tiempo disponible tanto para las clases como para el estudio fuera del aula. Aunque no se dispone de datos específicos sobre la distribución de becas ni sobre el acceso a beneficios como el comedor universitario, boletos educativos o albergue, se sabe que estas ayudas existen y que muchos estudiantes se benefician de ellas.

### **3. Resultados y discusiones**

#### **3.1 La Investigación en la formación universitaria**

Podemos reconocer que la universidad es una institución compleja, no solo por la diversidad de intereses y expectativas de los diferentes actores que la integran, sino también porque su núcleo central -el conocimiento- se trabaja y transforma principalmente a través de la investigación y la enseñanza, cuya interrelación está en permanente debate [5], [6]. Como sugiere [7], la investigación enriquece el aprendizaje cuando influye en la enseñanza, lo que resalta la conexión esencial entre ambas dimensiones. De esta manera, la generación de conocimiento no debe entenderse como una actividad aislada y escindida de la enseñanza y del aprendizaje, sino como un proceso que alimenta y fortalece continuamente estos procesos.

La investigación, según [8], es un proceso sistemático y riguroso que tiene como objetivo la construcción de conocimiento sobre una determinada problemática, ésta además usa métodos y técnicas que garantizan la validez y la confiabilidad de los resultados, así como la comunicación y la difusión de los mismos. La investigación puede tener diferentes propósitos, como describir, explicar, comprender, interpretar o transformar algún aspecto de la realidad. Según [9], la actitud hacia la investigación que desarrollan los estudiantes universitarios se presenta como un indicador clave de la calidad educativa que reciben. Ello resalta la importancia de fomentar una cultura investigativa en el ámbito universitario, ya que aquellos estudiantes que se involucran en la investigación no solo adquieren competencias técnicas y metodológicas, sino que también desarrollan un pensamiento crítico, necesario para enfrentar los desafíos de su campo profesional y de la sociedad en general. Además, la inclusión de la investigación en la formación de grado no solo fortalece la calidad de la enseñanza, sino que también potencia la contribución de la universidad al desarrollo social y económico [10]. En este sentido, el enfoque investigativo no debe limitarse a los programas de posgrado, sino que debe ser parte integral del currículum desde las primeras etapas de la formación

En el caso específico de la formación docente, diversos autores han subrayado la importancia de la investigación en el proceso de formación inicial. Autores en [11] y [12] sostienen que, desde una perspectiva crítica, la investigación busca revelar las creencias, valores y supuestos que subyacen en la práctica educativa, con el fin de establecer una relación dialéctica entre la teoría y la práctica a través de la reflexión crítica. De este modo, se genera un conocimiento que surge de la praxis y está orientado a mejorarla. Para estos autores, la investigación es una herramienta de autorreflexión continua, que guía la acción educativa con el propósito de encontrar soluciones a los problemas pedagógicos en lugar de limitarse a explicarlos.

En esta misma línea, [13] menciona el rol de la investigación-acción en la transformación de la realidad social. La autora enfatiza que este proceso investigativo no debe realizarse de manera aislada, sino que está necesariamente asociada a la colaboración y el intercambio de experiencias entre educadores, fomentando una comunidad de aprendizaje en la que los docentes compartan sus hallazgos, retos y logros. De esta forma, la práctica educativa se fortalece a través del diálogo y la reflexión colectiva.

#### **3.2 Experiencias de estudiantes de computación**

Las experiencias de los alumnos del PUC en torno a la investigación en su mayoría, se relacionan con trabajos realizados en distintas cátedras. De este modo, distintos espacios curriculares se convierten en espacios privilegiados para el desarrollo de habilidades referidas a la investigación, actitudes que posibilitan la indagación, la búsqueda, la duda.

“Para Práctica 1 y 2 tuvimos que realizar una observación de la gente para tener una idea de que es una observación y luego de eso revisamos más datos. Luego también en Historia y Epistemología he hecho también algunas investigaciones, y también para materias como Ingeniería que tuvimos que buscar sobre un tema y ahí hacer un trabajo integrador. También tuvimos otra materia con investigación, donde nos dio varios buscadores, y varios formatos como las normas APA, IEEE y cómo citar” (PC001.22).

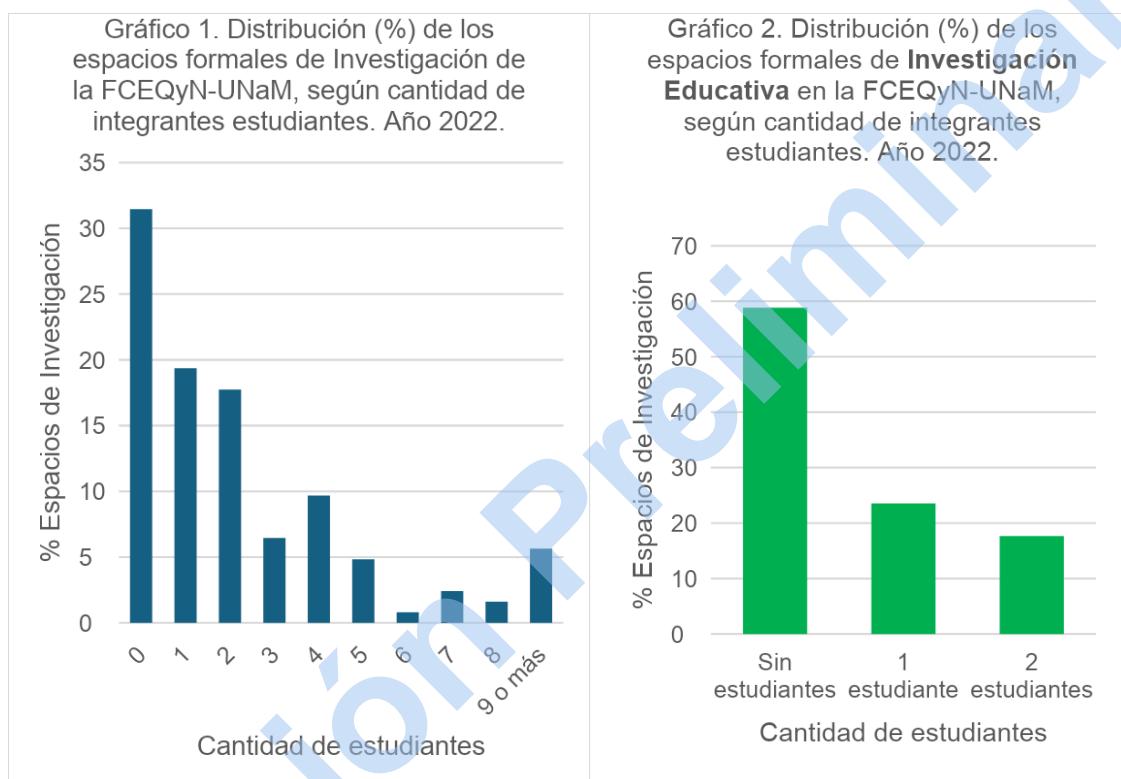
“Nosotros en problemática educativa habíamos hecho un trabajo de investigación que era problematizar algunos factores dentro del aula, sobre qué factores influyen, eso de aprendizajes... del sujeto que aprende dentro del aula, y cuáles son los factores que influyen en eso... buscar materiales bibliográficos, hablar con el docente, investigar un poco eso de cómo se da ese problema” (PC003.22)

“Realicé distintos proyectos de investigación cuando inicié la carrera. Hice en Orientación y Profesión docente mi primer proyecto de investigación y acercamiento a una escuela, donde indagamos lo que es el acompañamiento por parte de los padres en el proceso de enseñanza y aprendizaje de los chicos. Y ahí nos dimos cuenta que nosotros como investigadores teníamos algunos supuestos que después tuvimos que resignificar o significar de algún modo” (PC007.23).

Esto muestra una relación cercana con la investigación. Indagaciones que se sitúan en el centro de la práctica de la formación docente y de la futura práctica docente: rastreo bibliográfico, problematización de la realidad cotidiana, empleo de técnicas como la observación y las entrevistas, reconocimiento de buscadores académicos, formas de citación.

En este sentido, en [14] ponderan la imbricación de la investigación dentro de diferentes espacios curriculares, entendiéndola como clave “para un futuro desarrollo profesional, en una sociedad que necesita la creación y movilización del conocimiento” (p, 18). Por su parte, autores como en [15] hacen foco en la experiencia de investigar dentro del aula - como docentes o como estudiantes de formación docente-, pensándola como “un tipo de *formación para la investigación*, de mediano alcance, dirigida a “quien necesita dicha formación como apoyo para un mejor desempeño de su práctica profesional o a quien solo necesita ser un buen usuario de investigación” (p, 65).

Ahora bien, recordemos que en la universidad contamos con “espacios formales de investigación”: proyectos, trabajos o programas de investigación, acreditados mediante un proceso de evaluación por pares expertos. Para participar de ellos, los estudiantes deben estar inscritos en el SASPI, y registrados en la actividad. En un trabajo anterior [1], identificamos que, al 2022, el 31% de estos espacios dentro de la FCEQyN no incluía estudiantes. Esta ausencia dentro de numerosos equipos se profundizaba cuando analizamos aquellos espacios dedicados a la investigación educativa (un total de 17 en la unidad académica). En todos los casos referidos a investigación educativa, el máximo de estudiantes contabilizados era de dos, siendo lo más común (el 60%) que no contaran con auxiliares, tesis o becarios de grado entre sus integrantes.



Elaboración propia, fuente Secretaría de Investigación FCEQyN-UNAM 2023

Esta desconexión entre las competencias investigativas desarrolladas en el aula y la participación en proyectos formales abre interrogantes sobre la naturaleza de los apoyos y canales que facilitan (o no) la inserción de los estudiantes en los equipos de investigación. Las habilidades adquiridas en los espacios curriculares –como la capacidad para buscar y evaluar información académica, diseñar proyectos o conocer técnicas– no parecen encontrar una continuación en los espacios formales de investigación. Este desajuste puede tener diversas causas.

Por un lado, es necesario preguntarse hasta qué punto los estudiantes conocen la existencia de estos espacios formales; como así también ¿cuántos estudiantes del PUC están realmente informados o motivados para participar? Aquí entran en juego tanto la visibilidad de estos espacios como los incentivos que se ofrecen para la participación. En muchos casos, la sobrecarga académica, la falta de acompañamiento adecuado o la ausencia de recompensas claras (como créditos académicos o becas) podrían desincentivar la integración a los equipos formales. Al respecto, algunos estudiantes decían:

“No conozco a gente en investigación dentro de la facultad, pero conozco a personas que sí que realizan proyectos de investigación en la escuela, mi hermana, por ejemplo, está haciendo ahora un proyecto sobre la influencia de la que tiene la familia sobre los alumnos a la hora de desempeñarse” (PC004.23).

“Tuve una vez una profesora que se dedicaba a la investigación, en metodología de la investigación” (PC001.22).

“No conozco a nadie en investigación, más allá de los profes que supongo que hacen investigación... pero no conozco a alguna persona que se dedique a la investigación” (PC002.22).

“Considero que me incentivaron muy poco a realizar investigaciones en mi trayecto formativo. Únicamente realice investigaciones en algunas cátedras que lo exigen pero nada más, nunca tuve una motivación aparte de esa” (PC005.23).

“No conozco a nadie en investigación... por desconocer que uno como alumno puede formar parte de un trabajo de investigación, tampoco vi que haya publicaciones o algún medio difusión para ser parte de algún trabajo de investigación, quizás estaría bueno poner algo, no sé mandar un Gmail o una publicidad en un pizarrón acá del pasillo y que los chicos que están interesados y cumplan ciertos criterios puedan ser parte de algún proyecto o algún trabajo de investigación el cual quieran desarrollar a futuro” (PC004.23).

Si bien es cierto que no todos los estudiantes desean recorrer una trayectoria dedicada a la investigación, también es cierto que muchos que sí lo desean no siempre encuentran la información necesaria para incluirse en equipos. Estas experiencias de los estudiantes destacan esa realidad: el bajo conocimiento sobre los espacios formales de investigación dentro de la facultad y la limitada promoción que estos espacios reciben. La mayoría no tiene un conocimiento claro sobre los espacios formales de investigación dentro de la facultad. Las menciones de personas ajenas a la institución o de proyectos limitados a cátedras específicas refuerzan la idea de que, en muchos casos, los estudiantes del PUC no conocen con detalle las oportunidades de investigación disponibles.

Además del desconocimiento, el factor de la falta de incentivos aparece como un obstáculo significativo. La sensación general es que la investigación no se promueve activamente entre los estudiantes, salvo en los momentos donde se exige como parte de alguna materia específica. Como mencionaba uno de los entrevistados “únicamente realicé investigaciones en algunas cátedras que lo exigen, pero nada más, nunca tuve una motivación aparte de esa”. Esta afirmación sugiere que, en ausencia de un marco que fomente la curiosidad y el interés por la investigación de forma independiente, los estudiantes se limitan a cumplir con las tareas asignadas sin explorar más allá de lo estrictamente curricular.

### 3.3 Formación docente y su impacto en la educación

En estas instancias que llevamos del proyecto de investigación educativa y en nuestra formación como estudiantes universitarios, graduados y docentes, entendemos que es de gran impacto en la educación realizar investigación, por eso creemos que es significativo hablar de estos dos temas: la Formación docente y el Impacto en la educación.

La formación docente es el proceso mediante el cual los educadores adquieren los conocimientos, habilidades y actitudes necesarios para desempeñar su labor educativa de manera efectiva, realizando prácticas de enseñanza y aprendizaje que trascienden el aula, definiendo su campo laboral en circunstancias particulares. Se entiende a la “formación docente” a determinado proceso en el que se articulan prácticas de enseñanza y de aprendizaje orientadas a la configuración de sujetos docentes/enseñantes” [8, pp. 22-23]

En ese sentido se puede apreciar diferentes relaciones y trascendencias con respecto a lo que creen los estudiantes universitarios sobre la investigación en el transcurso de la formación docente.

“Yo creo que sí es necesaria la investigación en la formación docente ya que va a ayudar a mejorar las prácticas educativas, también sirve para adaptarse a las necesidades que surjan y que son cambiantes,... también a desarrollar métodos más efectivos y mantener actualizados de esta manera a los docentes con las últimas tendencias y desafíos que surjan en el campo educativo.” (PC004.23).

Analizando lo expuesto anteriormente, la formación docente permite prepararnos en el desempeño articulando prácticas de enseñanza y aprendizajes necesarias para la tarea diaria del docente. El ser docente implica una construcción a lo largo de la historia. Es crucial identificar y analizar cómo, en cada contexto y bajo ciertas ideas y prácticas dominantes, se han generado maneras específicas de relacionarse y apropiarse del conocimiento [16].

De esta manera, la formación docente proporciona a los estudiantes herramientas que permiten formarse.

“...hoy en día en formación docente hay varias herramientas de forma reflexiva, sobre todo. Te dan varias herramientas para desenvolverte en el aula y también como sujeto que enseña y sujeto que aprende,...” (PC001.22)

Entendemos que la formación del docente, tiene un gran impacto personal y social cuando se decide ser parte del sistema educativo, por tal motivo creemos que, si el docente incluye en su labor a la investigación educativa, puede aportar así nuevas estrategias, evaluar nuevas metodologías pedagógicas y ser parte de discusiones favorables para la enseñanza y su aprendizaje. Muchos autores tratan al impacto como positivo o negativo, pero debe ser visto más allá del aula y ver aún fuera de él, que esos impactos se mejoran o crecen cuando los sistemas cambian.

### **3.4 Desarrollo e Innovación profesional.**

El destino de la educación, tanto actual como futura, recae en quienes se preparan para liderarla. Ser parte de este compromiso nos posiciona en ejecutores de acciones que debemos ir incorporando. Por lo tanto, el desarrollo continuo del docente es fundamental para enfrentar su labor diaria de manera efectiva. Según lo expresado en [17], el proceso de investigación educativa tiene como objetivo el perfeccionamiento de los que participan en dichas situaciones, transformando sus conocimientos, actitudes y comportamientos. Esto se debe a que investigar representa, al principio, una experiencia desafiante, pues implica reflexionar y cuestionar las propias percepciones construidas a lo largo de la historia, especialmente en un entorno escolar cargado de normativas, valores y categorías oficiales que definen el posicionamiento teórico, lo que obliga al investigador a ver una realidad más compleja.

Mientras que la innovación depende en gran medida de los docentes, quienes deben sentir que los cambios propuestos tienen relevancia y se alinean con sus inquietudes y necesidades. En otras palabras, la participación activa y el compromiso del profesorado son esenciales para lograr transformaciones duraderas en la educación.

Creemos que la innovación en la educación nace a partir de grandes investigaciones y es vista, además como un elemento importante que trae numerosos beneficios para el ámbito educativo. En [18], algunos de estos beneficios son: mejora de la calidad de la enseñanza, proporcionar al docente fundamentos para la toma de decisiones, enriquecer y modernizar a la educación y el desarrollo docente. Sin embargo, es importante destacar que el profesorado debe estar dispuesto a realizar cambios de manera creativa, con aceptación y comunicación de todos los involucrados en el ámbito educativo. El docente necesita estar capacitado por su formación y además, se requiere de una planificación y diseño previos que permitan llevar a cabo la innovación, para evitar una desconexión entre lo esperado y lo real.

## **4. Conclusiones**

Llevar a cabo este proyecto de investigación nos permitió identificar inquietudes sobre la investigación educativa que no habíamos considerado en los primeros años del profesorado universitario en computación. El participar activamente como estudiantes avanzados y graduados, junto a docentes de la cátedra de Metodología de la Investigación Educativa, nos brindó la oportunidad de reflexionar profundamente sobre nuestra futura labor como docentes. En este proceso, comprendimos que la investigación no solo es un componente esencial del crecimiento profesional, sino también un motor para la innovación en el ámbito educativo.

La investigación educativa se presenta como una vía para promover una mejora continua en la enseñanza, impulsando transformaciones que resultan de la reflexión crítica de los docentes sobre sus propias prácticas. Este enfoque permite descubrir nuevas metodologías que optimizan la enseñanza, beneficiando tanto a los educadores como a los estudiantes. En este sentido, los resultados de nuestro trabajo revelan una variedad de perspectivas en torno a la producción de conocimiento científico en la universidad, y evidencian que los estudiantes del profesorado a menudo perciben su rol en este proceso de manera limitada.

A partir de estos hallazgos, consideramos fundamental promover una mayor conciencia y participación estudiantil en los proyectos de investigación. Es necesario que la universidad genere oportunidades y recursos que permitan a los estudiantes involucrarse activamente en estos espacios desde las primeras etapas de su formación. La creación de una cultura de investigación sólida, que reconozca y valore las contribuciones estudiantiles, es clave para incentivar vocaciones científicas y para consolidar la idea de que la investigación es parte integral del desarrollo profesional de los futuros docentes.

Como acciones concretas, destacamos la importancia de difundir y aprovechar programas existentes como las Becas de Estímulo a las Vocaciones Científicas (CIN), así como las becas ofrecidas por la Secretaría General de Ciencia y Técnica de la UNaM. Además, iniciativas como las “Cátedras en Acción 2024”, que vinculan docencia, extensión e investigación, junto con las complementaciones de títulos, ofrecen formas de formalizar y visibilizar la participación de los estudiantes en actividades investigativas. Estas acciones deben ser reconocidas como parte central del currículum universitario, contribuyendo a la formación de docentes-investigadores.

Finalmente, como equipo de investigación, consideramos crucial seguir indagando en estos temas para profundizar en la relación entre investigación, formación docente y mejora educativa. Continuar investigando las experiencias de los docentes-investigadores y el impacto de la investigación educativa en la FCEQyN será un paso importante hacia el fortalecimiento de la cultura científica en nuestra carrera y en la universidad en su conjunto.

## Referencias

- [1] Avalos, M. Radovic, A. Zacharski, J. y Díaz Gómez, A. (2023). "Investigación y formación docente en la FCEQyN-UNaM. Aproximaciones al campo", ponencia presentada en las II Jornadas de Estudios Socio-Antropológicos Leopoldo José Bartolomé. Universidad Nacional de Misiones.
- [2] Avalos, M., Werbes, M. y Radovic, A. (2024). "Investigación en la Universidad Nacional de Misiones. Perspectivas desde un profesorado de la FCEQyN". Ponencia aceptada para ser presentada en el IV Simposio de Investigación, Extensión y Desarrollo Local de Posadas.
- [3] McMillan, J., y Schumacher, S. (2005). Investigación educativa. Pearson Educación, Madrid.
- [4] Vasilachis, I. (2019). Estrategias de investigación cualitativa. Volumen II. Gedisa Editorial.
- [5] Barnett, R. (2008). Para una transformación de la universidad. Nuevas relaciones entre investigación, saber y docencia. Barcelona, España. Octaedro.
- [6] Calvo, G. R. (2021). La formación en investigación en la universidad: el caso de las carreras humanas y sociales: (1 ed.). Buenos Aires, Argentina, Miño y Dávila.
- [7] Adúriz-Bravo, A. (2007). A proposal to teach the nature of science (NOS) to science teachers: The 'structuring theoretical fields' of NOS. *Review of Science, Mathematics and ICT Education*, 1(2), 41-56.
- [8] Achilli, E. (2000). El sentido de la investigación en la Formación Docente. *Investigación y Formación Docente* (pp. 17 – 44). Laborde Editor.
- [9] Papanastasiou, E. (2005). Factor structure of the "Attitudes toward Research" Scale. *Statistics Education Research Journal*, 4, (1), 16-26.
- [10] Elsegood, L., Ávila Huidobro, R., Garaño, I., y Harguinteguy, F. (2014). Universidad, territorio y transformación social: reflexiones en torno a procesos de aprendizaje en movimiento. Avellaneda, Undav Ediciones.
- [11] J. Arnal, D. del Rincón y A. Latorre, "Naturaleza de la investigación educativa," en *Investigación Educativa: Fundamentos y Metodología*. Edición ilustrada. España: Labor, 1992, pp. 31 - 57.
- [12] Carr, W. (1996). *Una teoría para la educación: hacia una investigación educativa crítica*. Ediciones Morata.
- [13] Zoppi, A. M. (2012). Lo metodológico en la investigación educativa. En La investigación acción en la Autoformación Permanente de Profesores (pp. 153-163). Universidad Nacional de Jujuy. <https://anamariazoppi.com.ar/>
- [14] Colás Bravo, Pilar, & Hernández de la Rosa, Miguel Ángel. (2021). Research competences in university training. *Revista Universidad y Sociedad*, 13(1), 17-25.
- [15] Cervantes Holguín, E.(2019). Un acercamiento a la formación de docentes como investigadores educativos en México. *REICE: Revista Iberoamericana sobre Calidad, Eficacia y Cambio en Educación*, 17(4), 59-74.
- [16] Zoppi, A. (2009). Los docentes: Los docentes y el conocimiento. *Revista de Pedagogía Crítica*, Año 8, N° 7.
- [17] Pérez Gómez, A. (1993) Comprender la enseñanza en la escuela. Modelos metodológicos de investigación educativa. En: Gimeno Sacristán, J. y Pérez Gómez, A. Comprender y transformar la enseñanza. Ed. Morata.
- [18] Martínez, O. (2024). Educational Research: A Beacon that Illuminates the Path Towards Transformation. *Revista Scientific*, 9(Ed. Esp.), 10-18, e-ISSN: 2542-2987. <https://doi.org/10.29394/Scientific.issn.2542-2987.2024.9.E.0.10-18>

# ¿Es posible enseñar los fundamentos de la programación sólo con Arduino?: Análisis de un curso introductorio para estudiantes de secundaria usando bloques

Gonzalo Pablo Fernández<sup>1</sup> y Christian Cossio-Mercado<sup>2</sup>

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

Departamento de Ciencia y Tecnología. Buenos Aires, Argentina.

<sup>2</sup> Universidad de Buenos Aires. Facultad de Ciencias Exactas y Naturales.

Departamento de Computación. Buenos Aires, Argentina.

gpfernandez@dc.uba.ar, ccossio@dc.uba.ar

## Resumen

No es común que se utilicen placas Arduino como primera herramienta para enseñar a programar y, menos aún, que se las utilice para enseñar los fundamentos de la disciplina, sino que, en general, aparecen luego de haber aprendido a programar con otras herramientas, o se las usa para enseñar nociones de robótica y electrónica. En este trabajo se presentan los resultados de un curso introductorio de programación para estudiantes de secundaria, enfocado en los fundamentos y no sólo en escribir código, utilizando únicamente programación por bloques para Arduino. Así, cada concepto de programación se introdujo a través de actividades basadas en Arduino, sin complementar con otras herramientas, y como cierre del curso se realizó un proyecto grupal a elección. Se aprecia un progreso en la autonomía de los participantes, no sólo por la realización exitosa de las tareas propuestas, sino también por el desempeño alcanzado al concretar el proyecto final. A pesar de los resultados positivos, creemos que el usar sólo un entorno deriva en que los estudiantes crean que sólo aprendieron una tecnología concreta, en lugar de habilidades generales de programación. Así, proponemos que se utilicen varias herramientas para que los estudiantes puedan apreciar que están incorporando conceptos y habilidades generales, y no de un lenguaje de programación en particular.

**Palabras clave:** Enseñanza de programación, Arduino, Programación por bloques, Escuela Secundaria

## 1. Introducción

Al momento de diseñar un curso de programación es habitual que el equipo a cargo se plantee varias preguntas, teniendo en cuenta a sus estudiantes: ¿Qué necesitan los estudiantes para aprender a programar? ¿Cómo podemos asegurar que una persona sabe programar? ¿Cómo puede saber un estudiante que sabe programar? ¿Qué enseñar y cómo? ¿Por qué tendría sentido enseñar los conceptos fundamentales de la programación y no sólo a escribir código? Estas son algunas de las preguntas que motivan este trabajo, pero también la labor docente al momento de diseñar un nuevo taller, curso o materia con el objetivo de enseñar a programar a gente sin experiencia. En particular, en este trabajo haremos foco en la última pregunta.

### 1.1. ¿Qué se debe enseñar?

Entre las muchas cuestiones que se deben tener en cuenta al momento de planificar un curso introductorio de programación está cuáles deberían ser los contenidos esenciales a enseñar y en qué orden deberían presentarse. La mayoría de los enfoques clásicos consideran indispensable comenzar con variables y limitarse al paradigma imperativo [1], postergando el trabajo de crear subtareas por medio de procedimientos y funciones o, directamente, omitiéndolo. También es algo común hacer uso del sistema de entrada-salida a través del famoso “Hola mundo!”, como es habitual en la bibliografía. Los enfoques más recientes invierten este esquema, dándole un peso importante a la creación de subtareas, delegando las variables al final y dejando la comunicación con el sistema de entrada-salida por fuera de los contenidos esenciales [2].

### 1.2. ¿Cómo se debe enseñar?

Otro aspecto importante sobre el que también hay poco acuerdo es respecto a qué herramientas utilizar, tanto en lo respectivo a lenguajes de programación como a entornos de desarrollo. A fines de la década del 60, Feurzeig, Seymour y Solomon desarrollaron Logo [3], el primer lenguaje diseñado específicamente para enseñar a programar, aunque pasaría más tiempo hasta que la enseñanza de la programación se convirtiera en un tema de interés general y, como consecuencia, se desarrollara la investigación sobre cómo enseñar mejor a programar, en particular a niños y jóvenes.

Luego de Logo empezaron a surgir nuevos lenguajes de programación orientados a la enseñanza. En muchos estudios se destacó el problema de la sintaxis de los lenguajes formales como el principal obstáculo para aprender a programar [4, 5]. Posteriormente, Python empieza a popularizarse en el ámbito educativo, gracias a su sintaxis simplificada [6], además

de que es uno de los lenguajes más usados<sup>1</sup>, tanto en ámbitos educativos como profesionales. A su vez, aparecen diferentes entornos de programación visual, entre los que se destacan los entornos de programación por bloques como Scratch [7] y Alice [8]. Entre otras ventajas, estos entornos, diseñados específicamente para aprender a programar, simplifican casi totalmente el problema de generar programas con cierta sintaxis [7], lo que impulsó rápidamente su adopción en múltiples contextos educativos, sobre todo escolares [4].

En paralelo al de la sintaxis, surge otro problema con respecto a la enseñanza de la programación, relacionada a la dificultad de la abstracción [9]. Para ello, entre otros enfoques, se utilizan robots y otros dispositivos electrónicos para enseñar a programar [10, 11]. Estos permiten llevar la programación a un contexto concreto y físico, lo que es especialmente útil como motivación para trabajar con entidades abstractas como las listas y los números, en particular para estudiantes más pequeños, que se estima que aún no han desarrollado totalmente su capacidad de abstracción [12, 13, 14]. El trabajo con programación con efectos en el mundo físico aumenta la motivación y fomenta el pensamiento creativo, además de permitir resolver problemas cercanos a la vida cotidiana [15]. Adicionalmente, tanto el trabajo con programación con salida en el mundo físico como sólo en pantalla aportan al desarrollo de las habilidades asociadas al pensamiento computacional [16].

## 2. Estado del arte

Respecto a cómo se enseña actualmente programación, una de las estrategias predominantes es el uso de entornos de programación por bloques diseñados para controlar robots o dispositivos con salidas en el mundo físico [15], lo que resuelve tanto el problema de la sintaxis como el de la abstracción, además de trabajar con una salida tangible. Sin embargo, hay fuertes críticas tanto al uso de bloques como al uso de robots en la formación inicial de la programación, en tanto se obtienen mejores resultados en el desarrollo de pensamiento computacional usando sólo una salida en pantalla [17]. Estas críticas se basan en que como parte de la enseñanza no debería obviarse el enfrentarse con las dificultades de aprender sintaxis, que queda simplificado notablemente con el trabajo con bloques [18, 19], ni tampoco el esfuerzo requerido para trabajar con elementos abstractos ya que ambos son fundamentales para formarse en programación. A esto se suma el costo de compra y mantenimiento de los materiales, y el armado de una comunidad de soporte a los equipos docentes, que no debe despreciarse [20].

Los entornos de programación por bloques a veces son vistos como demasiado “infantiles”, por lo que se los suele evitar cuando se trabaja con adolescentes mayores o adultos. En estos casos el lenguaje predominante suele ser Python, pese que a que hay casos exitosos del uso de bloques en la educación superior [21, 22]. Esto último es importante, en tanto las personas adultas pueden tener una mayor formación y habilidades generales que los niños, pero también tienen dificultades al enfrentarse a la sintaxis de un lenguaje de texto o para resolver problemas que requieren trabajo mayormente abstracto.

En Argentina, en los cursos iniciales de programación del nivel superior se utilizan una gran variedad de lenguajes y entornos [23]. Mientras tanto, en educación media aparecen cada vez con más frecuencia planificaciones basadas en entornos de programación por bloques como Scratch o PilasBloques [24] (e.g., de acuerdo a las pautas en los manuales Program.AR [25, 26]). En cuanto a la Robótica, si bien hay una gran cantidad de escuelas enseñando con kits propietarios, como los de LEGO<sup>2</sup> y micro:bit<sup>3</sup>, también hay muchas herramientas para el trabajo con placas Arduino [27, 28]. Estas últimas tienen la ventaja de que permiten trabajar en dominios mucho más variados y no se limitan únicamente a trabajar con un modelo de robot particular, aunque traen consigo dificultades adicionales relacionadas al manejo de conceptos de electrónica básica, como la conexión de cables, y en lo respectivo a la programación, ya que el entorno de programación de Arduino (i.e., el Arduino IDE) es en texto y no es tan fácil de usar como los entornos por bloques ni, inclusive como otros lenguajes de texto como Python. Muchos de los entornos de programación por bloques para Arduino ocultan, en mayor o menor medida, la mayoría de estos aspectos, además de que los kits de robótica suelen venir con conectores prearmados, de tipo clip, que evitan tener que conectar cables individualmente.

Respecto a qué enseñar, la gran mayoría de las propuestas se pueden separar entre aquellas que enseñan programación con foco en lo técnico, apuntando a formar profesionales, y aquellas que enseñan con base en lo conceptual, poniendo el foco en los fundamentos de la programación. Los cursos y talleres del primer grupo suelen usar lenguajes como Python, en tanto su objetivo es que los estudiantes puedan empezar a escribir código del estilo que se haría en la industria. Los del segundo grupo en cambio se sostienen en las herramientas educativas que permiten estudiar a fondo los conceptos antes de empezar a estudiar los detalles de un lenguaje de uso profesional.

Las propuestas basadas en robótica educativa pueden encontrarse en el medio ya que buscan poner el foco en la enseñanza de conceptos de programación, pero muchas veces terminan enseñando conceptos más relacionados a la robótica o la mecánica que a la programación. Suele suceder algo similar con las propuestas que usan Arduino como primer acercamiento a la programación. Es habitual que estas herramientas sean utilizadas no como primer acercamiento a la programación, sino en una etapa posterior, a modo de transferencia de aprendizaje, durante la cual no se enseñan nuevos conceptos sino que se aplican los ya estudiados previamente en otros contextos.

En este trabajo presentamos el diseño y resultados de la realización de un curso breve para la introducción de la programación para estudiantes del nivel secundario, utilizando programación por bloques, donde se trabaja con placas

<sup>1</sup>The Top Programming Languages 2024 - IEEE Computer Society: <https://spectrum.ieee.org/top-programming-languages-2024>

<sup>2</sup>LEGO Education website: <https://education.lego.com>

<sup>3</sup>micro:bit website: <http://www.microbit.org>

Arduino, sensores y actuadores, donde el foco está en los fundamentos y no sólo en la habilidad de escribir código. Se muestra que el enfoque utilizado funciona efectivamente para introducir los conceptos básicos de programación, además de que se resumen algunas ideas principales de ventajas y desventajas del uso de este tipo de herramientas y algunas conclusiones sobre cómo estructurar mejor un curso de este tipo.

### 3. Metodología

#### 3.1. El curso

El curso consistió de cinco encuentros de tres horas cada uno, y se realizó por completo en español. Los primeros cuatro encuentros se enfocaron en enseñar los fundamentos de la programación (elementos del lenguaje, nociones de estrategia y división en subtareas, etc.), y el último fue dedicado exclusivamente al trabajo grupal con proyectos. Durante los primeros cuatro encuentros predominó el trabajo de tipo taller, pero intercalando espacios de discusión para reconstruir los conceptos y problematizar adecuadamente lo que se está haciendo. En promedio, cada encuentro consistió en dos horas de práctica, durante las cuales los alumnos trabajaron en grupos de dos o tres personas, 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. Dentro de la práctica se incluyó también en cada encuentro un espacio de exploración libre con algún módulo particular, pensado para fomentar el trabajo autónomo y motivar a los estudiantes.

Para el trabajo de taller se utilizó una herramienta de programación por bloques para Arduino llamada Arduino en la Escuela<sup>4</sup>[29]. 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. El último encuentro se destinó por entero al trabajo en grupo con proyectos, los cuales se podían elegir de una lista de propuestas, además de que cada equipo podía optar por trabajar con un proyecto propio, elegido con ayuda del equipo docente.

Para el diseño del curso se siguió un marco teórico-conceptual propio que se utiliza desde experiencias anteriores, definido bajo el nombre PRENDER [30], el cual integra en una sola propuesta a la Enseñanza para la Comprensión [31], la Evaluación para el Aprendizaje [32], Aprendizaje por Indagación [33, 34], el trabajo con problemas auténticos [35], entre otros marcos conceptuales. Además, seguimos algunos lineamientos conceptuales para la enseñanza de la programación [36], entre los que destacamos el de promover una *visión denotacional* de la programación, que pone el foco en el significado de los conceptos y en aquello que representan, en contraposición a la clásica *visión operacional*, que pone el foco en la forma en que estos conceptos se implementan en la práctica. Creemos que es indispensable motivar a que los estudiantes encaren los distintos aspectos de la programación desde este punto de vista para que adquieran una visión global y de alto nivel sobre la programación.

#### 3.2. Detalle del curso

A continuación se detallan los objetivos, contenidos y herramientas usadas en los primeros cuatro encuentros del curso.

##### Primer encuentro: Introducción a la Programación y Arduino

Este encuentro tiene dos objetivos principales. El primero es familiarizarse con los conceptos elementales relacionados a la electrónica, sin profundizar demasiado en los detalles técnicos. El foco del curso no es enseñar electrónica sino usarla como medio para enseñar las bases conceptuales de la programación. El segundo es empezar a trabajar con programación pero no usando el enfoque clásico de interpretar un programa como una mera secuencia de instrucciones sino como la descripción de un comportamiento con un propósito específico.

Además de introducir los conceptos de *sensor*, *actuador* y *programa* se presentó el elemento del lenguaje *comando*. La mayor parte de la práctica se dedicó a describir distintos comportamientos usando únicamente una luz LED. Para la parte exploratoria se utilizó la matriz de LEDs. La parte de discusión incluyó debates sobre qué es programar y sobre cómo explicar el comportamiento de distintos dispositivos presentes en la vida cotidiana (como las puertas del shopping o una pava eléctrica) haciendo énfasis en los sensores y actuadores involucrados.

##### Segundo encuentro: Comandos y Actuadores

El objetivo de este encuentro fue seguir explorando el uso de comandos para crear comportamientos más complejos, sin perder de vista que los programas deben poder comunicar la solución a un problema de manera clara y legible. Para ello se presentaron las nociones de *estrategia de solución* y *división en subtareas* además de introducir el elemento del lenguaje *procedimiento*. Para la parte práctica se siguió trabajando con el LED y luego se reemplazó por el zumbador (buzzer). La parte exploratoria consistió en utilizar el zumbador para generar melodías. Esta actividad también permitió revelar la necesidad de parámetros, que se mencionaron aunque su formalización se vería recién en el tercer encuentro. La parte de discusión se focalizó en el concepto de “computadora” como “entidad que realiza cómputos” en relación a las definiciones de “programación” y “programa” trabajadas en el encuentro anterior.

<sup>4</sup>Arduino en la Escuela - Demo: aeje.dc.uba.ar.

Cuadro 1: Encuesta inicial: Saberes previos de programación.

Pregunta	Tipo de respuesta	Código
¿Sabés programar? ¿En qué lenguaje(s)? Enumeralos todos.	Libre	PRE-PROG
¿Alguna vez usaste algún entorno de programación por bloques? ¿Cuál(es)? Marcá todos los que hayas usado.	Casillas de verificación de selección múltiple (Scratch, Pilas-Bloques, Alice, Lightbot, Code.org, AppInventor, Gobstones Jr., No usé ninguno de los anteriores, Otro)	PRE-BLOCK

### Tercer encuentro: Expresiones y Sensores

El objetivo de este encuentro fue adquirir una visión conceptual de alto nivel sobre la programación, cómo los distintos elementos del lenguaje (comandos y expresiones) permiten crear programas vistos como descripciones de comportamientos a partir de las interacciones con sus contrapartes en el mundo físico (actuadores y sensores, respectivamente) y cómo dividir en subtareas un problema complejo antes de ponerse a pensar qué comandos y expresiones primitivas son necesarias para escribir el código.

El encuentro comenzó retomando la noción del concepto de *parámetro* que, luego de ser formalizado, sirvió para definir, también, el concepto de *expresión*. La parte práctica consistió en resolver un único problema, aunque más elaborado que las actividades de los encuentros anteriores (i.e., la luz que se enciende cuando está oscuro), el cual motivó la necesidad de la *alternativa condicional*. Distintas variantes del problema permitieron presentar los conceptos de *tipo*, haciendo uso del valor numérico de intensidad lumínica en lugar del binario que indica la ausencia de luz, y de *operador*, comparando valores y componiendo condiciones booleanas. Los nuevos sensores utilizados fueron el sensor de luminosidad LDR y el sensor de movimiento PIR.

### Cuarto encuentro: Datos y comunicaciones

Para este encuentro el objetivo fue completar el inventario de elementos del lenguaje disponibles, introduciendo el concepto de *función* y trabajar con proyectos ligeramente más abiertos que los anteriores. La discusión permitió construir un mapa conceptual con todos los elementos del lenguaje, al categorizarlos según si son comandos o expresiones y según si son primitivos, compuestos o definidos por el usuario. La parte práctica comenzó con el desarrollo de una estación de medición y registro de datos. El sensor usado para el proyecto fue el micrófono, pero se desarrolló de forma que la variable medida fuera fácilmente reemplazable. Para el registro de los datos se utilizó el lector de tarjetas SD y reloj. Luego se implementó un asistente de estacionamiento usando el sensor de distancia (sonar) y el zumbador. En este segundo proyecto el foco estuvo en determinar cuál debería ser el comportamiento, ya que no era trivial. El cierre de la actividad sirvió para resumir en qué consiste la tarea de la programación, donde escribir el código es sólo el último paso: primero se debe entender el problema, dividirlo en problemas más pequeños que se puedan resolver por separado, proponer soluciones a estos problemas que tras probarlas pueden fallar, por lo que hay que volver a pensar la estrategia e iterar hasta conseguir una solución satisfactoria.

### Quinto encuentro: Trabajo con proyectos

A diferencia de los cuatro encuentros anteriores, este encuentro se dedicó completamente al desarrollo de un proyecto grupal a elección. Se presentó una guía para seleccionar un proyecto que debían diseñar y para el cual no se esperaba que implementen del todo, pero sí que construyeran un prototipo funcional de una parte del mismo. La guía permitía elegir un proyecto ya definido, armar uno nuevo a partir de combinar varios o, incluso, tomar uno incompleto y que el grupo decidiera cómo completarlo. Durante un tiempo los grupos trabajaron en un documento en el que describieron detalladamente el proyecto, con el nombre, objetivos, materiales necesarios, entre otras definiciones. Luego se les pidió que armen un prototipo con los sensores y actuadores disponibles y que programen, al menos, la principal funcionalidad del sistema. Para finalizar se les pidió un boceto para un poster de divulgación que presentarían a fin de año en un congreso de cierre de todos los talleres de ciencias de la facultad. Algunos de los proyectos desarrollados fueron un reproductor de música para karaoke, un juego de snake, una caja de seguridad y un rover.

### 3.3. Población

Las personas que asistieron al taller fueron estudiantes de entre 16 y 18 años de escuelas medias del área metropolitana de Buenos Aires (AMBA). Aunque sólo 5 (11 %) personas cursan sus estudios en instituciones con orientación en informática, el 88 % aseguró haber tenido formación en contenidos relacionados con programación (e.g., computación, informática y TICs). En particular, el 30 % del total declaró haber usado Scratch en la escuela, mientras que un 5 % dijo haber utilizado PilasBloques como parte de sus clases, y el 13 % había usado placas Arduino con anterioridad.

Cuadro 2: Encuesta final: Saberes previos de programación.

Pregunta	Tipo de respuesta	Código
¿Considerás que sabías programar antes de empezar el taller? ¿Por qué?	Libre	POST-PROG-START
Y ahora que terminaste el taller, ¿Considerás que sabés programar? ¿Por qué?	Libre	POST-PROG-END

Cuadro 3: Encuesta final: Evaluación del curso.

Pregunta	Tipo de respuesta	Código
¿Qué te pareció el taller en general? ¿Creés que te aportó algo haberlo realizado?	Libre	POST-GAIN
¿Qué te gustaría haber aprendido en el taller que no aprendiste? ¿Por qué te interesa?	Libre	POST-LACK

### 3.4. Encuestas

Como parte del curso se realizaron dos encuestas; una al comenzar el primer encuentro (encuesta inicial) y otra al finalizar el último encuentro (encuesta final). La motivación principal de estas encuestas fue medir, de forma general, el cambio en las concepciones sobre programación de los estudiantes, comparando los conocimientos al inicio y al final del curso. Además, se quería identificar el grado de conocimiento de los estudiantes acerca de conceptos básicos de programación, algunos de los cuales se abordarían en el curso, así como relevar la autopercepción de los estudiantes sobre sus conocimientos en programación después del curso, además de en perspectiva con respecto al comienzo del mismo.

En el Cuadro 1 se incluyen las preguntas de la encuesta inicial relacionados a los saberes previos sobre programación. Por otro lado, en la encuesta final se volvió a preguntar acerca de los saberes previos al comienzo del curso, como se muestra en el Cuadro 2. Respecto a los conocimientos sobre conceptos relacionados a la programación, en ambas encuestas se pidieron definiciones para cada uno de los siguientes conceptos: “programar”, “algoritmo”, “computadora”, “lenguaje de programación” y “programa”. En la encuesta final también se agregaron preguntas de evaluación del curso, como se resumen en el Cuadro 3.

### 3.5. Procesamiento preliminar de las respuestas

Como parte del diseño de las encuestas se optó por usar preguntas de respuesta abierta, para que sea más fácil y más corto de responder, además de para explorar mejor lo que los estudiantes tenían para decir, por lo que fue necesario realizar una tabulación de las respuestas para poder analizar los resultados y poder hacer comparaciones entre la encuesta inicial y la final.

Las respuestas a la pregunta “¿Sabés programar? ¿En qué lenguaje(s)? Enumeralos todos” de la encuesta inicial (PRE-PROG) se tabularon según las categorías del Cuadro 4. En el Cuadro 5 se puede ver los criterios para el tabulado de las respuestas a la pregunta “¿Alguna vez usaste algún entorno de programación por bloques? ¿Cuál(es)? Marcá todos los que hayas usado” de la encuesta inicial (PRE-BLOCK). Las respuestas a la pregunta “¿Considerás que sabías programar antes de empezar el taller? ¿Por qué?” de la encuesta final (POST-PROG-START) se tabularon según las categorías el Cuadro 6. En el Cuadro 7 se puede ver el tabulado para las respuestas a la pregunta “Y ahora que terminaste el taller, ¿Considerás que sabés programar? ¿Por qué?” de la encuesta final (POST-PROG-END).

De las respuestas a las preguntas “Y ahora que terminaste el taller, ¿Considerás que sabés programar? ¿Por qué?” (POST-PROG-END) y “¿Qué te pareció el taller en general? ¿Creés que te aportó algo haberlo realizado?” (POST-GAIN) de la encuesta final se identificaron aquellas que hacían referencia a haber aprendido a programar en general y aquellas que hacían referencia a haber aprendido sólo un lenguaje.

De las respuestas a “¿Qué te gustaría haber aprendido en el taller que no aprendiste? ¿Por qué te interesa?” de la encuesta final (POST-LACK) se identificaron aquellas que pedían otros lenguajes, aquellas que pedían programar en texto o en código (si bien no es lo mismo, agrupamos estas respuestas porque asumimos que quienes pidieron trabajar con código creen que el código tiene que ser en texto) y aquellas que pidieron estudiar algoritmos.

Las respuestas a las definiciones se tabularon en diferentes categorías para cada uno de los cinco conceptos (“programar”, “algoritmo”, “computadora”, “lenguaje de programación” y “programa”). Además, a cada una de esas categorías se le asignó una calificación categórica entre Bien (B), Regular (R), Mal (M) o No sabe / no contesta (N). En los cuadros 8 a 12 se muestran, para cada concepto las distintas categorías de tabulación para las respuestas, junto a su calificación, su descripción y algunos ejemplos.

Cuadro 4: ¿Sabés programar? (encuesta inicial): Tabulado de respuestas.

Categoría	Explicación
Sí	Dice que sí y/o incluye algún lenguaje de programación
No	Dice que no
Poco	Respuestas como “un poco” o “más o menos”
Web	Dice que sí pero al listar los lenguajes que conoce sólo menciona HTML y/o Css

Cuadro 5: ¿Usaste programación por bloques?: Tabulado de respuestas.

Categoría	Explicación
LightBot	Sólo marcó LightBot
PilasBloques	Marcó PilasBloques pero no marcó Scratch
Scratch	Marcó Scratch
Ninguno	Marcó la opción “No usé ninguno de los anteriores”

Cuadro 6: ¿Sabías programar antes del curso? (encuesta final): Tabulado de respuestas.

Categoría	Explicación
Sí	Asegura que sabía programar
No	Declara que no sabía programar
Poco	Respuestas como “un poco” o “más o menos”
Base	Respuestas como “sólo básico” o “sabía las bases”

Cuadro 7: ¿Ahora sabés programar? (encuesta final): Tabulado de respuestas.

Categoría	Explicación
Sí	Asegura que sabe programar
No	Declara que no sabe programar
Poco	Respuestas como “un poco” o “más o menos”
Base	Respuestas como “sólo básico” o “sé las bases”
Sí +	Dice que sabe y además, que sabe más de lo que sabía al principio
No +	Dice que no sabe pero que sabe más de lo que sabía al principio

Cuadro 8: Categorías tabuladas a las definiciones del concepto “programar”

Código	Calificación	Descripción	Ejemplos
DECIR	B	Foco en la comunicación	Decirle a la computadora lo que tiene que hacer ; Comunicar
CODE	M	Foco en escribir código como tarea principal	Escribir código ; Usar códigos para crear ...
INST	R	Foco en instrucciones como elementos esenciales que componen un programa	Dar órdenes ; Dar instrucciones
ALGO	R	Foco en los algoritmos como componentes	Proceso de ordenar algoritmos
PROBLEM	B	Foco en resolver un problema	Crear algo para resolver un problema ; Solucionar un problema mediante ...
CREA	B	Foco en desarrollar algo (no necesariamente un programa)	Crear un proyecto ; Desarrollar algo
SOFT	R	Foco en el software como producto final	Crear un software o una aplicación ; Diseñar una página web
CODE+INST	M	Combinación de CODE y INST	Utilizar códigos para dar instrucciones
CODE+SOFT	M	Combinación de CODE y SOFT	Crear una página web mediante un código
O	M	Otras respuestas difíciles de clasificar	La acción de hacer que un programa cumpla su función
NO SE	N	Prefiere no arriesgar	No sé
X	-	Respuesta inválida	Sí

Cuadro 9: Categorías tabuladas a las definiciones del concepto “computadora”

Código	Calificación	Descripción	Ejemplos
HARD	M	Foco en el Hardware, lo electrónico y/o lo tecnológico	Máquina electrónica ; Dispositivo tecnológico
CEREBRO	R	Foco en la analogía de la computadora como el cerebro de un sistema	El cerebro del sistema
PROG	B	Foco en la característica de ser programable	Máquina programable ; Autómata de propósito general ; ... con comportamiento variable
COMPU	B	Foco en su función de computar	Cualquier cosa que computa / que procesa información
AUTO	R	Foco en su comportamiento de ejecutar instrucciones automáticamente	Máquina que procesa y ejecuta órdenes
INTEL	M	Foco en la inteligencia	Máquina inteligente
PROG2	R	Más “sirve para programar” (como un editor de texto) que “es programable”	Máquina que sirve para programar ; Donde corremos los programas para ver si funcionan
HARD+PROG	R	Combinación de HARD y PROG	Máquina digital programable
HARD+SOFT	M	Combinación de HARD y SOFT	Compuesta por Hardware y Software
HARD+AUTO	R	Combinación de HARD y AUTO	Aparato electrónico para automatizar procesos
O	M	Otras respuestas difíciles de clasificar	Herramienta de trabajo ; Conjunto de componentes
NO SE	N	Prefiere no arriesgar	No sé

Cuadro 10: Categorías tabuladas a las definiciones del concepto “lenguaje de programación”

Código	Calif.	Descripción	Ejemplos
DATA	M	Foco en la forma de expresarlo con símbolos	Números y letras
LANG	R	Foco en su característica de ser un lenguaje (pero no aclarar nada más)	Un lenguaje para dar órdenes
PC	R	Foco en que puede ser interpretado (únicamente) por computadoras (quizás pensando en el binario)	El idioma que hablan las computadoras
IDE	M	Foco en el entorno de desarrollo asociado	Un entorno para programar
COMPILE	M	Parece estar pensando más en un compilador	Herramienta para traducir ...
CODE	M	Foco en el código o su generación	Líneas de código ; Para crear códigos
PROG	B	Foco en su objetivo de programar / crear un programa	Para programar
SINTAX	R	Foco en las reglas sintácticas y en ser la composición de un programa	Cómo se encuentra codificado un programa ; Reglas para escribir un programa
COMM	B	Foco en la comunicación entre personas y computadoras	Una forma de comunicarnos con la computadora
WEB	M	Sólo sirve para hacer páginas web	Para prograr una página web
INST	M	Foco en las instrucciones	Un set de instrucciones
ALGO	R	Foco en su objetivo de crear algoritmos	Para crear algoritmos
LANG+PROG	B	Combinación de LANG y PROG	El lenguaje en el que se programa
LANG+CODE	R	Combinación de LANG y CODE	El lenguaje en el que están escritos los códigos
O	M	Otras respuestas difíciles de clasificar	Forma en la que se estudia la programacion
NO SE	N	Prefiere no arriesgar	No sé

Cuadro 11: Categorías tabuladas a las definiciones del concepto “programa”

Código	Calif.	Descripción	Ejemplos
CODE	M	Foco en el código	Una composición de códigos
EXE	R	Foco en ser ejecutable	Algo ejecutable
PROC	M	Definición de proceso	Procesos corriendo en la computadora
ALGO	R	Foco en estar “compuesto” de algoritmos	Conjunto de algoritmos
INST	R	Foco en las instrucciones que lo componen	Instrucciones para la computadora
INST2	M	Sirve para dar instrucciones	Lo que le da instrucciones a la computadora
WEB	M	Sólo puede ser una página web	Sitio web
PROG	B	Foco en ser el resultado del proceso de programación	Creado con programación
AUTO	B	Foco en su capacidad de convertir a un autómata de propósito general en uno de propósito específico	Lo que convierte a un autómata de propósito general en uno de propósito específico
PRIM	M	Definición de comando primitivo	Lo que le dice al autómata qué puede hacer
FUNC	R	Mención a funciones	Un conjunto de funciones
IDE	M	Foco en el entorno de desarrollo asociado	Espacio donde se le dan instrucciones a la computadora
EXE+PROG	B	Combinación de EXE y PROG	Lo que el autómata ejecuta y nosotros programamos
ALGO+FUNC	R	Combinación de ALGO y FUNC	Algoritmos y funciones
PROG+CODE	R	Combinación de PROG y CODE	El resultado de un código
O	M	Otras respuestas difíciles de clasificar	Herramienta para realizar algo ; Aplicaciones
NO SE	N	Prefiere no arriesgar	No sé

Cuadro 12: Categorías tabuladas a las definiciones del concepto “algoritmo”

Código	Calificación	Descripción	Ejemplos
DATOS	M	Se refiere a los datos	Números
VAR	M	Se refiere a una variable	Variable
FUNC	M	Se refiere a una función	Una función ...
CODE	M	Se refiere al código	Código
PASOS	M	Foco en los pasos	Secuencia de pasos
LOOP	M	Se refiere a un ciclo	... que se repite
ACCION	M	Se refiere a una acción	Acción programada
PROG	B	Sinónimo de programar	Es lo mismo que programar
ORDEN	M	Se refiere a algo ordenado	Un orden sistemático / aleatorio
SISTEMA	M	Se refiere a un sistema	Es un sistema
INST	M	Se refiere a las instrucciones	Conjunto de instrucciones / de acciones
DATOS+ORDEN	M	Combinación de DATOS y ORDEN	Un conjunto de datos con orden
O	M	Otras respuestas difíciles de clasificar	Una secuencia a seguir según las respuestas del usuario
NO SE	N	Prefiere no arriesgar	No sé

## 4. Resultados

En total participaron 44 personas del curso. Es importante notar que no todas las personas asistieron a todos los encuentros. La encuesta inicial la contestaron 42 participantes y la encuesta final fue respondida por 32 personas. La cantidad de participantes que contestaron ambas encuestas a la vez fue 30. En líneas generales, se observó que a lo largo de las clases todos los participantes pudieron completar las tareas planteadas, y, finalmente, pudieron desarrollar en equipo un proyecto a elección.

### 4.1. Autopercepción respecto a conocimientos en programación

En la encuesta inicial 16 participantes (38 % de los que completaron la encuesta, 36 % del total) dijeron que no sabían programar. Otros 16 dijeron que sí, de los cuales 10 dijeron que sabían Python, 4 Javascript y 2 C++. Hubo 9 que dijeron que sabían programar pero sólo mencionaron HTML y/o Css como lenguajes de programación. La última persona dijo que sabía “un poco” y no mencionó ningún lenguaje particular. Por otro lado, 28 (67 % de los que completaron la encuesta, 64 % del total) participantes habían usado al menos un entorno de programación por bloques, siendo Scratch el más frecuente (con 22 participantes que mencionaron haberlo usado). En resumen, 32 participantes (76 % de los que completaron la encuesta, 72 % del total) tuvieron experiencia previa en programación, ya sea porque respondieron que sabían programar o porque mencionaron haber usado algún entorno de programación por bloques.

En la encuesta final 18 participantes (56 % de los que respondieron la encuesta, 41 % del total) afirmaron saber programar (sus respuestas a la pregunta POST-PROG-END se tabularon como “Sí” o “Sí +”). Otros 6 (19 % de los que respondieron la encuesta, 14 % del total) aseguraron que no sabían programar (sus respuestas a la pregunta POST-PROG-END se

Cuadro 13: Comparación de respuestas a las preguntas POST-PROG-START y POST-PROG-END de la encuesta final.

		POST-PROG-END				
		No	No+	Poco / Base	Sí	Sí+
POST-PROG-START	No	1	4	5	5	0
	Poco / Base	0	0	2	3	0
	Sí	1	0	1	5	5

Cuadro 14: Comparación de respuestas a la pregunta PRE-PROG de la encuesta inicial y a la pregunta POST-PROG-START de la encuesta final.

		POST-PROG-START		
		No	Poco / Base	Sí
PRE-PROG	No	8	2	2
	Poco / Base	1	0	0
	Web	2	1	4
	Sí	3	2	5

tabularon como “No” o “No +”).

En el cuadro 13 se comparan las respuestas a las preguntas de la encuesta final POST-PROG-START y POST-PROG-END (las que hacían referencia a si sabían programar antes y después de tomar el curso, respectivamente). Podemos observar que 13 participantes (41 % de los que respondieron la encuesta, 30 % del total) consideran que tras tomar el curso mejoraron considerablemente sus conocimientos en programación (pasaron de “No” a “Sí”, de “Poco” o de “Base” a “Sí” o de “Sí” a “Sí+”). Otros 9 participantes (28 % de los que respondieron la encuesta, 20 % del total) mejoraron ligeramente (pasaron de “No” a “No+” o de “No” a “Poco” o a “Base”) y a 8 parece no haberles afectado, ya sea porque no sabían antes y siguen sin saber después o porque sabían algo de antes y siguen sabiendo lo mismo después. Hay 2 personas a las que parece que tomar el curso les afectó negativamente ya que dijeron que sabían programar antes de tomar el curso pero una vez que finalizó dejaron de saber o pasaron a saber menos de lo que sabían antes.

Al comparar las respuestas a las preguntas PRE-PROG y POST-PROG-START (ambas referentes a los conocimientos previos pero la primera realizada durante la encuesta inicial y la segunda durante la encuesta final) podemos observar que una gran cantidad de participantes cambió su autopercepción sobre los conocimientos de programación que poseía antes de tomar el curso. Estos resultados se muestran en la tabla 14. De los 30 participantes que completaron ambas encuestas, 8 declararon en la encuesta final menor seguridad respecto a sus conocimientos previos en programación que lo declarado en la inicial (pasaron de “Sí” a “Poco” o a “Base” o de “Sí”, de “Web”, de “Poco” o de “Base” a “No”). Por otro lado, 4 participantes aseguraron en la encuesta final que sabían, en alguna medida, programar antes de tomar el curso (respondiendo con “Sí”, “Base” o “Poco” a POST-PROG-START) a pesar de que en la encuesta inicial lo negaron (respondiendo con “No” a PRE-PROG).

Al cruzar estos datos con las respuestas a la pregunta de si habían usado algún entorno de programación por bloques observamos que en la encuesta inicial hubo 8 personas que negaron saber programar a pesar de haber usado algún entorno de programación por bloques, de las cuales 4 cambiaron de opinión en la encuesta final y aseguraron que sí sabían programar antes de tomar el curso.

De las respuestas a POST-PROG-END, se detectaron 3 casos en que hacen referencia a haber aprendido sólo un lenguaje (como, por ejemplo, “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”) y 8 personas que hacen referencia a haber aprendido conceptos de programación en general (como, por ejemplo, “Creo que tengo al menos una buena base respecto a lo que es programar ya que pude entender todos los conceptos y cómo “unirlos”” o “Siento que entendí más sobre la lógica de programación y de cómo funciona de manera generalizada”). De las respuestas a POST-GAIN, se detectaron 2 casos que hacen referencia a haber aprendido sólo un lenguaje (como, por ejemplo, “Me parecio exelente y me aportó mucha información sobre lo que es el arduino y como se emplea” o “estuve bueno, pensé que veríamos más lenguajes de programación, no solamente trabajar con arduino, sensores y actuadores”) y 7 personas que hacen referencia a haber aprendido conceptos de programación en general (como, por ejemplo, “Bastante bueno y entretenido, creo que me aportó los conceptos básicos de la programación” o “me pareció muy bueno e interesante, me ayudo a entender más el ámbito de la programación”). En resumen, hubo 5 participantes que hicieron referencia a haber aprendido sólo un lenguaje en alguna de las dos preguntas y otros 12 a haber aprendido a programar en general.

Entre las respuestas a la pregunta POST-LACK se encontraron 9 pedidos de aprender otros lenguajes de programación (de los cuales 4 pidieron específicamente Python), 6 pedidos de trabajar con texto o con código y 2 pedidos de aprender algoritmos.

Cuadro 15: Cantidad (y porcentajes) de respuestas en la encuesta inicial a cada concepto según cada calificación.

Concepto	Bien	Regular	Mal	Ns/Nc
Programar	4 (9,52 %)	17 (40,47 %)	12 (28,57 %)	9 (21,42 %)
Algoritmo	0 (0,00 %)	0 (0,00 %)	25 (59,52 %)	17 (40,47 %)
Computadora	3 (7,14 %)	4 (9,52 %)	29 (69,04 %)	6 (14,28 %)
Lenguaje	7 (16,66 %)	12 (28,57 %)	14 (33,33 %)	9 (21,42 %)
Programa	3 (7,14 %)	6 (14,28 %)	20 (47,61 %)	13 (30,95 %)
Total	17 (8,09 %)	39 (18,57 %)	100 (47,61 %)	54 (25,71 %)

## 4.2. Evaluación de conceptos

Con respecto a las definiciones de los conceptos, se observó una importante inclinación hacia una visión operacional de la programación. Las categorías de respuestas más frecuentes para la definición de “programar” fueron aquellas que ponían el foco en el código (11 respuestas), aquellas que ponían el foco en las instrucciones (10 respuestas) y aquellas que ponían el foco en el software resultante (8 respuestas). En el caso del concepto “computadora”, 23 respuestas pusieron el foco en el hardware (hablando de “dispositivo electrónico” o de “equipo tecnológico”), de las cuales 7 mencionaron también el software como contraparte (como, por ejemplo, “sistema de hardware y software”). Entre las respuestas a la definición de “lenguaje de programación” se encontraron 9 que ponían el foco en el hecho de que eran lenguajes. De entre esas, 3 hicieron referencia además al hecho de que sirven para programar y una más al hecho de que sirven para escribir código. Las 5 restantes no aportaron información adicional. Por otro lado, se encontraron 8 respuestas que ponían el foco en escribir código (además de la ya mencionada entre las 9 anteriores). En las respuestas a la definición de “programa” vuelven a predominar aquellas que ponen el foco en el código, con 8 apariciones. Las siguientes categorías más frecuentes son aquellas que ponen el foco en las instrucciones y aquellas que parecen estar definiendo “proceso” en lugar de “programa” (con 3 apariciones cada una). Las respuestas a la definición de “algoritmo” fueron mucho más variadas. Las más frecuentes fueron las que ponían el foco en los pasos (con 4 apariciones). En segundo puesto empataron aquellas que hacían referencia a un ordenamiento, aquellas que hacían referencia a un ciclo o repetición, aquellas que hacían referencia a código y aquellas que hacían referencia a “un sistema” (con 3 apariciones cada una).

La cantidad de respuestas tabuladas como “No sé” fueron 9 para la definición de “programar”, 6 para la definición de “computadora”, 9 para la definición de “lenguaje de programación”, 13 para la definición de “programa” y 17 para la definición de “algoritmo”. La tabla 15 muestra la cantidad de respuestas (junto a sus respectivos porcentajes) a cada definición según cada una de las cuatro posibles calificaciones realizadas por el equipo docente. En la tabla 16 se muestra cómo se modificaron las percepciones respecto a los conceptos durante el curso, a partir de cómo fueron tabuladas las definiciones provistas en la encuesta inicial y en la encuesta final. Sólo se cuentan las respuestas de quienes estuvieron presentes en ambas encuestas. Los porcentajes no suman 100 porque se muestran únicamente las categorías que son de interés para el estudio. Además, algunos pares de categorías (como CODE y SOFT) tienen intersección no vacía.

En todos los conceptos se reduce la cantidad de respuestas “Ns/Nc”. Las definiciones para el concepto “programar” tabuladas como DECIR (aquellas que ponen el foco en el hecho de comunicar una solución) aumentaron y las tabuladas como CODE (aquellas que ponen el foco en escribir código) y como SOFT (aquellas que ponen el foco en el software resultante del proceso) disminuyeron. Sin embargo, también aumentaron las definiciones tabuladas como “INST” (aquellas que ponen el foco en las instrucciones). Para el concepto “computadora” disminuyeron las definiciones tabuladas como HARD (aquellas que ponen el foco en lo electrónico o lo tecnológico) y aumentaron las tabuladas como PROG (aquellas que ponen el foco en el hecho de ser programable) y como COMPU (aquellas que ponen el foco en su tarea de computar funciones y procesar información). Para el concepto “lenguaje de programación” aumentaron tanto las definiciones tabuladas como PC (aquellas que ponen el foco en que es el idioma de las computadoras) como las tabuladas como COMM (aquellas que ponen el foco en que permite la comunicación entre personas y computadoras). Además, muchas de las concepciones erróneas (como confundir el lenguaje con el entorno de desarrollo asociado o con un compilador) que aparecieron en la encuesta inicial desaparecieron en la final. En las respuestas al concepto de “programa” no hubo grandes cambios aunque sí desaparecieron varias de las concepciones erróneas que aparecieron en la encuesta inicial (como confundir el programa con el proceso en ejecución o pensar únicamente en páginas web). En el caso del concepto “algoritmo” aumentaron las respuestas tabuladas como PASOS (aquellas que lo definen como una secuencia de instrucciones). También desaparecieron varias concepciones erróneas de la encuesta inicial como asociarlo a una función, a un ciclo que se repite, a un orden, a un sistema o incluso a código.

Al observar las calificaciones a las respuestas se observan importantes mejoras en la encuesta final respecto a la encuesta inicial en la mayoría de los conceptos. En total, la cantidad de respuestas calificadas con Bien subió de 12 (8 %) a 40 (27 %) y la cantidad de respuestas calificadas con Mal bajó de 65 (43 %) a 47 (31 %). El concepto “computadora” es el que presenta la mejora más notable, siendo el que mayor cantidad de respuestas calificadas como Mal tuvo en la encuesta inicial así como el que mayor cantidad de respuestas calificadas como Bien tuvo en la encuesta final. Además, también fue el concepto que más subió en cantidad de respuestas calificadas como Bien y el que más bajó en cantidad de respuestas calificadas como Mal. En el caso del concepto “algoritmo”, no hubo cambios significativos. Por otro lado, también se aprecia un importante decremento en la cantidad de respuestas calificadas como Ns/Nc, tanto en total (pasando de 42 en la encuesta inicial a 19 en la encuesta final) como en cada uno de los conceptos individualmente. Toda esta información se puede observar en los gráficos de la figura 1.

Cuadro 16: Cantidad (y porcentajes) de respuestas en cada encuesta a cada concepto según cómo fueron tabuladas.

Programar			Lenguaje de Programación		
	PRE	POST		PRE	POST
DECIR	0 (0,00 %)	8 (26,66 %)	PC	1 (3,33 %)	8 (26,66 %)
CODE	6 (20,00 %)	3 (10,00 %)	COMM	1 (3,33 %)	6 (20,00 %)
SOFT	7 (23,33 %)	1 (3,33 %)	Ns/Nc	6 (20,00 %)	1 (3,33 %)
INST	7 (23,33 %)	16 (53,33 %)			
Ns/Nc	7 (23,33 %)	0 (0,00 %)			
Computadora			Programa		
	PRE	POST		PRE	POST
HARD	19 (63,33 %)	3 (10,00 %)	Ns/Nc	10 (33,33 %)	4 (13,33 %)
PROG	1 (3,33 %)	8 (26,66 %)			
COMPU	1 (3,33 %)	10 (33,33 %)			
Ns/Nc	5 (16,66 %)	2 (6,66 %)			
Algoritmo					
	PRE	POST			
PASOS	2 (6,66 %)	9 (30,00 %)			
Ns/Nc	14 (46,66 %)	12 (40,00 %)			

Las mejoras no sólo se aprecian en la visión global de las respuestas sino también al comparar los pares de respuestas de una misma persona. En la tabla de la figura 2 se muestran los cambios en las calificaciones a las respuestas individuales, tanto en total como para cada categoría. En particular, se destaca que de las 42 respuestas calificadas como N en la encuesta inicial, 29 se calificaron de otra forma en la encuesta final (10 pasaron a B, 7 a R y 12 a M). Por otra parte, de las 19 respuestas calificadas como N en la encuesta final, las únicas 6 que habían sido calificadas de otra forma en la encuesta inicial habían obtenido la calificación M. El total de respuestas que mejoraron considerablemente (de M a B o de N a B) fue 27 (18 %). El total de respuestas que mejoraron un poco (de M a R, de R a B o de N a R) fue 31 (21 %). El total de respuestas que empeoraron un poco (de B a R o de R a M) fue 8 (5 %). El total de respuestas que empeoraron considerablemente (de B a M) fue 4 (3 %).

Por último, en lo respectivo al trabajo de los estudiantes, se aprecia una mejora cualitativa en la autonomía, en tanto pudieron cumplir con las tareas solicitadas en cada clase, aun las que implicaban una expresión personal individual (e.g., el armado una canción con el buzzer), además de poder desarrollar el proyecto final a elección.

## 5. Discusión

De acuerdo al desarrollo del curso, se pudo ver que los participantes adquirieron habilidades de programación y resolución de problemas durante las clases, y que pudieron completar el trabajo final. Adicionalmente, se notó una mejora en los conceptos relacionados a programación entre las dos encuestas, lo que podría explicarse, en parte, a lo aprendido durante el curso. En particular, se notó una mejora en la definición del concepto de algoritmo, pese a que a lo largo de las clases no se trabajó explícitamente con su definición, pero sí se realizaron varios algoritmos como parte de las soluciones a los problemas planteados. Aún así, hubo 2 participantes que pidieron estudiar algoritmos como respuesta a qué le faltó al curso. Omitir la palabra “algoritmo” fue intencional, en tanto no queríamos acotar la visión de que programar es sólo definir una “secuencia de comandos”, pero, quizás, esto influyó negativamente en la percepción de los estudiantes.

En lo respectivo a la autopercepción de los conocimientos, se notó que la mayoría de los estudiantes notaron una mejora en sus conocimientos a partir de lo que habían puesto en la primera encuesta, y que algunos pocos indicaron que antes sabían y luego del curso no. Esto es un punto de mejora, ya que, más allá de que alguien podría darse cuenta que antes creía que sabía y, finalmente, no era cierto, se busca que todos los participantes salgan del curso con una mejor autopercepción de lo que saben.

Dentro de los comentarios recibidos, una persona indicó que sólo había aprendido un lenguaje (i.e., Arduino) y no a programar en general. Esto podría deberse a que, justamente, usaron sólo una herramienta a lo largo del curso, y, aunque otros participantes no lo comentaron, quizás podría ser una percepción más general. En cualquier caso, es algo común que los estudiantes de nivel medio estén ávidos de aprender nuevos lenguajes de programación, porque es habitual que tengan en cuenta como una variable a analizar la cantidad de lenguajes que se ven en las posibles carreras universitarias que podrían elegir para iniciar sus estudios, por lo que esto requiere mayor análisis en futuras ediciones.

Por último, luego del análisis de los resultados de la encuesta evaluamos que sería necesario preguntar por separado si saben programar y, por otro lado, qué lenguajes conocen. Esto en especial es importante en tanto a veces los estudiantes no conocen un lenguaje con nombre conocido, como Python, pero sí tienen experiencia en programación por medio de la programación por bloques o por haber utilizado algún otro lenguaje visual o tangible. Además, permitiría poner más claridad sobre la creación de páginas web, que es un contenido que suele aparecer en las escuelas secundarias, pero que, en general, no se ve desde el punto de vista de la programación, sino que sólo se presenta con el fin de crear contenidos estáticos.

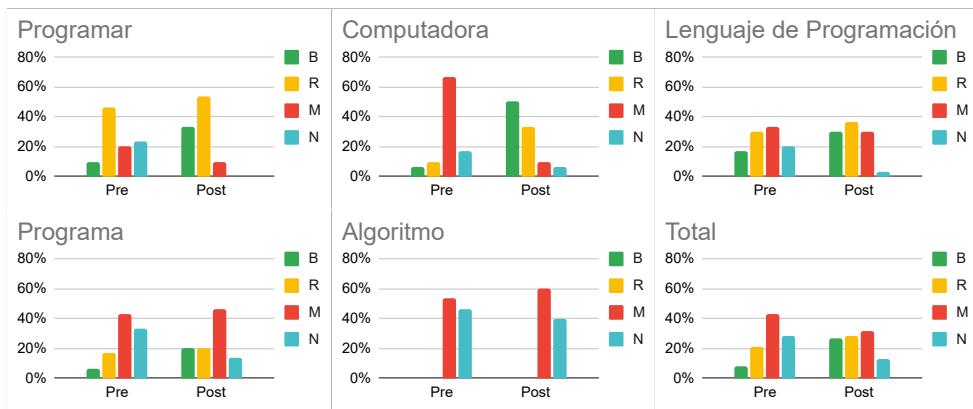


Figura 1: Comparación de las calificaciones otorgadas a las respuestas a cada uno de los conceptos en cada encuesta.

Programar		POST				Computadora		POST				Lenguaje		POST									
		B	R	M	N		B	R	M	N		B	R	M	N		B	R	M	N			
PRE	POST	0	2	1	0	Total	2	0	0	0	Total	2	2	1	0	Total	4	4	4	0	Total		
B	0	2	1	0	3	R	5	7	1	0	13	M	1	4	1	0	6	N	4	3	0	0	7
R	5	7	1	0	13	M	9	8	2	1	20	N	3	0	1	1	5	Total	15	10	3	2	2
M	1	4	1	0	6	N	3	0	1	1	5	Total	9	11	9	1	6	Total	9	11	9	1	6
N	4	3	0	0	7	Total	10	16	3	0	38	Total	10	16	3	0	38	Total	9	11	9	1	6
PRE	POST	0	2	1	0	Total	2	0	0	0	Total	0	0	0	0	Total	0	0	0	0	Total		
B	0	0	2	0	2	R	1	2	2	0	5	M	4	2	7	0	13	N	1	2	3	4	10
R	1	2	2	0	5	M	0	0	11	5	16	N	0	0	7	7	14	Total	6	6	14	4	19
M	4	2	7	0	13	N	0	0	7	7	14	Total	0	0	18	12	30	Total	40	43	47	19	121
N	1	2	3	4	10	Total	0	0	18	12	30	Total	0	0	18	12	30	Total	40	43	47	19	121

Figura 2: Comparación de las calificaciones otorgadas a las respuestas individuales a cada uno de los conceptos en cada encuesta.

## 6. Conclusiones

Es muy valiosa la utilización de placas Arduino, ya que, más allá de trabajar con programación física, permite desarrollar proyectos de diverso tipo, y que no sean solamente los planteados por las empresas que fabrican los kits, además de que abre la puerta para trabajar con más conceptos relacionados a otras ciencias, como, por ejemplo, la física, ya que se pueden pensar proyectos donde se utilicen sensores y actuadores específicos para ciertas áreas. Sin embargo, para el uso de Arduino requiere un mayor conocimiento técnico sobre los materiales que se compran, a diferencia de lo fácil que es adquirir un kit de robótica completo para trabajar en el aula, además de que se requieren mayores conceptos de electrónica básica para que las cosas funcionen y que no se dañen los elementos de trabajo. Finalmente, un factor importante para que no esté totalmente extendido el uso de Arduino en el nivel introductorio de programación es la falta de herramientas de uso masivo que sean simples e intuitivas y sirvan para la programación para esta plataforma.

Es necesario analizar alternativas para el diseño de cursos introductorios de programación. Más allá de que los resultados fueron buenos, se cree que es mejor realizar un curso con múltiples herramientas, en tanto esto suma al objetivo de que los estudiantes salgan del curso con la autopercepción de que, efectivamente, aprendieron a programar en general, y no sólo aprendieron a usar una herramienta determinada (que ni siquiera se parece a los lenguajes más conocidos, como Python, sino que utiliza bloques para generar los programas). Esta visión, donde cada herramienta/lenguaje de programación aparece exactamente para resolver cierto tipo de problemas donde mejor se adapta, y se presentan distintos tipos de herramientas a lo largo de un curso, cada una con sus ventajas y desventajas, la denominamos “enfoque heterólogo”, y creemos que permite un aprendizaje más robusto.

Complementariamente, para contenidos como la programación, donde hay múltiples lenguajes, bibliotecas y frameworks a aprender, es muy importante que los estudiantes salgan de un curso introductorio con la confianza de que tiene un mínimo de habilidades generales, de forma de que los motive a aprender nuevas cosas y, por sobre todo, continuar una práctica sostenida más allá de lo visto en el curso.

## 7. Trabajo futuro

Se planifica realizar el diseño de un nuevo curso introductorio a la programación para estudiantes del nivel secundario, pero utilizando múltiples herramientas/lenguajes, donde cada una aparece en el curso en tanto resuelve una limitación de herramientas/lenguajes anteriores. Por ejemplo, ante la necesidad de trabajar con eventos, se utiliza MIT App Inventor.

Con el fin de mejorar la autopercepción de los aprendizajes por parte de los estudiantes participantes, se desarrollarán prácticas de metacognición específicas para el trabajo en cursos introductorios de programación, en tanto creemos que la conciencia que tenga un estudiante de lo que sabe tiene impacto directo sobre el éxito o no de curso, más allá de que la persona pueda realizar todas las tareas presentadas en el mismo. Con respecto a esto último, quedaría ver hasta qué punto afecta a una persona el hecho de, aún pudiendo realizar las mismas tareas que otra, tenga la autopercepción de que no sabe de un tema (en este caso, a programar en forma general, y no sólo con un lenguaje determinado). Nuestra hipótesis es que afecta totalmente, y que una persona que no tiene suficiente confianza sobre sí mismo no se verá en condiciones de aprender más por su cuenta. Esto es esencial en áreas como la programación, donde el aprendizaje autónomo de plataformas y lenguajes es un práctica habitual.

Finalmente, se avanzará con la descripción del enfoque heterólogo propuesto, tanto para diferentes contextos (i.e., no sólo para nivel secundario) y para contenidos introductorios y más avanzados.

## 8. Agradecimientos

A Gastón Pérez, del CEFIEC (Exactas, UBA), por su generosidad en la revisión del trabajo y sus valiosas sugerencias.

## Referencias

- [1] R. Mason, u. undefined undefined, B. A. Becker, T. Crick y J. H. Davenport. «A Global Survey of Introductory Programming Courses». En: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education* (2024), págs. 799-805. DOI: 10.1145/3626252.3630761. URL: <https://dl.acm.org/doi/pdf/10.1145/3626252.3630761>.
- [2] J. Dabbah, M. Garzón, M. J. Gomez, M. C. Martínez y P. E. Martínez López. *Propuesta curricular para la inclusión de las Ciencias de la Computación en la educación obligatoria de Argentina*. Fundación Sadosky, Buenos Aires, 2023. URL: <https://curriculum.program.ar/>.
- [3] H. Abelson, N. Goodman y L. Rudolph. *LOGO Manual*. Artificial Intelligence Lab, Massachusetts Institute of Technology, 1974. URL: <http://hdl.handle.net/1721.1/6226>.
- [4] L. P. Flannery, B. Silverman, E. R. Kazakoff, M. Umaschi Bers, P. Bontá y M. Resnick. «Designing ScratchJr: support for early childhood learning through computer programming». En: *Proceedings of the 12th International Conference on Interaction Design and Children* (2013), págs. 1-10.
- [5] L. Morgado, M. G. Cruz y K. Kahn. «Preschool cookbook of computer programming topics». En: *Australasian Journal of Educational Technology* 26.3 (2010), págs. 309-326. ISSN: 1449-5554. DOI: 10.14742/ajet.1077. URL: <http://ajet.org.au/index.php/AJET/article/view/1077>.
- [6] L. Mannila, M. Peltomaki y T. Salakoski. «What About a Simple Language? Analyzing the Difficulties in Learning to Program». En: *Computer Science Education* 16.3 (2006), págs. 211-227.
- [7] M. Resnick, J. H. Malone, A. Monroy Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman e Y. Kafai. «Scratch: Programming for all». En: *Communications of the ACM* 52.11 (2009), págs. 60-67. DOI: 10.1145/1592761.1592779.
- [8] W. P. Dann, S. Cooper y R. Pausch. *Learning to program with Alice*. Prentice-Hall: Upper Saddle River, NJ, 2005.
- [9] E. W. Dijkstra. «On the cruelty of really teaching computing science». En: *Communications of the ACM* 32.12 (1989), págs. 1398-1404.
- [10] S. Magnenat, J. Shin, F. Riedo, R. Siegwart y M. Ben-Ari. «Teaching a core cs concept through robotics». En: *Proceedings of the 2014 Conference on Innovation and technology in computer science education* (2014), págs. 315-320.
- [11] A. Druin y J. A. Hendler. *Robots for kids: exploring new technologies for learning*. Morgan Kaufmann, 2000.
- [12] E. R. Kazakoff, A. Sullivan y M. Umaschi Bers. «The effect of a classroombased intensive robotics and programming workshop on sequencing ability in early childhood». En: *Early Childhood Education Journal* 41.4 (2013), págs. 245-255.
- [13] L. P. Flannery y M. Umaschi Bers. «Let's dance the 'robot hokey-pokey': children's programming approaches and achievement throughout early cognitive development». En: *Journal of Research on Technology in Education* 46.1 (2013), págs. 81-101.
- [14] W. Yang, D. T. K. Ng y H. Gao. «Robot programming versus block play in early childhood education: Effects on computational thinking, sequencing ability, and self-regulation». En: *British Journal of Educational Technology* 53.6 (2022), págs. 1817-1841. ISSN: 14678535. DOI: 10.1111/bjet.13215.
- [15] F. Kalelioglu y S. Sentance. «Teaching with physical computing in school: the case of the micro:bit». En: *Education and Information Technologies* 25.4 (2020), págs. 2577-2603. ISSN: 1573-7608. DOI: 10.1007/s10639-019-10080-8.

- [16] A. Juškevičienė, G. Stupurienė y T. Jevsikova. «Computational thinking development through physical computing activities in STEAM education». En: *Comput Appl Eng Educ* 2021.29 (2020), págs. 175-190. DOI: 10.1002/cae.22365.
- [17] N. Fanchamps, L. Slangen, M. Specht y P. Hennissen. «Effect of SRA-programming on computational thinking through different output modalities». En: *Journal of Computers in Education* 10.2 (2023), págs. 433-462. ISSN: 2197-9995. DOI: 10.1007/s40692-022-00236-w. URL: <https://doi.org/10.1007/s40692-022-00236-w>.
- [18] D. Parsons y P. Haden. «Programming osmosis: Knowledge transfer from imperative to visual programming environments». En: *Proceedings of The Twentieth Annual NACCD Conference* (2007), págs. 209-215.
- [19] K. Powers, S. Ecott y L. M. Hirshfield. «Through the looking glass: teaching CS0 with Alice». En: *Proceedings of the 38th ACM Technical Symposium on Computer Science Education* (2007), págs. 213-217.
- [20] V. Basel. «Uso de domótica como herramienta para la enseñanza de programación». En: *Actas Jornadas Argentinas de Didáctica de las Ciencias de la Computación 2021* (2021), págs. 116-125. URL: <https://jadicc2021.program.ar/wp-content/uploads/2022/03/actas-jadicc-2021.pdf>.
- [21] D. J. Malan y H. H. Leitner. «Scratch for budding computer scientists». En: *Proceedings of the 38th ACM Technical Symposium on Computer Science Education* (2007), págs. 223-227.
- [22] B. Moskal, D. Lurie y S. Cooper. «Evaluating the effectiveness of a new instructional approach». En: *Proceedings of the 35th ACM Technical Symposium on Computer Science Education* (2004), págs. 75-79.
- [23] J. M. Massa, M. C. Leonardi, M. V. Mauco, L. Felice y J. H. Doorn. «Hacia un Análisis de Contenidos y Lenguajes de Programación en Materias iniciales de Programación de Carreras de Ciencias de la Computación». En: *Actas Jornadas Argentinas de Didáctica de las Ciencias de la Computación 2023* (2023).
- [24] A. Sanzo, F. Schapachnik, P. Factorovich y F. Sawady O'Connor. «Pilas Bloques: A scenario-based children learning platform». En: *2017 Twelfth Latin American Conference on Learning Technologies* (2017), págs. 1-6.
- [25] P. E. Martínez López, F. Aloisio, D. A. Ciolek, F. Martínez, D. Pari y P. Tobia. *Ciencias de la computación para el aula: 1er. ciclo de secundaria (libro para docentes)*. Ciudad Autónoma de Buenos Aires, Argentina: Fundación Sadosky, 2019. URL: [http://program.ar/descargas/cc\\_para\\_el\\_aula-1er\\_ciclo\\_secundaria.pdf](http://program.ar/descargas/cc_para_el_aula-1er_ciclo_secundaria.pdf).
- [26] C. Banchoff, H. Czemerinski, J. Dabbah, F. Frizzo, C. Queiruga y P. Venosa. *Ciencias de la computación para el aula: 2do. ciclo de secundaria (libro para docentes)*. Ciudad Autónoma de Buenos Aires, Argentina: Fundación Sadosky, 2019. URL: [https://program.ar/descargas/cc\\_para\\_el\\_aula-2do\\_ciclo\\_secundaria.pdf](https://program.ar/descargas/cc_para_el_aula-2do_ciclo_secundaria.pdf).
- [27] G. P. Fernández, M. B. Ticona Oquendo y C. Cossio-Mercado. «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* (2021), págs. 32-44.
- [28] M. Lobos, S. Bast y G. Astudillo. «Evaluación de calidad de Entornos de Desarrollo Integrado de Robótica Educativa». En: *Actas JADiCC 2022* (2022), págs. 156-167.
- [29] G. P. Fernández y C. Cossio-Mercado. «AeIE: a versatile tool for teaching programming and robotics using Arduino». En: *50a Conferencia Latinoamericana de Informática (CLEI)* (2024).
- [30] C. Cossio-Mercado y G. P. Fernández. «PRENDER: Una propuesta didáctico-pedagógica para la enseñanza de las Ciencias de la Computación». En: *Jornadas Argentinas de Didáctica de las Ciencias de la Computación 2023* (2023). URL: [https://www.researchgate.net/publication/380173508\\_PRENDER\\_Una\\_propuesta\\_didactico-pedagogica\\_para\\_la\\_enseñanza\\_de\\_las\\_Ciencias\\_de\\_la\\_Computacion](https://www.researchgate.net/publication/380173508_PRENDER_Una_propuesta_didactico-pedagogica_para_la_enseñanza_de_las_Ciencias_de_la_Computacion).
- [31] M. Stone Wiske. *La enseñanza para la comprensión*. Buenos Aires, Argentina: Paidós, 1999.
- [32] R. Anijovich y G. Cappelletti. *La evaluación como oportunidad*. Buenos Aires, Argentina: Paidós, 2017.
- [33] J. J. Schwab. «What Do Scientists Do?» En: *Behavioral Science* 5.1 (1960), págs. 1-27.
- [34] J. J. Schwab. «The teaching of science as enquiry». En: *The teaching of science*. Ed. por J. J. Schwab y P. F. Brandwein. Cambridge, Harvard University Press, 1962, págs. 1-103.
- [35] J. I. Pozo e Y. Postigo. «La solución de problemas como contenido procedimental en la educación obligatoria». En: *La solución de problemas*. Ed. por J. I. Pozo, M. del Puy Pérez, J. Domínguez, M. Á. Gómez e Y. Postigo. Editorial Santillana, 1994.
- [36] P. E. Martínez López. *Las bases conceptuales de la programación. Una nueva forma de aprender a programar*. EBook (1era ed.). La Plata, el autor, 2013. ISBN: 978-987-33-4081-9.

# ¿Cómo llevar lo que sabemos de Metacognición a las clases de programación?

Christian Cossio-Mercado<sup>1</sup>, Gonzalo Pablo Fernández<sup>2</sup> y Gastón Pérez<sup>3</sup>

<sup>1</sup> Universidad de Buenos Aires. Facultad de Ciencias Exactas y Naturales. Departamento de Computación. Buenos Aires, Argentina.

<sup>2</sup> Universidad Nacional de Quilmes. Departamento de Ciencia y Tecnología. Buenos Aires, Argentina.

<sup>3</sup> Universidad de Buenos Aires. Facultad de Ciencias Exactas y Naturales. Instituto de Investigaciones CeFIEC. Buenos Aires, Argentina.

cossio@dc.uba.ar, gpfernandez@dc.uba.ar, gastonperez@ccpems.exactas.uba.ar

## Resumen

En los últimos años se hizo común el hablar del Pensamiento Computacional, aunque no está del todo claro qué es ni hay certezas de cómo enseñarlo. Sin embargo, mucha gente afirma que, junto con otras habilidades, desarrolla funciones metacognitivas transferibles a otros ámbitos de la vida, principalmente asociadas a la resolución de problemas. Adicionalmente, se sabe que las actividades de metacognición mejoran la comprensión de los temas y el aprendizaje en general. Muchos estudios miden la adquisición de habilidades al enseñar a programar, pero pocos estudian esta transferencia de conocimientos y, menos aún, el desarrollo de la metacognición en los estudiantes. En este trabajo proponemos agregar a las clases de programación actividades de metacognición explícitas, de forma que los estudiantes desarrollen estas habilidades, en lugar de esperar que se desarrolle espontáneamente sólo por el hecho de estar escribiendo código en una computadora. Así, tomamos un curso corto de iniciación a la programación para estudiantes de secundaria y se agregaron reflexiones metacognitivas por medio de preguntas al final de cada clase. Encontramos que las actividades propuestas efectivamente permitieron a los estudiantes reflexionar sobre su propio conocimiento y sobre las dificultades que les surgieron al realizar las actividades de programación, así como, también, fortalecer la confianza respecto a los conocimientos adquiridos durante el curso. Este trabajo puede servir a docentes de programación para saber cómo incorporar actividades de metacognición en sus clases y, en forma más general, cómo propiciar espacios para la reflexión y la transferencia de conocimiento entre distintos contextos y cursos.

**Palabras clave:** Metacognición, Programación

## 1. Introducción

Considerando el continuo cambio tecnológico de nuestras sociedades, una de las habilidades clave de un programador<sup>1</sup>, es poder aprender autónomamente durante el resto de su vida independientemente del tipo de lenguaje de programación. Esta habilidad involucra ser consciente de los propios procesos de pensamiento, de las estrategias de las que se dispone y de su utilidad, pero también regular los procesos de aprendizaje, planificar mecanismos de acción frente a una tarea, monitorear dichos mecanismos y evaluar su utilidad para próximas situaciones [1]. Cuando se aprende a programar, se espera que el estudiante no sólo escriba código fuente, sino que también logre planificar cómo resolver esta tarea, organizar el programa, seguir el progreso y pueda evaluarla a la luz de los objetivos propuestos para el mismo [2, 3, 4].

Recientes investigaciones en el área de la enseñanza de la programación muestran que el desarrollo de estas habilidades puede potenciarse a través de propuestas que involucren la metacognición (MC) [2, 5, 6, 3, 4, 7, 8]. La MC es la reflexión explícita sobre el propio pensamiento, que involucra una reflexión sobre el conocimiento de la cognición, pero también sobre la regulación de la cognición [9]. Respecto de la primera, implica una reflexión sobre el conocimiento de la propia cognición. Esto incluye el conocimiento sobre qué se conoce, cuáles son las estrategias de pensamiento que se poseen y en qué momento son utilizables. Respecto de la regulación, esto involucra habilidades de planificación, monitoreo y evaluación de la propia cognición. Las/os investigadores en el área de la didáctica, sugieren que fomentar la MC en las clases ayudaría a los estudiantes a mejorar sus estrategias de resolución de problemas, desarrollar nuevas así como construir autonomía [10, 1, 11].

Para desarrollar la MC en las clases existen múltiples estrategias, pero todas ellas comparten la necesidad de explicitar el pensamiento y volverlo objeto de reflexión [12, 13]. La idea fundamental es permitir a los estudiantes transitar desde un aprendizaje implícito, característico de las prácticas tradicionales de enseñanza, donde las decisiones tomadas por los estudiantes no son conscientes ni reflexivas, a un aprendizaje explícito, caracterizado por una conciencia y regulación del uso de estrategias de aprendizaje, así como de sus modos de pensar. Lograr esto en las clases presenta ciertas dificultades, tales como que los estudiantes están acostumbrados a una enseñanza de tipo tradicional, basada en la transmisión-recepción de información, que supone un aprendizaje predominantemente conceptual, memorístico, descontextualizado, crítico e irreflexivo, propiciando un rol pasivo en el estudiante en cuanto a su aprendizaje. Es de esta manera que en nuestra tradición escolar se asume que, por un lado, el aprendizaje de las estrategias metacognitivas no es necesario, ya que basta con

<sup>1</sup>En el escrito se decidió utilizar el género masculino para una fácil lectura. Sin embargo, entendemos que esta forma de escribir deja por fuera otros géneros conocidos y por conocer.

acumular saberes para que estos transmuten y generen la capacidad de usarlos [14]. Dado este marco tradicional en el que se enmarcan la mayoría de las prácticas de enseñanza, es posible encontrarse con estudiantes que no logren diferenciar qué saben o aprendieron de lo que no saben o necesitan aprender. Frente a esta situación, no puede esperarse entonces que logren emplear estrategias metacognitivas más avanzadas como planificar, evaluar su aprendizaje o emplear estrategias más eficientes de aprendizaje [13]. Dada esta dificultad, los investigadores del área sugieren ofrecer más autonomía a los estudiantes, aumentando la demanda metacognitiva a medida que se avanza en las tareas a realizar. Se espera que, además de ceder de a poco el control al estudiante sobre sus propios modos de aprendizaje, logren que las estrategias utilizadas sean internalizadas y utilizadas de manera independiente, y hasta cierto punto, descontextualizadas [15].

Evaluuar el desarrollo de la MC en los estudiantes puede realizarse a partir de múltiples instrumentos. Por ejemplo, antes o después de una tarea se pueden utilizar tests como *Judgments of Learning (JoL)* que permite conocer juicios metacognitivas sobre la propia capacidad para codificar y retener información, *Feeling of Knowing (FoK)* que evalúa la habilidad de monitorear el contenido de la memoria y la capacidad para buscar en la memoria o *Ease of Learning (EoL)* que mide la habilidad de monitorear la dificultad relativa del proceso de comprensión [16, 17]. Estos instrumentos no están enfocados es un área específica del conocimiento, sino que son más bien generales, aunque también existen otros test que sí permiten caracterizar las estrategias que los estudiantes ponen en juego al aprender a programar [18]. Más allá de esto, dichos test ofrecen información al docente de un momento determinado del aprendizaje, evaluando una percepción del conocimiento, dejando por fuera otros aspectos de la MC. Es así que, durante el aprendizaje, se pueden utilizar otro tipo de herramientas que ofrezcan información más descriptiva del proceso, tales como entrevistas, protocolos de pensamiento en voz alta, elaboración de bases de orientación, autoreportes de aprendizaje [12, 19].

De acuerdo a la revisión de prácticas y bibliografía encontramos que en las aulas de programación no se están aprovechando las prácticas de metacognición conocidas. Además, encontramos poca investigación empírica sobre la relación entre la metacognición y la enseñanza de la programación [2, 20, 8, 21]. En este trabajo se proponen prácticas y algunos lineamientos para aplicar metacognición en las clases de programación. Se toma como base el análisis de resultados de una propuesta de curso introductorio que incluyó actividades metacognitivas. Además, se definen algunas prácticas recomendadas, sugerencias y herramientas para la incorporación de la metacognición en cursos de programación.

## 2. Desafíos y dificultades en la enseñanza y el aprendizaje de la programación

Durante la enseñanza y el aprendizaje de la programación existen algunos desafíos y dificultades a las que se enfrentan estudiantes y docentes, respectivamente. Entre ellas podemos destacar algunas que son relevantes para este trabajo.

### 2.1. Resolver problemas vs. escribir en código

Una de las dificultades esenciales en la enseñanza y aprendizaje de programación es cómo ampliar la perspectiva de los programadores desde una visión meramente de generación de código (lo que en términos comunes se le suele llamar “picar código”) a un enfoque de resolución de problemas. La resolución de problemas lo asociamos más al término “programación”, donde se incluyen varias prácticas y conceptualizaciones, como el análisis del problema, la descomposición en partes más simples, y, entre otras cosas, poder determinar qué herramientas algorítmicas y otros recursos se pueden utilizar, mientras que “escribir código” lo asociamos sólo a la construcción de código que cumple ciertas funcionalidades.

### 2.2. Dificultades al construir código

Aunque sostengamos que en un curso de iniciación en programación el foco de la enseñanza debe estar en las estrategias de resolución de problemas y no en la construcción del código necesario para resolverlo, también es cierto que escribir código, y realizar pruebas sobre él, suele ser muy útil en el proceso de programación, en tanto permite tener realimentación inmediata sobre la estrategia propuesta. Esto último es relevante para el aprendizaje, ya que permite validar la solución propuesta y construir sobre el error, desde un enfoque de la programación interactiva [22].

En este marco, identificamos 4 categorías de dificultades, que detallamos a continuación:

- 1. Dificultades en relación a la estrategia de resolución para el problema en cuestión:** El primer paso para llegar al código es determinar cuál va a ser la estrategia que se va a usar. Al enseñar a programar se suele sugerir la metodología de trabajo *top-down* que consiste en plantear primero el problema principal a resolver completo y dividirlo en subproblemas más chicos y manejables. Al escribir el código, esto se traduce en crear un procedimiento o una función cuyo nombre haga referencia al problema general y que su cuerpo se componga de otros procedimientos y funciones, cada una correspondiendo a una de las subpartes de la descomposición del problema propuesta, tales que al combinarlas resuelven el problema principal. Las dificultades de esta categoría hacen referencia a dividir el problema en subtareas, a determinar la responsabilidad de cada subtarea y a definir cómo combinar las subtareas para obtener una solución al problema completo.
- 2. Dificultades en relación a la sintaxis del lenguaje:** Una vez definida la estrategia se deben escribir sus partes, tanto las subtareas como los otros elementos del lenguaje que permiten combinarlas. En este punto suelen aparecer dificultades ligadas a cómo se escriben tales elementos y cómo se combinan entre sí. Los comandos pueden combinarse

con otros comandos mediante una secuenciación o mediante una anidación. Tanto los comandos como las expresiones pueden combinarse con otras expresiones mediante el pasaje de argumentos. En la mayoría de los lenguajes de programación basados en texto la composición mediante secuenciación se hace con un carácter de salto de línea o con un punto y coma; la composición basada en anidación se hace usando llaves (o, en el caso de Python, con dos puntos y agregando sangría en el cuerpo) y la composición mediante pasaje de argumentos se hace usando paréntesis. En los entornos basados en bloques estas composiciones vienen definidas por la forma de los conectores de los bloques, usando el conector tipo “diente” para secuenciación, el conector tipo “boca” para anidación y el conector tipo “oreja” para pasaje de argumentos. Las pistas visuales que proveen los entornos de programación por bloques reducen las dificultades asociadas a la sintaxis, aunque no las eliminan por completo, ya que en las primeras etapas del aprendizaje no está clara aún la distinción entre comandos y expresiones, lo que puede llevar a intentos fallidos por hacer composiciones inválidas (e.g., una anidación entre un comando y una expresión o un pasaje de argumentos entre comandos).

3. **Dificultades en relación a la semántica de los elementos del lenguaje:** A la par de las dificultades sintácticas surgen las dificultades semánticas, asociadas a comprender el significado (en cuanto al comportamiento durante la ejecución) de los distintos elementos del lenguaje. Algunos ejemplos de estas dificultades son confundir las alternativas con repeticiones, algo muy común en entornos de programación por bloques por su similitud sintáctica, confundir definiciones de procedimientos o funciones con su invocación, como creer, por ejemplo, que alcanza con definir un procedimiento para que este se ejecute), y confundir parámetros con argumentos.
4. **Dificultades en la organización del código:** Las dificultades de esta categoría se refieren a las que surgen después de haber construido el código y tienen que ver con entender el comportamiento que el código intenta describir, identificar la porción de código responsable de determinada parte del comportamiento o la modificación del mismo. La estrategia de resolución *top-down* ayuda considerablemente a reducir estas dificultades, aunque suele ser muy difícil que los estudiantes la adopten desde el principio. Por eso es muy importante que los cursos de iniciación en programación refuerzen la adopción de este enfoque de resolución de problemas desde el principio. También es importante promover el uso de nombres declarativos para todos los elementos que deben ser nombrados por el programador: procedimientos, funciones, variables, parámetros e índices.

### 2.3. Pasaje de bloques a texto

Como ya se mencionó, gracias a su sintaxis simplificada, los entornos de programación por bloques reducen las dificultades asociadas a errores sintácticos y permiten que los estudiantes se concentren en los aspectos semánticos del programa que están construyendo [23, 24]. Además, el mecanismo de construcción de código *drag-and-drop* facilita la composición de código de forma intuitiva [25]. Estas ventajas, entre otras, permiten suavizar la curva de aprendizaje de los estudiantes al aprender a programar durante los primeros encuentros, antes de pasar a trabajar con lenguajes más complejos o propósito más general.

A pesar de las ventajas de los entornos de programación por bloques, sobre todo como primer contacto con lenguajes de programación, también existen críticas respecto a los malos hábitos que generan [26, 27]. Algunas propuestas se centraron en generar un enfoque híbrido, que utiliza en una primera etapa un entorno de programación por bloques, pero que migra rápidamente a una segunda etapa en la que se utiliza un lenguaje textual [24, 28, 29, 30]. De esta forma, en un comienzo se pueden enseñar los conceptos y las construcciones programáticas y practicar la semántica de los programas, y posteriormente, teniendo ya algo de práctica en la construcción de programas, pueden enfrentarse con mayor solidez a las dificultades de la sintaxis.

En general, utilizar la estrategia del enfoque mixto reduce considerablemente las dificultades de sintaxis durante la primera etapa y en muchos casos también reduce las de semántica en la segunda (ya que los principales problemas surgen por la nueva sintaxis basada en texto, pero los conocimientos sobre semántica se pueden reutilizar). Sin embargo, también plantea nuevos desafíos. Por un lado, no es claro que la transferencia semántica ocurra tan linealmente como se espera. Hay casos en los que a los estudiantes les cuesta utilizar en la segunda etapa el conocimiento semántico adquirido en la primera. En casos extremos, se siente como si tuvieran que aprender todo de nuevo. Por otro lado, el salto de complejidad de pasar de programar en un entorno visual a uno textual puede ser muy intenso si no se hace con cuidado.

### 2.4. Dimensiones de aprendizaje

Al enseñar a programar se pueden caracterizar dos grandes dimensiones sobre aquello que se aprende: los “Conceptos computacionales” y las “Prácticas computacionales” [3]. La primera se refiere justamente a los conceptos de la disciplina, que incluyen a los elementos del lenguaje así como también otras entidades con las que deben interactuar durante el proceso, como el programa, la computadora o el lenguaje de programación. La segunda se asocia más a aquellas prácticas que realizan en general los programadores, como interpretar código o depurarlo.

### 2.5. Vinculación del trabajo con programación con el mundo real

Es común que la enseñanza en muchas disciplinas se sienta desvinculada del mundo real o que los estudiantes se cuestionen por la utilidad de aquello que se les está enseñando. En los cursos de programación tradicionales no es algo que

aparezca con demasiada frecuencia ya que suelen utilizar un enfoque muy orientado a la producción y el desarrollo. La situación es diferente cuando se utiliza el enfoque conceptual basado en los fundamentos. El hecho de utilizar entornos de programación por bloques, o incluso lenguajes basados en texto, pero que no son de uso industrial, como Gobstones<sup>[31]</sup>, provoca la sensación de que lo que están aprendiendo es “de juguete” y no les va a servir para programar en la “vida real”. El sustento de este enfoque se basa en que lo que se busca transmitir a los estudiantes no es el conocimiento específico del lenguaje usado sino las técnicas de programación que aplican a todos ellos.

### 3. Metodología

Este trabajo se desarrolló en el contexto de un curso corto de programación inicial para estudiantes de escuelas de nivel medio, en el que se introdujeron diversas actividades para fomentar la metacognición. El curso está orientado a personas sin conocimientos previos y tiene como objetivo introducirlos a la programación, con foco en las bases conceptuales del área. El curso se dictó en el marco del Programa Talleres de Ciencia organizado por la Dirección de Orientación Vocacional<sup>2</sup> de la FCEN-UBA.

#### 3.1. Diseño del curso

El curso consistió de seis encuentros de tres horas cada uno, y se realizó por completo en español. Los primeros cinco encuentros se enfocaron en aprender los conceptos básicos de programación, y el último fue dedicado exclusivamente al trabajo grupal en el proyecto final. Durante los primeros cinco encuentros predominó el trabajo de tipo taller, pero intercalando espacios de discusión sobre los conceptos y problematizar lo que se está haciendo. En promedio, cada encuentro consistió en dos horas de práctica, durante las cuales los alumnos trabajaron en grupos de dos o tres personas, 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. El último encuentro se destinó al trabajo en grupo para la elaboración del proyecto final, donde cada grupo eligió un tema que podía elegirse de una lista de propuestas, o, adicionalmente, cada equipo podía optar por trabajar en un proyecto propio, elegido con ayuda del equipo docente.

Para el diseño del curso se siguió el marco teórico-conceptual PRENDER [32], que se utiliza desde experiencias anteriores, el cual integra elementos de la Enseñanza para la Comprensión [33], la Evaluación para el Aprendizaje [34], el Aprendizaje por Indagación [35, 36], el trabajo con problemas auténticos [37], entre otros marcos conceptuales, además de los lineamientos para la enseñanza de las bases conceptuales de la programación en [38].

Una particularidad del curso es que adopta el enfoque heterólogo [39], el cual consiste en ir variando el lenguaje de programación y/o el entorno usado en cada clase.

El uso de este enfoque surgió como modificación con respecto al taller dado en el año 2023, que se dictó por completo utilizando una única herramienta [40]. Ese taller tuvo resultados alejadores, se detectó una falta de confianza por parte de los y las estudiantes que se la atribuyó al hecho de haber utilizado un único lenguaje de programación a lo largo del curso, generando la percepción de que no estaban aprendiendo a programar, sino que estaban aprendiendo a usar una herramienta particular [39]. Por esta razón, se decidió volver al enfoque con múltiples herramientas, esperando que esto generara mayor confianza en los estudiantes, además de la percepción de que no estaban aprendiendo una (o varias) herramienta particular sino que estaban aprendiendo a programar en general.

Además de variar los lenguajes usados, también se utilizaron distintas formas de construcción del código: usando entornos de programación por bloques en los primeros encuentros y lenguajes basados en texto en los últimos.

##### 3.1.1. Contenidos

El primer encuentro consistió en una introducción a la resolución de problemas. Las primeras actividades involucraron discusiones respecto a qué es un problema y qué es programar para luego presentar un problema ficticio e intentar resolverlo buscando una solución para la cual podría ser necesario programar aunque no necesariamente escribir código en una computadora. El problema en cuestión se resolvía utilizando la técnica de búsqueda binaria. Más adelante se realizaron actividades en las que sí se requirió escribir un programa. En un caso, el programa se construyó en el pizarrón para un autómata interpretado por un docente del curso. En el otro, utilizando el entorno de programación por bloques *Robótica en la Escuela*<sup>3</sup>. El cierre de ambas actividades permitió comparar los dos tipos de programas construidos e introducir los conceptos de “programa”, “autómata”, “comando” y “procedimiento”.

En el resto de los encuentros predominó el trabajo en la computadora, resolviendo problemas con diferentes entornos y lenguajes de programación, aunque cada uno comenzó con una actividad sin computadoras, a la que denominamos “anzuelo” [32], en la que se introducen una o más analogías que luego se recuperarán a lo largo de la clase para anclar los conceptos nuevos y ayudar a la comprensión del problema.

<sup>2</sup>DOV Exactas UBA - Sitio web: exactas.uba.ar/dov

<sup>3</sup>Robótica en la Escuela - Sitio web: roboticaenlaescuela.dc.uba.ar

En el segundo encuentro se trabajó con *Arduino en la Escuela*<sup>4</sup> [41] y se presentaron los conceptos “sensor”, “actuador”, “expresión” y “alternativa condicional”. En el tercer encuentro se trabajó con *AppInventor* [42] y se presentaron los conceptos “evento” y “parámetro”. En el cuarto encuentro se trabajó con *Sonic Pi* y se presentaron los conceptos “repetición” y “sincronización”. En el quinto encuentro se trabajó con *Python* y se presentaron los conceptos “variable”, “función” y “lista”.

### 3.1.2. Actividades metacognición en las clases de programación

La metacognición en el curso tuvo un rol fundamental en diferentes instancias del mismo. Al finalizar cada encuentro se propuso una actividad metacognitiva a través de algunas preguntas para reflexionar, presentadas en un formulario online que se completó en los últimos minutos de cada clase. Dichas actividades se organizaron desde menor demanda a mayor demanda metacognitiva [12]. Involucraron, entre otras, la comparación explícita sobre lo que se pensaba antes de comenzar el curso, una reflexión sobre los aprendizajes y las dificultades para alcanzarlos, la reflexión explícita sobre la conexión entre las nuevas ideas y las ideas previas. Algunas de estas actividades pueden entenderse como rutinas de pensamiento en el sentido que se propone en [13].

## 3.2. Encuestas

Además de las encuestas breves de metacognición de cada encuentro, se realizaron otras dos encuestas de mayor extensión; una al principio del primer encuentro (encuesta inicial) y la otra al final del último encuentro (encuesta final). La encuesta inicial, que sirvió de pre-test, tenía como objetivo rescatar el conocimiento metacognitivo de los participantes del curso respecto de la programación, de los entornos de programación, de los aprendizajes previos al curso, los sentimientos de dificultad del aprendizaje de determinados elementos de programación, y las expectativas del curso. La encuesta final, que se usó como post-test, tenía como objetivo, también, rescatar el conocimiento metacognitivo respecto de la programación, de la autopercepción sobre la comprensión de determinados elementos de programación, y de evaluación del taller en relación a las expectativas previas. Ambos test contaron con preguntas que tuvieron en cuenta los tests JoL, FoK y EoL mencionados en la Sección 1. Varias preguntas de la encuesta inicial se repitieron en la encuesta final para medir el impacto del curso.

## 3.3. Población

Las personas que asistieron al taller fueron estudiantes de entre 15 y 18 años de escuelas medias del área metropolitana de Buenos Aires (AMBA). En total participaron 38 personas de los cuales 34 asistieron a al menos el 75 % de los encuentros. Los cuatro participantes que no alcanzaron el 75 % de asistencia fueron descartados de los análisis siguientes.

Sólo el 18 % cursa sus estudios en alguna escuela técnica con orientación en computación o informática. Aún así, el 96 % aseguró haber tenido formación en contenidos relacionados con programación (e.g., computación, informática y TICs) en alguna materia.

Nota: Los porcentajes están calculados con base en 27 de los 34 participantes ya que son quienes completaron la encuesta inicial al principio del curso.

## 3.4. Métodos de análisis

Dada la complejidad propia del análisis de los procesos de enseñanza y aprendizaje, es pertinente abordarla desde un enfoque mixto. Los análisis de tipo cuantitativo y cualitativo, pueden complementarse de manera de ofrecer un panorama más complejo de lo sucedido durante la implementación de la propuesta [43, 44]. La finalidad no será medir aprendizajes, sino identificar procesos que llevan adelante los estudiantes al aprender programación y analizar qué permite esta propuesta que incluye actividades metacognitivas. En relación al análisis cuantitativo, se decidió implementar elementos de la estadística descriptiva, que permitieran describir algunas de las variables de las encuestas iniciales y finales. Se utilizó una distribución de frecuencias que permitió presentar y analizar los datos [44]. En relación al análisis cualitativo, se realizó un análisis temático [45]. Este tipo de análisis permite la organización sistemática del conjunto de datos, identificando patrones de significados. A partir de ellos se construyen categorías que estuvieron guiadas por la pregunta ¿qué reflexiones metacognitivas emergen durante el curso? La construcción involucró la generación inicial de códigos para clasificar los datos; luego, una clusterización de dichos códigos en categorías que reflejaban patrones significativos relacionados con la investigación; finalmente, se revisaron y validaron las categorías mediante la triangulación entre investigadores [45].

# 4. Resultados y discusión

## 4.1. Análisis cuantitativo de las encuestas inicial y final

Al preguntarles si sabían programar, el 22 % respondió que sí, el 22 % que no y el 56 % restante que no podía asegurar si sabía o no. Independientemente de la respuesta a la pregunta anterior, 70 % dijo haber aprendido a programar en la escuela,

<sup>4</sup>Arduino en la Escuela - Sitio web: aeletdcuba.ar

30 % mediante cursos y 22 % de forma autodidacta (la pregunta permitía seleccionar varias respuestas).

Al preguntarles por los lenguajes de programación conocidos, el 89 % mencionó alguno, ya sea de bloques o textual. 74 % mencionó al menos uno de bloques (siendo Scratch el más mencionado con 63 %) y 74 % mencionó al menos uno textual (siendo Python el más mencionado con 59 %). Aunque la mayoría mencionó tanto lenguajes textuales como de bloques, un 15 % sólo mencionó lenguajes textuales y otro 15 % sólo de bloques.

Nota: Los porcentajes están calculados con base en 27 de los 34 participantes ya que son quienes completaron la encuesta inicial al principio del curso.

En la encuesta inicial se les pidió calificar su conocimiento en programación con un número en una escala del 1 al 5. El gráfico de la izquierda de la Figura 1 muestra la frecuencia de respuestas obtenidas para cada valor de la escala. El promedio fue 2,2 (desv. est. 0,8). Esto daría cuenta que los participantes del curso creen que no poseen tantos conocimientos sobre programación. En la encuesta final se les pidió que volvieran a calificar su conocimiento en programación con la misma escala. El gráfico de la derecha de la Figura 1 muestra la frecuencia de respuestas obtenidas para cada valor de la escala. El promedio fue 3,3 (desv. est. 0,9). En función de estos gráficos podemos decir que aumentó su autoperccepción respecto del conocimiento en programación, aunque se hubiese esperado un número más alto. Esto abre preguntas en torno a cómo aumentar la autoconfianza de los estudiantes en esta área.

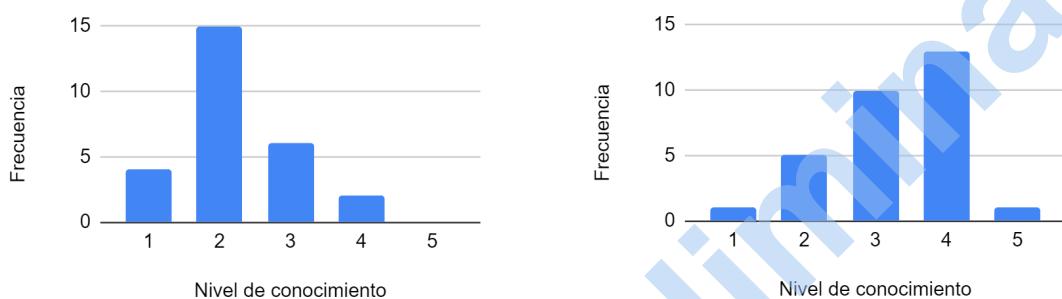


Figura 1: Frecuencia de respuestas a cada valor de la escala en la pregunta “¿Cuánto crees que sabes programar?” de la encuesta inicial (izquierda) y en la pregunta “¿Cuánto crees que sabés programar AHORA que ya terminó el curso?” de la encuesta final (derecha).

## 4.2. Análisis cualitativo de las actividades de MC

A continuación, se presentan las categorías correspondientes al análisis de las reflexiones metacognitivas que se realizaron a lo largo del curso.

### 4.2.1. Cambio de pensamiento de escribir código a resolver problemas

A partir de las respuestas recibidas se aprecia, en líneas generales, que los estudiantes reflexionan acerca de que antes del curso pensaban que la programación era hacer programas o escribir código, y que luego del curso piensan a la programación como un modo de resolver problemas. En las clases 1 y 5 se solicitó a los estudiantes responder: “En función de lo que trabajamos hoy, completá la siguiente frase *Antes pensaba que la programación era ..., mientras que ahora pienso que la programación es...*”. Este tipo de actividad metacognitiva, retomada de [13], permite en este contexto reflexionar sobre las concepciones alternativas y distinguirlas de lo aprendido. En particular, la actividad metacognitiva permitió la emergencia de una concepción alternativa extendida en esta población, respecto de que la programación es escribir código. Algunos ejemplos que muestran esto ya aparecen incluso en el primer encuentro:

“*Antes pensaba que la programación era escribirle que querías que hiciera la computadora, mientras que ahora pienso que la programación es, más generalmente, una forma de solucionar diversos problemas computacionales, dividiéndose en más pequeños.*”

(Julián)

“*Antes pensaba q la programación era el desarrollo de servicios digitales como aplicaciones o páginas webs, mientras que ahora pienso que la programación es la implementación de soluciones digitales ante los distintos problemas que se pueden presentar.*”

(Héctor)

“*Antes pensaba que era solo un código y ahora pienso que sirve para solucionar problemas tecnológicos de la vida cotidiana.*”

(Ciro)

Luego, al ir atravesando el curso, los estudiantes reflexionan sobre la potencialidad de la escritura del código para solucionar problemas. Esta idea se asemeja mucho más a lo que se espera en la enseñanza de la programación.

Aparecieron respuestas similares a la misma pregunta en el quinto encuentro:

*“Escribir un código, ahora creo que es la ciencia que consigue automatizar procesos para solucionar un problema específico.”*

(Diego)

*“Antes pensaba que la programación era hacer códigos para un programa, mientras que ahora pienso que la programación es la solución de problemas a base de códigos.”*

(Fernanda)

*“Antes pensaba que la programación era simplemente escribir código, mientras que ahora pienso que la programación es pensar fuera de la computadora.”*

(Luis)

Adicionalmente, en el quinto encuentro se les preguntó explícitamente “¿Te parece que programar es lo mismo que escribir código?” Entre las respuestas a esa pregunta encontramos:

*“No, programar es diferente de escribir código porque programar implica un cierto nivel de pensamiento sobre los algoritmos que se van a utilizar en el sistema y escribir código es solo la acción de implementar estos.”*

(Julian)

*“No, ya que programar busca la solución a un problema y el código no.”*

(Fernanda)

*“Yo creo que existen distintas maneras de programar aparte de los códigos.”*

(Pedro)

*“No, escribir código es una parte de programar. Programar es pensar la solución a un problema y dar las órdenes, mediante el código, para lograr que la computadora haga lo que queremos.”*

(Elena)

#### 4.2.2. Percepciones sobre las dificultades al aprender a programar

Los estudiantes reflexionan respecto de la dificultad que involucra la programación, sobre todo antes de comenzar el curso. Al igual que en el caso anterior, estas reflexiones emergen a partir de la actividad metacognitiva de las clases 1 y 5.

*“Antes pensaba que la programación era difícil y tenías que estudiar mucho para saber hacerlo, mientras que ahora pienso que la programación es mucho más simple y que con saber algunas cosas podes ir complementándolo.”*

(Manuela)

*“Antes pensaba que la programación era muy complicado y que se necesitaba mucho conocimiento, y hoy me di cuenta de que es algo mucho más simple.”*

(Indio)

*“Antes pensaba que la programación era complicada y ahora pienso que es más fácil, pero hay que saber bien todo.”*

(Ciro)

Es relevante destacar que este tipo de actividades metacognitivas habilita a los estudiantes a reflexionar y explicitar las propias dificultades, lo cual da pistas a los docentes para encarar los pasos siguientes en la enseñanza. Durante las clases 2 a 5, se solicitó a los estudiantes que explicitaran “¿Qué dificultades encontraste durante el encuentro y cómo las solucionaste?”.

En lo relativo a la organización del código, algo que pocas veces se tiene en cuenta al enseñar a programar es dar lineamientos para estructurar adecuadamente el código haciendo división en subtareas, además de nombrar adecuadamente los elementos que se definen, para que el código sea fácil de entender y de corregir. En este curso, estas ideas fueron centrales y es notable que hayan surgido dificultades asociadas a ellas y que fueron solucionadas siguiendo los lineamientos mencionados:

*“Las dificultades que encontré fue no saber cómo facilitar un proceso, pero luego descubrí la solución que era una estructura.”*

(Esteban)

*“Era difícil modificar el código en las distintas consignas por lo desordenado que estaba. Para solucionarlo utilice comandos definidos.”*

(Bárbara)

Esto también fue destacado al responder la pregunta “¿Qué nuevas ideas sobre programación aprendiste hoy?”. Una de las respuestas rescata el hecho de haber aprendido a definir procedimientos:

*“Aprendí la utilidad de convertir comandos primitivos en comandos definidos para facilitar el entendimiento del programa.”*

(Bárbara)

En los cuestionarios de los encuentros 2 y 3, en los que se trabajó con los entonces de programación por bloques Arduino en la Escuela y AppInventor, respectivamente, la mayoría de las dificultades que surgieron estuvieron asociadas a la semántica de los elementos del lenguaje y a la estrategia para la resolución del problema. Algunos ejemplos encontramos son los siguientes:

*“Hoy tuvimos dificultades con las condicionales simples, pero las solucionamos con un Si no.”*

(Patricio)

*“Lo que vi difícil es poner los tiempos para poner letras y luego hacer pausas para poner otra letra.”*

(Pedro)

*“Se nos complicó el mover el teléfono hacia un lado y que la bola se mueva hacia ese mismo lado. Sin embargo pudimos solucionarlo realizando preguntas y probando en la compu distintos valores.”*

(Ringo)

*“Las dificultad que encontré fue que al momento de intentar de hacer funcionar la bola de bolos, ya que tenía que hacer que rote al sitio donde roto el celular; y lo solucione ya que un profesor me ayudó con pistas a como hacer que funcione.”*

(Pedro)

Sin embargo, se detectaron también algunas dificultades relacionadas a la sintaxis del lenguaje, aunque en proporción, fueron mucho menos. Algunos ejemplos son:

*“La función de los bloques, lo resolví experimentando.”*

(Octavio)

*“Encontré dificultades a la hora de programar con bloques, y las solucioné creando comandos.”*

(Ramiro)

En el cuarto encuentro se empezó a trabajar con Sonic Pi, que es un lenguaje basado en texto. Aunque se esperaba que la proporción de dificultades asociadas a la sintaxis aumentara, sólo se recibió un único comentario al respecto:

*“Durante el encuentro me encontré con problemas principalmente de sintaxis, pero por fortuna Sonic Pi te marca bastante los mismos.”*

(Luis)

El resto de las dificultades encontradas estuvieron asociadas a los conceptos o a la estrategia:

*“Quizá me costó un poco la parte de las notas o cómo combinarlas para que quede un sonido decente.”*

(Héctor)

*“La única dificultad que encontré no era programativa (sic) sino más bien musical, ya que me costaba encontrar los tiempos.”*

(Umma)

*“Cometí algunos errores al saltarme partes de códigos necesarios.”*

(Bernardo)

Por otro lado, en el quinto encuentro, en el que se trabajó con Python, aumentó considerablemente la proporción de dificultades asociadas a la sintaxis:

*“La tabulación, resolví tabulando.”*

(Andrés)

*“No sabía que la estructura de las líneas era tan necesaria.”*

(Agustín)

#### 4.2.3. Sobre el pasaje de bloques a texto

En general, se plantea un problema cuando en los cursos se viene trabajando con herramientas por bloques y se desea hacer la transición a trabajar con lenguajes basados en texto.

Al diseñar el curso se tuvo todo esto en cuenta y es por ello que se decidió que en los primeros se trabaje exclusivamente con bloques y luego, una vez que los estudiantes tuvieran suficiente práctica creando programas, hacer el pasaje a texto. Se utilizó el lenguaje de la herramienta Sonic Pi como etapa intermedia antes de llegar a Python, en tanto el primero es un lenguaje basado en texto, pero, a efectos del curso, donde se lo utilizó de manera muy gradual y simple, no implicó un cambio muy grande para los estudiantes, dado que aunque debía tipar el código, el conjunto de instrucciones usado fue muy limitado, así como la sintaxis del lenguaje que tuvieron que aprender.

Se destaca que las mayores dificultades asociadas a la sintaxis del lenguaje surgieron en el quinto encuentro, al empezar a trabajar con Python y no en el cuarto que fue el primer contacto que tuvieron con un lenguaje de programación textual. De hecho, hubo un comentario que mencionó explícitamente que haber usado bloques le ayudó a entender mejor el lenguaje textual, como respuesta a la pregunta “Hoy comenzamos a programar en texto ¿Qué cosas de lo que ya sabías te ayudaron a entender esta nueva forma de escribir programas?”:

*“Me ayudó utilizar bloques para entender mejor la estructura de un programa y el uso de comandos.”*  
(Julián)

Otras respuestas hicieron mención de cosas que ya sabían de antes, aunque sin hacer referencia explícita a que es algo que aprendieron al programar con bloques:

*“La forma de agrupar y organizar acciones y repetirlas.”*  
(Karen)

*“Me ayudó a entender el tema de que cuando estoy en un loop necesito indentar.”*  
(Indio)

Es posible, sin embargo, que no se estén refiriendo a lo que aprendieron en los encuentros anteriores. Como veremos en la sección 4.2.5, muchos parecen haber entendido en esta pregunta que nos referíamos a cosas que sabían antes de comenzar el curso.

Aparentemente, el pasaje de bloques a texto no fue tan abrupto como se esperaba. Sin embargo, sí fue un cambio grande pasar de Sonic Pi a Python. Esta observación tiene dos implicancias importantes: por un lado, Sonic Pi resultó ser un buen lenguaje intermedio para la transición de bloques a texto, en tanto no le resultó complicado a los estudiantes, mientras que, por otro, la dificultad en el pasaje de bloques a texto no viene dada exclusivamente por la forma en que se construye el código.

Otra explicación posible es que la forma en la que se utilizó Sonic Pi está mucho más cerca de los entornos de programación por bloques que de los lenguajes textuales de uso general, debido a que sólo utilizamos algunos elementos del lenguaje. Durante el desarrollo del cuarto encuentro se observaron muchos errores sintácticos en los primeros intentos por escribir programas en Sonic Pi, principalmente asociados al uso de mayúsculas en lugar de minúsculas o el manejo de espacios (ya sea poniéndolos donde no van, u omitiéndolos donde son necesarios), o por no cerrar los bloques de los ciclos. Estos errores se resolvieron casi inmediatamente con la ayuda de los docentes a cargo y ni siquiera fueron mencionados como dificultades en el cuestionario.

#### 4.2.4. Dimensiones de aprendizajes sobre programación

Los estudiantes reflexionan sobre los aprendizajes alcanzados en las clases (de 2 a 5) ante la pregunta metacognitiva “¿Qué nuevas ideas sobre programación aprendiste hoy?”.

*“Aprendí una idea que no sabía que estaba en la programación como los sensores como por ejemplo de proximidad o cuando no hay luz.”*  
(Pedro)

*“Aprendí sobre comandos, primitivos y no primitivos, de sensores y de actuadores.”*  
(Ramiro)

*“Aprendí sobre sensores y condicionales de estos, sobre sistemas de coordenadas y movimiento en estos.”*  
(Julián)

*“Hoy aprendí diferentes formas de solucionar problemas como simplificar códigos largos.”*  
(Manuela)

*“Hoy aprendí que podemos crear y modificar un programa simple con diferente accionadores para los celus.”*  
(Tobías)

Según la caracterización de las dimensiones de aprendizaje planteadas en 2.4, podemos observar que Lucia, Pablo y Nicolás explicitan haber aprendido “Conceptos computacionales”, en tanto mencionan conocimientos sintácticos o esquemáticos. Por otro lado, Gabriel y Esteban explicitan haber aprendido “Prácticas computacionales”, ya que en sus reflexiones emergen ciertas habilidades o conocimiento estratégico sobre la resolución de problemas (simplificar códigos, convertir comandos o crear y modificar programas). Este tipo de actividades metacognitivas permiten al docente identificar estas distinciones entre lo aprendido.

En el cuestionario del quinto encuentro en particular, se notó una gran parte de respuestas que se limitaban a referirse al lenguaje que aprendieron:

*“Aprendí sobre cómo empezar a usar python.”*  
(Daniel)

*“Cómo programar en python.”*  
(Ciro)

*“Nada muy nuevo, ya que había trabajado con python anteriormente.”*  
(Umma)

Sin embargo, también se encontraron respuestas que ponían el foco en los conceptos aprendidos y no en los detalles del lenguaje usado:

*“Principalmente, la diferencia entre procedimientos y funciones.”*  
(Sebastián)

*“Aprendimos sobre algoritmos de sorteo y su implementación.”*  
(Julián)

#### **4.2.5. Impacto de los conocimientos previos en el curso**

En el cuestionario del cuarto encuentro se buscó trabajar con el conocimiento metacognitivo declarativo, ya que en esa clase habían usado un lenguaje basado en textos por primera vez. Así, se les preguntó “Hoy comenzamos a programar en texto ¿Qué cosas de lo que ya sabías te ayudaron a entender esta nueva forma de escribir programas?”.

Varios aseguraron que les ayudó haber trabajado antes con otros lenguajes textuales y cómo esto les impactó positivamente en el trabajo de la clase:

*“Ya sabía python lo cual me ayudó a entender la mayoría del programa.”*

(Umma)

*“Yo sabía la estructura de código de python y me sirvió para entender el loop del código.”*

(Ciro)

*“Me ayudó haber programado en código en el lenguaje C++ aunque la sintaxis es distinta.”*

(Elena)

*“Yo ya sabía un poco la estructura de Python, la cual se repite en varios lenguajes incluido el de Sonic Pi.”*

(Luis)

Por otro lado, se puede hacer un contraste con un comentario que apareció en la respuesta a la pregunta “¿Qué dificultades encontraste durante el encuentro y cómo las solucionaste?” del segundo encuentro, ya que aquí una estudiante afirmó que saber programación en texto le complicó el trabajo con bloques:

*“Al saber algo de programación en código, directamente, se me complicó pasar esa idea que tenía al formato de módulos. Al final con ayuda lo logré, pero el tener ese conocimiento previo me complicó las cosas al final.”*

(Elena)

#### **4.2.6. Comprensión sobre el mundo que les rodea**

Entre las respuestas a la pregunta “¿Qué nuevas ideas sobre la programación y los celulares te surgieron a partir de lo trabajado hoy?” del tercer encuentro, se encontraron algunos comentarios relacionados a que a través del curso pudieron tener una mayor comprensión de algunas herramientas tecnológicas o desarrollos conocidos:

*“Me dio una idea de como funcionan muchas redes sociales que utilizo al igual que algunos videojuegos de celular.”*

(Fernanda)

*“Con la clase de hoy, se me vinieron ideas de cómo las personas crearon otras apps como videojuegos, redes sociales, etc.”*

(Pedro)

Esto marca algo que podría ser un indicio de transferencia. Sin embargo, no explicitan qué vínculos encuentran entre un campo y el otro.

Por otro lado, también se mostró que el trabajo con el curso les amplió su visión sobre la tarea de programar y qué implica. Esto lo observamos en una respuesta a la pregunta “En función de lo que trabajamos hoy, completá la siguiente frase *Antes pensaba que la programación era ..., mientras que ahora pienso que la programación es...*” del quinto encuentro:

*“Antes pensaba que la programación era solo un código y ahora pienso que hay muchas maneras de verlo (codigo, edición de sonido, robótica, etc).”*

(Nicolás)

También lo encontramos en una respuesta a la pregunta “¿Qué nuevas ideas sobre programación aprendiste hoy?” del cuarto encuentro:

*“que se puede programar muchas más cosas de lo que pensaba.”*

(Manuela)

### **5. Conclusiones**

Algunas investigaciones se enfocan en las funciones ejecutivas, pero sin prestar atención a la metacognición. Así, suponen que por el hecho de trabajar el Pensamiento computacional, los estudiantes desarrollan mejores habilidades ejecutivas [46, 47]. Esto es un problema porque se asume que estas habilidades se desarrollan espontáneamente sin cultivarlas conscientemente. Esto deja por fuera a estudiantes que no logran el desarrollo de estas habilidades, las cuales se ven necesarias para aprender más y mejor programación. Hacerlo de manera implícita no da la oportunidad de corregirse, retroalimentarse y revisarse. En esta propuesta, mostramos cómo pueden comenzar a desarrollarse estas habilidades de manera explícita y andamiajada por el docente.

La propuesta permitió que los estudiantes realizaran reflexiones metacognitivas explícitas tendientes a hacerse conscientes sobre sus propias concepciones de programación, así como de aquellos conocimientos o habilidades que ya traían,

además de hacerse conscientes sobre las dificultades del proceso de aprendizaje. El identificar dificultades permite buscar soluciones específicas a las mismas y poder avanzar en el proceso, además de hacerse conscientes sobre aquello que aprendieron y lo que falta aprender.

Finalmente, dejamos algunas sugerencias para utilizar la Metacognición en las clases de programación. En primer lugar, hacer visible el pensamiento [13]. Esto es, ofrecer instancias donde “dejemos de programar” por un momento y analicemos qué hicimos, cómo lo hicimos, qué procesos pusimos en juego. También destinar momentos de la clase para hacer metacognición. Existen diferentes actividades posibles que permiten andamiar el proceso metacognitivo [12] identificando alcances y limitaciones de los estudiantes a partir de sus propias reflexiones. Por ejemplo, la propuesta de revisar las ideas (como la actividad de “Antes pensaba... ahora pienso...”) que permite identificar el proceso de aprendizaje de los estudiantes.

## Referencias

- [1] R. Swartz, R. Reagan, A. Costa, B. Beyer y B. Kallick. *El aprendizaje basado en el pensamiento*. Madrid: Ediciones SM España, 2014.
- [2] Ü. Çakiroğlu y B. Er. «Effect of using metacognitive strategies to enhance programming performances». En: *Informatics in Education* 19.2 (2014), págs. 181-200.
- [3] S. Y. Lye y J. H. L. Koh. «Review on teaching and learning of computational thinking through programming: What is next for K-12?». En: *Computers in Human Behavior* 41 (2014), págs. 51-61. DOI: 10.1016/j.chb.2014.09.012. URL: <https://doi.org/10.1016/j.chb.2014.09.012>.
- [4] R. Scherer, F. Siddiq y B. Sánchez Viveros. «A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions». En: *Computers in Human Behavior* 109 (2020), pág. 106349. ISSN: 0747-5632. DOI: 10.1016/j.chb.2020.106349. URL: <https://www.sciencedirect.com/science/article/pii/S0747563220301023>.
- [5] Y. P. Huang, H. Kim, Y. Pan, X. L. Zheng e Y. F. Tu. «Promoting elementary school students' programming learning: Effects of metacognitive vs. cognitive scaffolding». En: *Journal of Research on Technology in Education* (2024), págs. 1-16. URL: <https://doi.org/10.1080/15391523.2024.2338086>.
- [6] W. Li, C. Y. Liu y J. C. R. Tseng. «Effects of the interaction between metacognition teaching and students' learning achievement on students' computational thinking, critical thinking, and metacognition in collaborative programming learning». En: *Education and Information Technologies* 28 (2023), págs. 12919-12943. URL: <https://doi.org/10.1007/s10639-023-11671-2>.
- [7] Y. Shin, J. Jung, J. Zumbach y E. Yi. «The Effects of Worked-Out Example and Metacognitive Scaffolding on Problem-Solving Programming». En: *Journal of Educational Computing Research* 61.6 (2023), págs. 1312-1331. URL: <https://doi.org/10.1177/07356331231174454>.
- [8] Y. Zhou, C. S. Chai, X. Li, C. Ma, B. Li, D. Yu y J. C. Liang. «Application of metacognitive planning scaffolding for the cultivation of computational thinking». En: *Journal of Educational Computing Research* 61.6 (2023), págs. 1123-1142. DOI: 10.1177/07356331231160294.
- [9] G. Pérez y L. González Galli. «Metacognition and Self-Regulation in Science Learning». En: *Rethinking Science Education in Latin-America*. Springer, 2024, págs. 291-308.
- [10] C. T. W. da Rosa y L. M. Darroz. *Cognição, linguagem e docência: aportes teóricos*. Brasil: Ilustração, 2022.
- [11] C.-Y. Wang, B.-L. Gao y S.-J. Chen. «The effects of metacognitive scaffolding of project-based learning environments on students' metacognitive ability and computational thinking». En: *Education and Information Technologies* 29.5 (2024), págs. 5485-5508. URL: <https://doi.org/10.1007/s10639-023-12022-x>.
- [12] G. Pérez y L. González Galli. «Actividades para fomentar la metacognición en las clases de biología». En: *Tecné, Episteme y Didaxis* 47 (2020), págs. 235-249. URL: <https://doi.org/10.17227/ted.num47-7970>.
- [13] R. Ritchhart, M. Church y K. Morrison. *Hacer visible el pensamiento*. Ciudad Autónoma de Buenos Aires: Paidós, 2014.
- [14] J. Pozo. *Psicología del Aprendizaje Humano: Adquisición de conocimiento y cambio personal*. Madrid: Morata, 2014.
- [15] A. Ben-David y N. Orion. «Teachers' voices on integrating metacognition into science education». En: *International Journal of Science Education* 35.18 (2013), págs. 3161-3193.
- [16] T. O. Nelson. «Metamemory: A theoretical framework and new findings». En: *Psychology of learning and motivation* (1990), págs. 125-173.
- [17] M. G. Rhodes. «Judgments of learning: Methods, data, and theory». En: *The Oxford handbook of metamemory* 1 (2016), págs. 65-80.
- [18] S. Torres Jiménez, J. J. Ramírez-Echeverry y F. Restrepo-Calle. «The Development and Validation of the Questionnaire to Characterize Learning Strategies in Computer Programming (CEAPC)». En: *Journal of Educational Computing Research* 61.8 (2024), págs. 103-138.

- [19] P. Pintrich, C. Wolters y G. Baxter. «Assessing Metacognition and Self-Regulated Learning». En: *Issues in the Measurement of Metacognition*. Lincoln, Nebraska: Buros Institute of Mental Measurements, 2000, págs. 43-97.
- [20] L. Silva, A. Mendes, A. Gomes y G. Fortes. «What Learning Strategies are Used by Programming Students? A Qualitative Study Grounded on the Self-regulation of Learning Theory». En: *ACM Transactions on Computing Education* 24.1 (2024), págs. 1-26. URL: <https://doi.org/10.1145/3635720>.
- [21] D. Loksa, L. E. Margulieux, B. A. Becker, M. Craig, P. Denny, R. S. Pettit y J. Prather. «Metacognition and self-regulation in programming education: Theories and exemplars of use». En: *ACM Transactions on Computing Education* 22.4 (2022), págs. 1-31. URL: <https://doi.org/10.1145/3487050>.
- [22] S. Krusche y J. Berrezueta-Guzman. «Introduction to Programming using Interactive Learning». En: *2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEE&T)* (2023), págs. 178-182. DOI: 10.1109/CSEET58097.2023.00037.
- [23] J. Maloney, M. Resnick, N. Rusk, B. Silverman y E. Eastmond. «The scratch programming language and environment». En: *ACM Transactions on Computing Education* 10.4 (2010), págs. 1-15. DOI: 10.1145/1868358.1868363.
- [24] D. J. Malan y H. H. Leitner. «Scratch for budding computer scientists». En: *Proceedings of the 38th ACM Technical Symposium on Computer Science Education* (2007), págs. 223-227.
- [25] D. Weintrop. «Minding the gap between blocks-based and text-based programming: Evaluating introductory programming tools». En: *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (2015).
- [26] O. Meerbaum-Salant, M. Armoni y M. Ben-Ari. «Habits of programming in Scratch». En: *Proceedings of the 16th annual joint Conference on Innovation and technology in computer science education* (2011), págs. 168-172.
- [27] D. Weintrop y U. Wilensky. «To Block or not to Block, That is the Question: Students' Perceptions of Blocks-based Programming». En: *Proceedings of the 14th International Conference on Interaction Design and Children* (2015), págs. 199-208. DOI: 10.1145/2771839.2771860. URL: <https://dl.acm.org/doi/10.1145/2771839.2771860>.
- [28] W. Dann, D. Cosgrove, D. Slater, D. Culyba y S. Cooper. «Mediated transfer: Alice 3 to Java». En: *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (2012), págs. 141-146.
- [29] M. Armoni, O. Meerbaum-Salant y M. Ben-Ari. «From Scratch to "Real" Programming». En: *ACM Transactions on Computing Education* 14.4 (2015), págs. 1-15. DOI: 10.1145/2677087.
- [30] C. M. Lewis. «How programming environment shapes perception, learning and goals: Logo vs. Scratch». En: *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (2010), págs. 346-350.
- [31] P. E. Martínez López, D. Ciolek, G. Arévalo y D. Pari. «The Gobstones method for teaching computer programming». En: *XXV Simposio de Educación Superior en Computación (SIESC'17), dentro de la XLIII Conferencia Latinoamericana de Informática (CLEI'17)* (2017), págs. 1-9.
- [32] C. Cossio-Mercado y G. P. Fernández. «PRENDER: Una propuesta didáctico-pedagógica para la enseñanza de las Ciencias de la Computación». En: *Jornadas Argentinas de Didáctica de las Ciencias de la Computación 2023* (2023). URL: [https://www.researchgate.net/publication/380173508\\_PRENDER\\_Una\\_propuesta\\_didactico-pedagogica\\_para\\_la\\_enseñanza\\_de\\_las\\_Ciencias\\_de\\_la\\_Computacion](https://www.researchgate.net/publication/380173508_PRENDER_Una_propuesta_didactico-pedagogica_para_la_enseñanza_de_las_Ciencias_de_la_Computacion).
- [33] M. Stone Wiske. *La enseñanza para la comprensión*. Buenos Aires, Argentina: Paidós, 1999.
- [34] R. Anijovich y G. Cappelletti. *La evaluación como oportunidad*. Buenos Aires, Argentina: Paidós, 2017.
- [35] J. J. Schwab. «What Do Scientists Do?» En: *Behavioral Science* 5.1 (1960), págs. 1-27.
- [36] J. J. Schwab. «The teaching of science as enquiry». En: *The teaching of science*. Ed. por J. J. Schwab y P. F. Brandwein. Cambridge, Harvard University Press, 1962, págs. 1-103.
- [37] J. I. Pozo e Y. Postigo. «La solución de problemas como contenido procedimental en la educación obligatoria». En: *La solución de problemas*. Ed. por J. I. Pozo, M. del Puy Pérez, J. Domínguez, M. Á. Gómez e Y. Postigo. Editorial Santillana, 1994.
- [38] P. E. Martínez López. *Las bases conceptuales de la programación. Una nueva forma de aprender a programar*. EBook (1era ed.). La Plata, el autor, 2013. ISBN: 978-987-33-4081-9.
- [39] G. P. Fernández y C. Cossio-Mercado. «¿Es posible enseñar los fundamentos de la programación sólo con Arduino?: Análisis de un curso introductorio para estudiantes de secundaria usando bloques». Enviado a *Jornadas Argentinas de Didáctica de las Ciencias de la Computación 2024*. 2024.
- [40] G. P. Fernández y C. Cossio-Mercado. «Enseñanza de fundamentos conceptuales de programación usando Arduino». En: *Jornadas Argentinas de Didáctica de las Ciencias de la Computación 2023* (2023). URL: [https://www.researchgate.net/publication/380174232\\_Esenanza\\_de\\_fundamentos\\_conceptuales\\_de\\_programacion\\_usando\\_Arduino](https://www.researchgate.net/publication/380174232_Esenanza_de_fundamentos_conceptuales_de_programacion_usando_Arduino).
- [41] G. P. Fernández y C. Cossio-Mercado. «AeIE: a versatile tool for teaching programming and robotics using Arduino». En: *50a Conferencia Latinoamericana de Informática (CLEI)* (2024).

- [42] D. Wolber, H. Abelson, E. Spertus y L. Looney. *App Inventor: Create Your Own Android Apps*. O'Reilly Media, Inc., 2011.
- [43] J. Creswell. *Educational research. Planning, conducting, and evaluating quantitative and qualitative research*. Boston: Pearson, 2012.
- [44] R. Hernández Sampieri, C. Fernández Collado y P. Baptista Luciano. *Metodología de la Investigación (6ta edición)*. México: Mc Graw Hill, 2014.
- [45] V. Braun y V. Clarke. *Successful qualitative research: A practical guide for beginners*. Los Ángeles: Sage, 2013.
- [46] C. R. Castro, L. H. Rodríguez Rodríguez y L. F. Castillo. «La enseñanza del pensamiento computacional y sus efectos sobre las funciones ejecutivas de los niños». En: *Revista EIA* 21.42 (2024), págs. 1-21. DOI: 10.24050/reia.v21i42.1740.
- [47] C. Montuori, F. Gambarota, G. Altoé y B. Arfé. «The cognitive effects of computational thinking: A systematic review and meta-analytic study». En: *Computers & Education* 210 (2024), pág. 104961. ISSN: 0360-1315. DOI: 10.1016/j.compedu.2023.104961. URL: <https://www.sciencedirect.com/science/article/pii/S0360131523002385>.

Versión Preliminar

# Implementación de Microlearning utilizando Susurros: Una Propuesta en el ámbito Universitario.

Claudia Alderete<sup>1,2</sup>, Pablo Vera<sup>1,2</sup>

<sup>1</sup>Universidad Abierta Interamericana, Facultad de Tecnología Informática. Centro de Altos Estudios en Tecnología Informática. Buenos Aires, Argentina

<sup>2</sup>Universidad Nacional de La Matanza, Departamento de Ingeniería e Investigaciones Tecnológicas. Buenos Aires, Argentina.

[claudiagabriela.alderete@uai.edu.ar](mailto:claudiagabriela.alderete@uai.edu.ar) , [pablomartin.vera@uai.edu.ar](mailto:pablomartin.vera@uai.edu.ar)

## Resumen

Este artículo explora la implementación de breves consejos o tips, presentados como “susurros”, en Historias y Reels de Instagram dentro de un contexto educativo. Es importante destacar el uso comunicacional de esta red social y su potencial como entorno virtual, como es el caso de Instagram. El Microlearning se utiliza para fortalecer habilidades específicas relacionadas con los contenidos de una materia en la Universidad. Se presentan varios ejemplos que ilustran cómo estos pequeños fragmentos, a modo de cápsulas de contenido, se van acumulando para formar un contenido más complejo e integrado, similar a bloques que se ensamblan para construir una estructura mayor y más sólida. Finalmente, se exponen los resultados de su implementación.

**Palabras claves:** Microlearning; Instagram; Susurros; Historias y Reels.

## 1. Introducción

Cada vez más las personas y en particular los jóvenes utilizan sus dispositivos móviles para sus actividades cotidianas, esa persistencia en la conectividad, brindan oportunidades para el aprendizaje en un escenario informal y frecuente.

En este contexto el Microlearning se entiende como pequeñas dosis de contenidos, fragmentados de forma homogénea (pequeño y preciso), ubicuos (en diferentes contextos), continuos (relacionado con la repetición) y graduales (desde lo más pequeño a lo más grande, pudiendo insertarse un contenido con otro).

El Microlearnig es utilizado en distintos ámbitos, como lo es en la educación y en la capacitación a empleados. Se puede mencionar el caso de Google, que utiliza Microlearning en sus pequeños cursos “Whisper Courses” (cursos de susurro), implementado en correos electrónicos brindando una sugerencia a modo de “susurro” para que un gerente los pruebe. Más aun, esta empresa ha creado un sitio con material de Microlearning [1].

Entre los diversos y variados recursos que pueden utilizarse para la implementación del Microlearning en la educación, se encuentran: videos breves, imágenes, comics e historietas, memes, juegos, mapas conceptuales, infografías, flash cards y mensajes entre otros. Como repositorio de estos se pueden utilizar diversas plataformas de aprendizaje, blogs, páginas web y redes sociales.

En la actualidad hay varios trabajos relacionados con las redes sociales y su implementación en la educación. La red social más significativa relacionada con el Microlearning es Instagram, por su esencia de breve, instantáneo y donde las imágenes pueden funcionar como marcadores de contenidos. Existe la posibilidad de realizar un breve seguimiento de visitas y vistas a las publicaciones e historias. Con respecto a las Historias las cuales duran solo 1 día, se articulan perfectamente a la idea de tips, consejo del día o susurro; los cuales llevados a un ambiente de cursado en una materia bien podría ser un consejo sobre algo visto la clase anterior, o para reforzar contenidos a modo de susurros antes de un examen.

Existe la posibilidad de un seguimiento rápido respecto de los posteos realizados, pudiendo obtener información de cuantos y quienes las han visto.

Este artículo pone en relieve la importancia de los denominados 'susurros' como microcontenidos pedagógicos, integrados dentro de una estrategia de Microlearning en una asignatura específica de la Universidad Nacional de La Matanza. Esta investigación forma parte del trabajo de tesis para la Maestría en Tecnología Educativa de la Universidad Abierta Interamericana.

## 2. Trabajos Relacionados

Numerosos artículos abordan tanto el Microlearning como el uso de las redes sociales en la educación. A continuación, se presentan algunos de ellos en los apartados dedicados a Microlearning y Redes Sociales.

## 2.1. Microlearnig

De las extensas publicaciones y artículos académicos es oportuno mencionar algunas implementaciones de Microlearnig respecto de la enseñanza, en el ámbito Universitario como lo son:

El caso de la Universidad de Santiago de Chile [2] en un diseño curricular de la carrera de Ingeniería Mecánica.

Artículo de la Universidad de Ostrava (República Checa), para una materia de Arquitectura y Sistemas Operativos [3].

En otros niveles educativos, se toman dos publicaciones de universidades de España:

Caso de la Universidad de Coruña, para la enseñanza en un Laboratorio de Ciencia, donde participaron alumnos de grado educación inicial [4].

Caso de la Universidad de Murcia donde implementaron Microlearning en la enseñanza de música en el nivel secundario [5].

En cursos de formación y perfeccionamiento docente:

Universidades de España [6], [7], Israel [8] y Perú [9] [10].

## 2.2. Redes Sociales

Entre los trabajos realizados sobre Redes sociales en educación se pueden mencionar los siguientes:

Caso de la Universidad de Benin (UNIBEN) en el que se realizaron intervenciones en las redes sociales para complementar conferencias presenciales, en los que se implementaron mensajes con contenidos relacionados a unidades de instrucción independientes (una imagen y un texto descriptivo). Al finalizar la intervención los estudiantes manifestaron haberlos utilizado como una referencia móvil y como herramienta de estudio [11].

En la Universidad de Madrid, en la Carrera de ciencias del Deporte, se realizaron publicaciones en Instagram con actividades de descripción de ejercicios físicos y consejos nutricionales a lo largo de todo un año. Los resultados obtenidos fueron favorables [12].

## 3. Propuesta de Implementación

La propuesta que se describe es a partir de la construcción de diversos recursos los cuales se implementan a través de imágenes para realizar publicaciones en Instagram como posteos en el Feed, en Historias y videos en Reels. Con consejos a modo de “susurros” para los estudiantes, publicados luego del proceso de enseñanza aprendizaje del mismo en el aula.

Se realizan además “cuestionarios” en las Historias de la mencionada red social, a modo de repaso, antes de los exámenes.

A continuación, se exponen algunos de los recursos implementados a modo de ejemplo.

### 4.1. Susurros en Historias y Feed

Los susurros o consejos pueden presentarse en Historias o en el Feed de Instagram, cambiando el tamaño del contenido a mostrar, puesto que así lo requiere la plataforma para su mejor visualización. Se puede observar el mismo “susurro” en la Figura 1.



Figura 1. Susurros implementados en Instagram. A la izquierda en una Historia, y a la derecha en el Feed.

Como se mencionó anteriormente, una de las características del Microlearning es la integración de pequeños contenidos, que se van uniendo para formar conjuntos más amplios y completos. De esta manera, se construye un tema o contenido más complejo a partir de lo más simple.

En este contexto, se presenta la secuencia de objetos de Microlearning implementados como “susurros” en el feed de Instagram. Esta serie de “susurros” se detalla en la Tabla 1, que se encuentra a continuación, donde se comienza con un micro-tema como lo es; cuando se debe realizar Pasaje directo entre bases de distintos sistemas de numeración.

Luego se explica la relación que debe existir entre las bases para realizar el pasaje directo, para continuar con un ejemplo.

Paso seguido, se publica el susurro donde se muestra la regla práctica para realizar el complemento a la base de un número binario. Este tema es correlativo al anterior. Se finaliza con un ejemplo.

El tema que exhibe la Suma de dos números binarios en una ALU (Unidad Aritmética y Lógica) y muestra como lo realiza. Para que esto su entienda se debe tener conocimiento previo de pasar números de otras bases a binario. Si un número es negativo, se debe realizar el complemento a la Base, tema que debe asimilarse con anterioridad, para finalmente pasar a la suma.

Esta secuencia de pequeños temas reflejados en los susurros es como se encadenan esos objetos de Microlearning.

Tabla 1. Secuencia de Susurros.

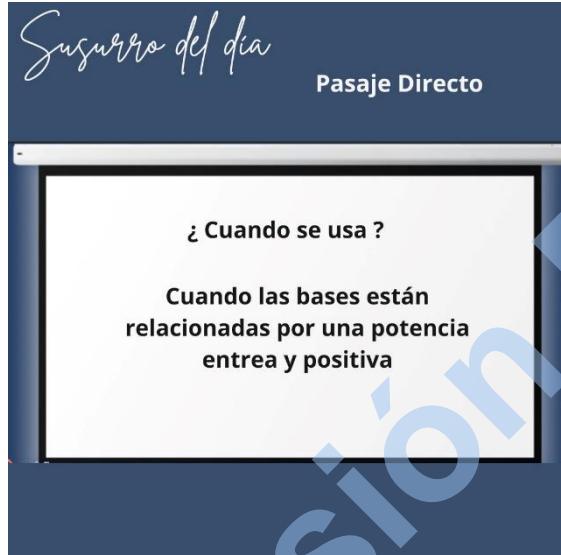
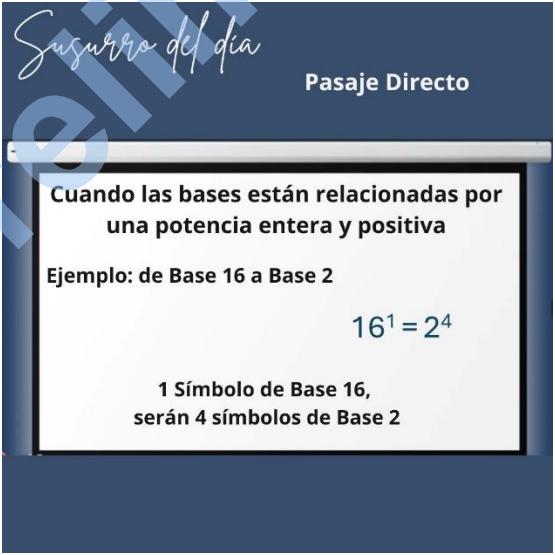
	
<p><i>Susurro del día</i></p> <p><b>Pasaje Directo</b></p> <p>¿Cuando se usa ?</p> <p>Cuando las bases están relacionadas por una potencia entera y positiva</p>	<p><i>Susurro del día</i></p> <p><b>Pasaje Directo</b></p> <p>Cuando las bases están relacionadas por una potencia entera y positiva</p> <p>Ejemplo: de Base 16 a Base 2</p> $16^1 = 2^4$ <p>1 Símbolo de Base 16, serán 4 símbolos de Base 2</p>

Figura 2. Susurro del día: Pasaje Directo.

Muestra cuando se debe utilizar el Pasaje Directo entre diferentes Bases de Sistemas de Numeración.

Figura 3. Susurro del día: Pasaje Directo 2.

Explica como debe ser la relación entre las Bases de un Sistema de Numeración para que se pueda realizar el Pasaje Directo.

*Susurro del día*

### Pasaje Directo Ejemplo

**Ejemplo: de Base 16 a Base 2**

$$16^1 = 2^4$$

B	3
1 0 1 1	0 0 1 1

Base 16      Base 2

Figura 4. Susurro del Día: Ejemplo de Pasaje Directo.

Se realiza el pasaje directo entre las Bases 16 y 2. Se puede observar que cada símbolo de Base 16 serán 4 símbolos de Base 2. Por la relación que existe entre ambas bases.

*Susurro del día*

### Complemento a la Base. Método Práctico

- Recorro de derecha a izquierda el número binario hasta el primer “1”.
- Luego cambio unos por ceros y ceros por unos.

**Ejemplo 1:**

0	0	1	0	1	0	1	1
1	1	0	1	0	1	0	1

**Ejemplo 2:**

0	0	1	0	1	1	0	0
1	1	0	1	0	1	0	0

Figura 5. Susurro del Día: Complemento a la Base.

Se describe de manera concisa el método práctico para obtener el complemento a la base de un número en binario. Este “susurro” incluye dos ejemplos ilustrativos.

*Susurro del día*

### Suma de dos números en Complemento a la Base.

**PASOS a Seguir:**

- 1) Pasar ambos números a binario en 8 bits.
- 2) Si puedo DEBO hacer pasaje Directo.
- 3) Paso los números y hay algún número negativo hago el Complemento a la Base.
- 4) Sumo en Binario
- 5) Muestro lo que da la ALU (Unidad Aritmética y Lógica)
- 6) Escribo como queda el Registro de Estados.

Figura 6. Susurro del Día: Suma de Números con Complemento a la Base.

Aquí se relatan brevemente los pasos a seguir para realizar una suma de dos números en binario.

*Susurro del día*

### Suma de dos números en Complemento a la Base.

**Sumar  $+2B_{16}$  y  $-73_8$  en una ALU de 8 bits**

1. Pasar a Binario ambos números
2. Completo con ceros a la izquierda para obtener 8 bits
3. Hago el Complemento a la Base SOLO del número que es negativo
4. Realizo la Suma.

2	B
0 0 1 0 1 0 1 1	Base 16
7	3
1 1 1 0 0 1	Base 8

0 0 1 0 1 0 1 1
1 1 0 1 0 1 0 1

Figura 7. Susurro del Día: Ejemplo de Suma con Complemento a la Base.

Se expone un ejemplo de suma de dos números en Complemento a la Base. Uno de los números está en Base 16 y el segundo en Base 8.

Se enumeran los pasos a seguir:

Se pasan ambos números a binario.

1. El número en base 8 al realizar el pasaje directo queda en 6 bits, se completa con ceros a la izquierda para obtener 8 bits.
2. Se realiza el Complemento a la Base del número negativo
3. Se muestra la suma realizada, cual es el resultado de la ALU (Unidad Aritmética y Lógica) y los flags del Registro de Estados.

## 4.2. Susurros en Reels

Los Reels de Instagram son muy vistos y pueden implementarse los mismos consejos haciendo aparecer de a poco, cada uno de los objetos intervenientes en la imagen, pudiendo agregarse voz o música. A continuación se muestra la secuencia de fotogramas que conforman el mini video en la Tabla 2.

Tabla 2. Secuencia de fotogramas en un Reels de Instagram.

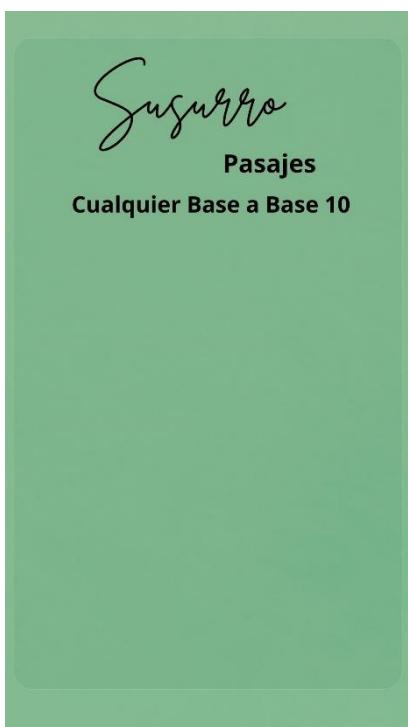


Figura 8. Fotograma 1

Tema: Pasaje entre bases. Cualquier base a Base 10  
Primer texto: Definición del tema.

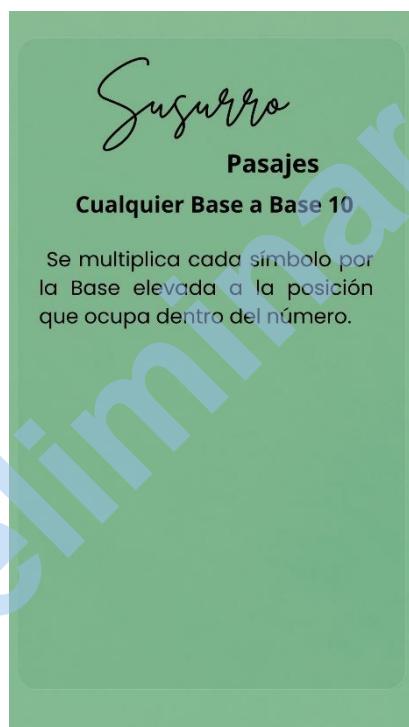


Figura 9. Fotograma 2.

Aparece la definición del Teorema Fundamental de la numeración.  
Lo que se debe hacer para realizar el pasaje.

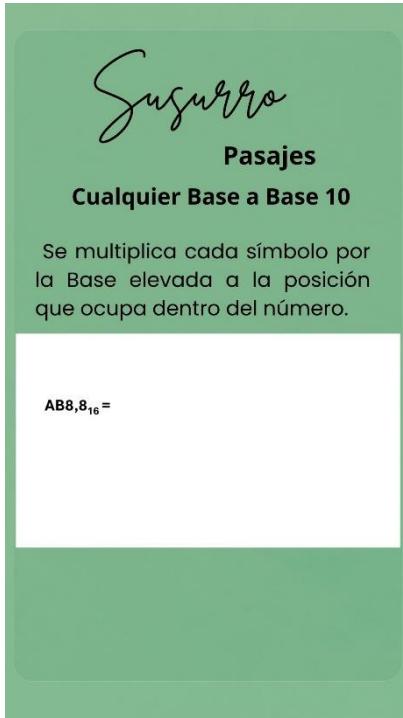


Figura 10. Fotograma 3.

Se propone un número en Base 16 para realizar el pasaje.

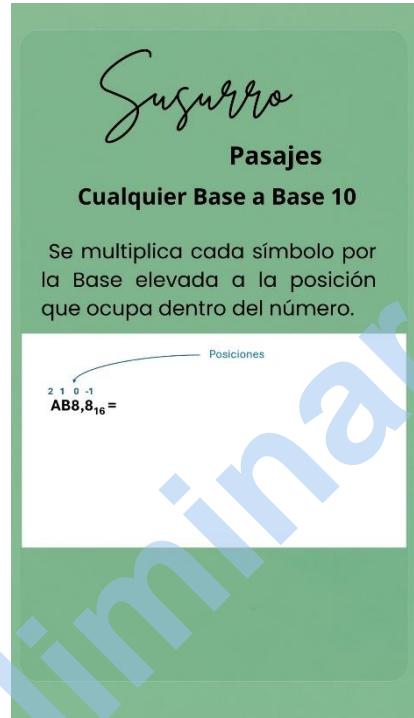


Figura 11. Fotograma 4

Se agregan las posiciones de cada uno de los símbolos que conforman el número.

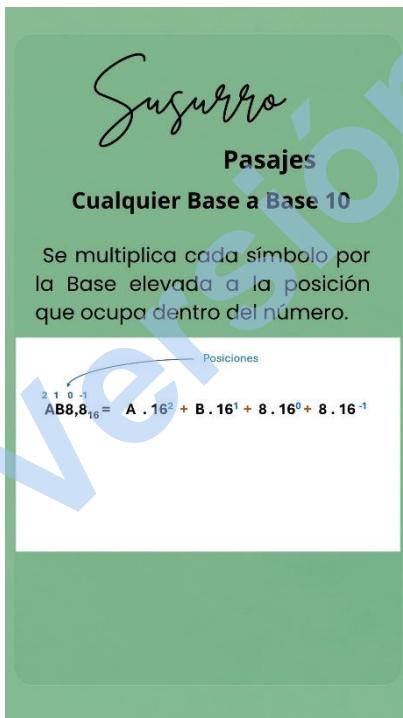


Figura 12. Fotograma 5

Se multiplica cada símbolo en base 16 por la base destino elevada a la posición que ese símbolo ocupa dentro del número.

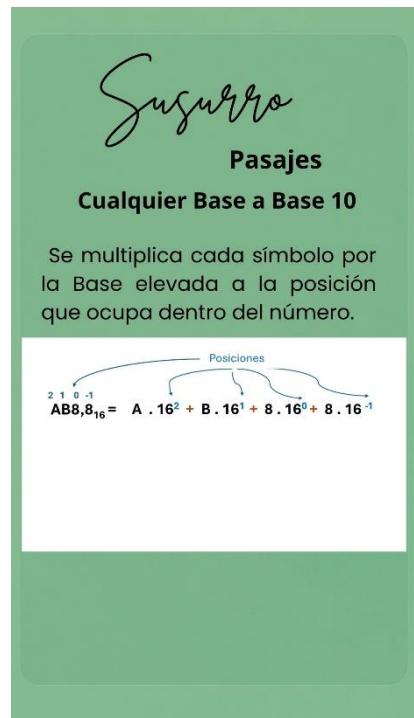


Figura 13. Fotograma 6

Se resalta que la potencia la cual la base está elevada corresponde con la posición del símbolo dentro del número.

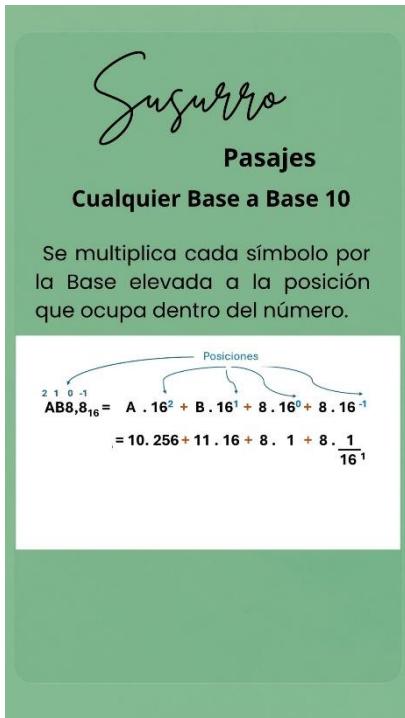


Figura 14. Fotograma 7

Se reemplaza el símbolo en Base 16 por su representación en Base 10 y se resuelven las potencias de la Base.

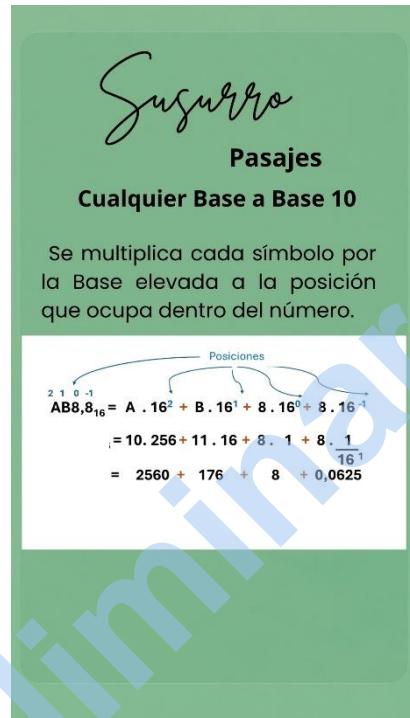


Figura 15. Fotograma 8

Se realizan las multiplicaciones.

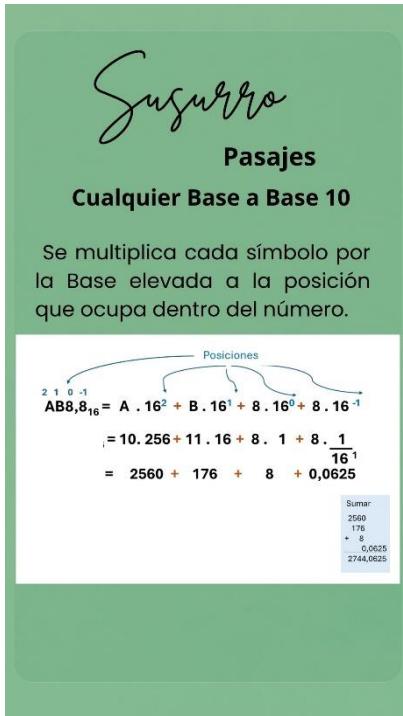


Figura 16. Fotograma 9

Aparece como cálculo auxiliar la suma.

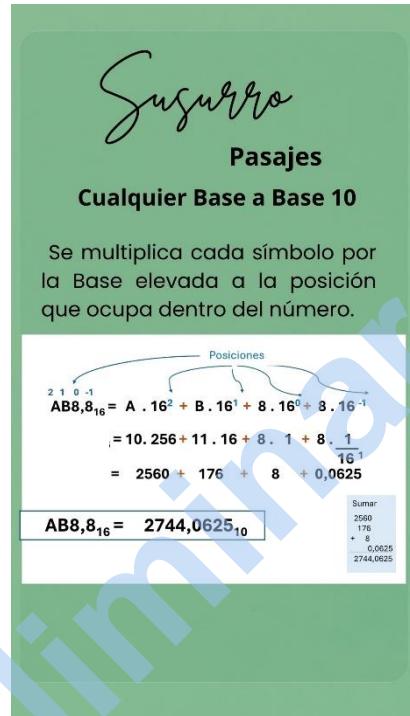


Figura 17. Fotograma 10

Se muestra el resultado: el número en base 16 y su equivalente en base 10.

#### 4.5. Resultados Obtenidos

Estos recursos de Microlearning se han desarrollado para un curso de la materia y se han implementado en Instagram. Es importante destacar que su uso no es obligatorio, sino que se consideran un complemento a los materiales ya existentes en la cátedra.

Se realizó una breve encuesta a los estudiantes de un curso obteniendo los resultados de 41 encuestados. En la Figura 18 se observa un breve resumen.

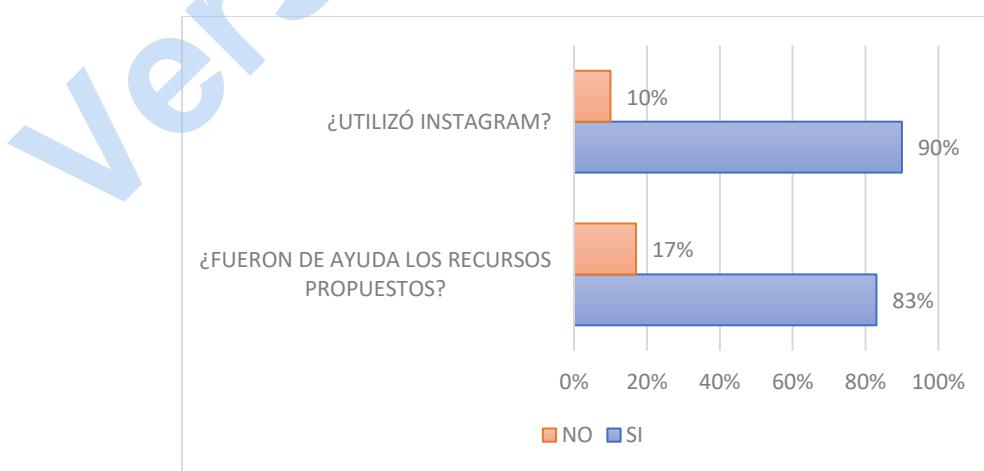


Figura 18. Resultados de la Encuesta.

La mayoría de los alumnos conocía y había utilizado los recursos de Microlearning. Como se muestra en la primera barra de la Figura 18, el 83% de los estudiantes indicó que estos recursos les fueron útiles para estudiar o repasar conceptos. Es relevante mencionar que el porcentaje de uso aumenta cuando se trata de la cuenta de Instagram implementada para el curso.

Como puede observarse todos los recursos son utilizados por la mayoría de los estudiantes, destacándose el acceso a Instagram.

## 5. Conclusiones

La implementación de "susurros" (consejos breves) basados en el concepto de Microlearning ha demostrado ser efectiva para mejorar el aprendizaje de los alumnos. Estos recursos han logrado hacer que el proceso de aprendizaje sea más eficaz, asertivo y agradable.

La excelente aceptación por parte de los estudiantes se refleja en los resultados de las encuestas realizadas. Esta experiencia sugiere que los "susurros" pueden ser aplicados en diversas materias y contenidos (teóricos o prácticos), y pueden ser integrados en cualquier plataforma o repositorio educativo.

Además, esta herramienta es versátil y puede ser utilizada en entornos de aprendizaje presencial, semipresencial (Blended Learning) o virtual. Los resultados indican que el aprendizaje puede ser divertido y percibido como una experiencia natural y cotidiana.

## Bibliografía

- [1] Whisper Courses de Google. Página web: <https://www.thinkwithgoogle.com/>
- [2] A. R. Avagliano., & A. S. Vega, “Mejora del Proceso de Enseñanza y Aprendizaje en la Carrera de Ingeniería de Ejecución Mecánica: Diseño Micro-curricular Basado en Resultados de Aprendizaje.” Formación universitaria, 2013
- [3] R. Polase, & T. Javorcik, “Results of pilot study into the application of MicroLearning in teaching the subject Computer Architecture and Operating System Basics.” In 2019 International Symposium on Educational Technology (ISET) (pp. 196-201). IEEE, 2019.
- [4] C. Dulsat-Ortiz, C. “Microenseñanza en el laboratorio de ciencias para el alumnado del grado de educación infantil”. Revista científica, 2019, (36), 367-380
- [5] J. Palazón Herrera, J. “Aprendizaje móvil basado en microcontenidos como apoyo a la interpretación instrumental en el aula de música en secundaria”. Píxel-Bit. Revista de Medios y Educación, 2015, 46, 119-136.
- [6] A. N. León, & P. R. Santiago. “Microenseñanza una técnica para motivar el enseñar y aprender investigando”. Perspectivas docentes, 2013, (52), 23-31.
- [7] Torrente Javier., et al. “Proactive: fomentando la creatividad de los profesores a través del aprendizaje basado en juegos - Una guía para Profesores”. Proyecto Europeo LLP (en línea), 2011
- [8] A. R. Y. E Peleberg, A. R. Y. E. Microenseñanza: un innovador procedimiento de laboratorio para mejorar la enseñanza y el entrenamiento de profesores. Traducción del original publicado en Unesco’s Bulletin Prospects in Education, 1(3), 1970.
- [9] R. Watson. La microenseñanza en la UPC. RIDU, 3(1), 5, 2007.
- [10] G.G Toro Mejíz, “La microenseñanza y el desempeño de los estudiantes de la especialidad de tecnología del vestido en la práctica docente continua de la Facultad de Tecnología de la Universidad Nacional de Educación Enrique Guzmán y Valle”, 2013.
- [11] I.I. Osaigbovo, & C.F. Iwegin,, 2028
- [12] M.L. Gómez-Ortiz. J. Bobkina & E.D Romero, 2020

# La Inteligencia Artificial en Educación Media de la República Argentina

Valeria González Angeletti<sup>1</sup>

<sup>1</sup>Universidad Nacional de Córdoba, Córdoba, Argentina  
[vgonzalez.cba@gmail.com](mailto:vgonzalez.cba@gmail.com)

## Resumen

El objetivo de este estudio fue analizar el nivel de inclusión del tema 'Inteligencia Artificial' en los sitios web de los Ministerios de Educación de cada jurisdicción educativa argentina. Presenta los datos recolectados en el mes de julio de 2024, y entre los hallazgos se evidencia una disparidad notable en la inclusión del área en los Diseños curriculares, y escasa Formación docente. La investigación subraya la necesidad de desarrollar políticas educativas que promuevan una integración equitativa de la IA en todo el país, asegurando que todos los estudiantes tengan acceso a una educación en el área, y permita el pleno desarrollo de la ciudadanía moderna y los desafíos del siglo XXI.

**Palabras clave:** Inteligencia artificial; Diseños curriculares; Nivel medio; Planes de estudio; Ciencias de la computación

## 1. Introducción

Somos la primera generación que -para algunas tareas puntuales- logró que las máquinas no se distingan de los humanos. La tecnología ha avanzado lo suficiente, como para volverse indistinguible de sus creadores. Caímos en lo que Brynjolfsson, Erik [1] denomina "La trampa de Turing", aquella donde los sistemas artificiales inteligentes imitan tan bien a los humanos, que hasta pueden reemplazarlos. La Inteligencia Artificial (IA) se ha ido integrando en los diferentes sectores productivos, como una fuerza transformadora que impacta en los procesos, estrategias, logística y toma de decisiones. Estudios de casos examinan cómo las tecnologías de IA -aprendizaje automático, procesamiento del lenguaje natural y robótica-, están revolucionando las industrias y redefiniendo el futuro [2].

Andrew Ng [3] afirma que la Inteligencia Artificial (IA) es la nueva electricidad. El autor expresa que el desarrollo acelerado de esta tecnología transformará todas las áreas de la vida y augura que, así como damos por sentado que las personas saben leer y escribir, algún día será común que los ciudadanos alberguen en sus conocimientos la codificación (escritura de código), en especial de IA. Es por este motivo que afirmamos que la IA provocará una revolución que necesariamente deberá incluirse en la educación formal de manera efectiva, ya que la aceleración tecnológica sin precedentes interpelará indefectiblemente las prácticas pedagógicas.

Introducidos en el tema, decimos que la Inteligencia Artificial es una disciplina incluida en las Ciencias de la Computación y entendida como aquella que utiliza sistemas informáticos para llevar a cabo tareas y actividades que históricamente dependían de la cognición humana, que implica, entre otros, diferentes utilidades; como la automatización de ciertas acciones, redacción textual, búsqueda de información, identificación de patrones, generación de asistentes de voz, etc. [4]. Más allá de las normativas, prácticas escolares y bases curriculares; la IA ya ha interpelado a las instituciones educativas: estudiantes y docentes usan sistemas de IA a diario. Los Estados utilizan sistemas de detección de riesgo de fracaso escolar con modelos de IA, los docentes para conocer las actividades que realizan sus estudiantes en espacios digitales, plataformas de colaboración, sistemas de búsqueda de la información, etc. [5].

De esta manera, el presente trabajo es una continuación de lo expuesto en el artículo 'Análisis Diseños Curriculares de Inteligencia Artificial en Educación Media'<sup>1</sup>[6]. Creemos necesario analizar cómo se introduce la IA en las escuelas de Nivel Medio de la República Argentina, qué saberes de IA circulan, qué y cómo se enseña la IA. Para ello, en este trabajo comenzaremos por explorar las páginas de los Ministerios de educación de cada Provincia argentina, sus Diseños curriculares y propuestas de Formación docente. Sistematizar esta información, puede contribuir a generar programas y dispositivos que orienten el abordaje de la IA en el secundario, para que no sea el mercado y los usos de los grupos

<sup>1</sup> Acceder al artículo <https://doi.org/10.24215/18509959.37.e19>

hegemónicos los que definan la oferta educativa de este campo en la escuela, sino que los contenidos estén orientados por criterios éticos, y de formación emancipadora y crítica [7].

## 2. Diseño curricular y Formación docente en Nivel Medio

### 2.1. Introducción

EL currículum obligatorio ha sido históricamente un tema de debate. El estudio histórico de la educación amplió los enfoques de investigación al incluir temas como las leyes y reformas educativas, la creación de instituciones y currículos, y los conflictos sobre la enseñanza y la formación docente. En Francia, la Revolución de 1789 marcó un punto de inflexión al transformar la educación en un campo de disputa política entre fuerzas progresistas y conservadoras. Este cambio reflejó la transición de la educación de un asunto privado a un problema social, impulsando la necesidad de medidas legales que acompañaran la difusión de nuevas ideas pedagógicas. Según Chartier [8], surgieron debates sobre la obligatoriedad, laicidad, gratuidad y financiación de la educación, así como la configuración del sistema escolar. La instauración de un cuerpo docente profesional y la modernización de las escuelas se presentaron como logros del Estado, que buscaba garantizar el acceso universal a la educación y fomentar la participación activa de la ciudadanía en la vida republicana.

En Argentina, las discusiones sobre la conformación y modernización del Sistema Educativo Nacional han sido una constante desde los inicios del Estado Nacional. Principalmente, uno de los principales desafíos fue cómo homogeneizar a una ciudadanía dispersa en un vasto territorio con un desarrollo desigual, en un esfuerzo por consolidar la nación. La creación del Consejo Nacional de Educación (CNE) en 1881 y la sanción de la Ley 1420 en 1884, fueron hitos fundamentales que impulsaron la institucionalización de un sistema público de educación centralizado. Esta ley consagró la laicidad, gratuidad y obligatoriedad de la educación primaria [9]. La necesidad de un sistema educativo moderno emergió como una respuesta a una realidad que se percibía como bárbara, con el objetivo de "torcerle la mano a un destino que apartaba a Argentina del proyecto civilizador" [10].

El sistema educativo secundario en nuestra República, surgió en un contexto de modernización estatal durante la última parte del siglo XIX, impulsado por la necesidad de formar ciudadanos aptos para integrarse en un proyecto nacional emergente. En 1863 se crea el primer colegio secundario, el Colegio Nacional de Buenos Aires. Acosta [11] expresa que inicialmente, la educación secundaria se estructuró en torno a la creación de diversas modalidades y tipos de instituciones como los bachilleratos, que se centraban en una formación generalista orientada al acceso universitario o trabajo estatal; y las escuelas técnicas, comerciales y normales, que respondían a las demandas específicas del mercado laboral y la formación docente. Esta segmentación interinstitucional reflejaba no sólo una diferenciación en las trayectorias educativas, sino también una jerarquización social en función de las finalidades y prestigio de cada modalidad, con los bachilleratos posicionándose como el principal modelo. En este marco, el currículum se describía como una herramienta fundamental para la homogenización y generalización de la enseñanza, especialmente a través del Plan Rothe<sup>2</sup>, que estableció un ciclo común en la enseñanza secundaria para uniformar la formación básica y posponer la especialización. Este proceso de generalización del currículum, contribuyó a la consolidación de un enfoque educativo más amplio y menos segmentado.

### 2.2. Diseños curriculares en Nivel medio

En el mundo, la disciplina del currículum surgió a principios del siglo XX en respuesta a nuevas dinámicas sociales, y se dividió en dos tendencias principales. La primera, impulsada por John Dewey, se centraba en las experiencias escolares y el desarrollo individual del estudiante. La segunda, representada por Franklin Bobbit, se enfocaba en una secuencia clara de contenidos educativos. Con sus diferencias, en ambos casos se destacaba la idea de estructura organizativa por disciplinas de la enseñanza en la escuela. La internacionalización de la disciplina del diseño curricular comenzó inicialmente desde la perspectiva de planes y programas, trascendiendo el debate interno de la comunidad académica de Estados Unidos para recibir aportaciones y desarrollos de todo el mundo. Se establecieron conceptos tales como el marco de referencia de un plan de estudios, análisis de la práctica profesional y el objeto de transformación. En España, Zabalza introdujo en la década de los ochenta el concepto de "desarrollo curricular" en contraposición al de "diseño curricular" (curriculum design). Este enfoque resaltaba la evolución y adaptación continua del currículo, en lugar de verlo como un conjunto fijo de planes y programas [12].

El diseño curricular trata entonces, la manera en que una sociedad elige, organiza, distribuye, transmite y evalúa el conocimiento educativo que considera público; y revela la distribución del poder y los principios de control social predominantes en dicha sociedad [13]. Es un dispositivo desarrollado para planificar el contenido de la enseñanza formal y cumple múltiples funciones. Un instrumento que sirve para planificar los contenidos a lo largo de un ciclo escolar,

<sup>2</sup> El Plan Rothe fue implementado en el año 1916 en Argentina, y buscó crear una base educativa común que permitiera uniformar el currículum y asegurar que todos los estudiantes recibieran una formación generalista antes de orientarse hacia caminos más especializados.

asegurando una distribución coherente y organizada del conocimiento. También, un documento normativo que especifica los contenidos obligatorios establecidos por cada país y jurisdicción (provincia). Los sociólogos del conocimiento señalan que los contenidos de cada currículum reflejan las orientaciones ideológicas de quienes los elaboran, sugiriendo que cada espacio curricular es un campo de batalla ideológico. Furlán expresa que su estructura formal implica una síntesis cultural que se manifiesta en diversas formas: documentos oficiales, planes de estudio, lineamientos, justificación, criterios de evaluación y acreditación, programas de formación docente, y en la creación de nuevas estructuras organizacionales, así como en innovaciones, cambios y reformas. Este proceso de estructuración no sólo define qué se enseña, sino también cómo y por qué, reflejando y perpetuando las dinámicas de poder y control social dentro de la sociedad [14].

En Argentina, los diseños curriculares han evolucionado significativamente desde la Ley Federal de Educación de 1993, hasta la Ley de Educación Nacional de 2006. Este cambio legislativo marcó un hito, al intentar promover una educación inclusiva y de calidad para todos los niveles educativos de las 24 jurisdicciones educativas nacionales. Pese a esto, si bien se otorgó mayor independencia local, se centralizaron los mecanismos de control en el gobierno nacional y profundizaron las diferencias, generando una fragmentación del sistema educativo [15]. A comienzos del siglo XXI, en Argentina surgió la necesidad de dar respuesta desde los diseños curriculares a los cambios sociohistóricos, formulando propuestas de alfabetización tecnológica que estén a la altura de los nuevos desafíos[16]. Sin embargo, los nuevos contextos escolares producto de la reformas impulsadas, evidenciaron una distribución desigual en la implementación del Diseño curricular y la oferta educativa: "(...) los resultados de la investigación educativa señalan una importante variabilidad de propuestas de enseñanza, al igual que una gran diversidad de maneras de concebir lo prioritario, lo significativo, lo que debe incluirse y excluirse como contenido escolar. (...)" [17]

### **2.3. Las Ciencias de la Computación en el Diseño Curricular de Nivel Medio**

Muchos países han impulsado la inclusión de las Ciencias de la Computación en la educación obligatoria, desarrollando diversas propuestas curriculares. Si bien su inclusión curricular fue casi nula, en Argentina se destacan algunas iniciativas centradas en la inclusión digital, como lo fue el programa Conectar Igualdad que distribuyó netbooks a estudiantes y docentes. Estudios [18, 19, 20] revelan cierto grado de insuficiencia en el abordaje de esta disciplina, generan propuestas, y formulan la necesidad de una reflexión sobre la incorporación de la computación equitativa en los planes de estudio. Desde 2015, la enseñanza de programación se ha considerado estratégica en nuestro país, reflejada en iniciativas curriculares como la incorporación de programación y robótica en los Núcleos de Aprendizajes Prioritarios de 2018. Sin embargo, la implementación de estos conocimientos varía entre las jurisdicciones educativas, lo que destaca la necesidad de un análisis para definir homogéneamente y equitativamente los contenidos y perspectivas futuras.

Entonces, entre los desafíos que se evidencian en la enseñanza de Ciencias de la Computación en las escuelas secundarias argentinas, se destaca la falta de uniformidad en la implementación del currículo de computación, con variaciones significativas entre provincias y escuelas. [21] expresan que a pesar de los esfuerzos recientes para incluir la disciplina como una materia troncal en las escuelas, la enseñanza de esta área de conocimiento sigue siendo un desafío, con una integración parcial y desigual. Aunque se han dado pasos importantes, los autores consideran necesario un enfoque más consistente y sistemático para asegurar que todos los estudiantes reciban una educación adecuada en la materia, lo que incluye la capacitación continua de docentes y la actualización de los recursos educativos.

Por otro lado, las líneas de investigación y desarrollo en torno a la inclusión de las Ciencias de la Computación en la educación secundaria se centran en varios aspectos clave. [22] afirman que, por un lado, la enseñanza de la computación como disciplina escolar busca justificar su incorporación en el currículo, destacando su importancia en la democratización del conocimiento y el desarrollo económico. Por el otro, se exploran los contenidos específicos de la computación, con el objetivo de diseñar modelos conceptuales que integren la disciplina de manera troncal en la escolaridad obligatoria. Además, se trabaja en la definición de enfoques metodológicos para la formación continua de docentes, abordando desde la formación disciplinar y didáctica, hasta la evaluación de prácticas educativas. Estas líneas de investigación se consideran interrelacionadas y se plantean como un esfuerzo conjunto para desarrollar modelos teórico-conceptuales que fortalezcan la enseñanza de la computación en la educación secundaria.

### **2.4. La necesidad de Formación docente en Inteligencia Artificial**

Al revisar la historia de la educación en Argentina, resulta interesante observar cómo desde sus inicios surgieron conceptos recurrentes como la "modernización del sistema educativo", el "modelo educativo orientado a las necesidades industrialistas", la "formación de ciudadanos frente a la exclusión", el "programa transformador del país" y la "necesidad de formar docentes titulados". Históricamente, dentro del proyecto de sistematización educativa surgió la necesidad de formar docentes acreditados, lo que llevó a la creación de escuelas normales en distintas regiones del país. Inicialmente, el Estado argentino estableció prioridades, y en este proceso, la cantidad de escuelas y la urgencia de su implementación prevalecieron sobre la formación docente, calidad educativa y una planificación adecuada. Sin embargo, con la implementación de las escuelas normales, la formación docente se convirtió en el eje de las expectativas, consolidando prácticas político-pedagógicas que influenciaron profundamente los métodos de enseñanza. Como señala [23], el

normalismo<sup>3</sup> fue fundamental para sostener la expansión educativa, y la formación docente nacional desempeñó un papel central en el proyecto civilizatorio. Este enfoque contribuyó a construir una cultura escolar modeladora, que dejó una huella identitaria al considerar a la escuela como el espacio privilegiado para la enseñanza.

En la actualidad, la creciente incorporación de tecnología en esferas ciudadanas, profesionales, industriales, educativas y empresariales requiere inevitablemente de habilidades técnicas y digitales especializadas. El objetivo de alcanzar un "pleno desarrollo de la ciudadanía", que se planteaba en la época de "civilización o barbarie" [24], junto con la necesidad de modernizar el Sistema Educativo Nacional y formar docentes bajo una pedagogía homogénea, siguen siendo temas relevantes. Así como la Ley 1420 de 1884 estableció que los responsables de la enseñanza debían poseer competencias certificadas [9], hoy los ciudadanos deben adquirir una "alfabetización en inteligencia artificial". Esto implica desarrollar competencias que les permitan evaluar críticamente las tecnologías de IA, comunicarse y colaborar efectivamente con ellas, y utilizarlas como herramientas para desempeñar diversas actividades[25].

En 2015, la Declaración de Qingdao subrayó la importancia de desarrollar planes de estudio en los institutos de formación docente que los capaciten para utilizar adecuadamente las TIC y fomentar innovaciones tecnológicas [26]. La UNESCO [27] también destacó la necesidad de un compromiso gubernamental sólido para el desarrollo de programas de formación docente y la creación de un currículo adecuado. Debe ser inmediato que las políticas educativas modernas incluyan una formación docente integral en IA, permitiendo así un desarrollo transversal de conocimientos en esta área. Por su lado, el Consenso de Beijing [28] refuerza esta idea al señalar la necesidad de revisar y redefinir continuamente las competencias docentes, para trabajar eficazmente en entornos educativos donde la IA juega un papel central. Se evidencian algunos esfuerzos impulsados por universidades y programas nacionales, como Program.ar<sup>4</sup>, la Fundación Sadosky<sup>5</sup>, HumanIA<sup>6</sup>; que ofrecen orientación en la implementación de conocimientos en Ciencias de la Computación en las escuelas [29]; y el plan Argentina Programa 4.0<sup>7</sup>, enfocado en la industria del software, que fomentan que el sistema educativo pueda adaptarse a las demandas actuales.

### 3. La Inteligencia Artificial en el Nivel medio

#### 3.1. Introducción

El término Inteligencia Artificial ha sido utilizado en diversos campos y aplicado con diferentes sentidos. Es con los recientes logros en *Deep Learning* (aprendizaje profundo), que logra notoriedad social y la utilización de sus aplicaciones atrae a los usuarios. En 1995 Wang [30] la definió como la capacidad de un sistema de procesamiento de información para adaptarse a su entorno mientras opera con conocimientos y recursos. La concepción de la IA varía según el contexto de uso, lo que subraya la importancia de revisar y sintetizar los factores que influyen en la disposición de los usuarios a adoptar sistemas de IA en diversos sectores. Una definición aceptada de la IA, es aquella entendida como una entidad no natural capaz de cumplir o superar las tareas asignadas, considerando factores culturales y demográficos.

Según [31], la IA se clasifica en tres categorías: Inteligencia Artificial Estrecha (*Artificial Narrow Intelligence*, ANI), que incluye sistemas de inteligencia artificial modernos, como el software de reconocimiento de voz (por ejemplo, Siri de Apple), que ayuda a los usuarios a través del aprendizaje automático y no puede transferir conocimientos entre sistemas o tareas; Inteligencia Artificial General (*Artificial General Intelligence*, AGI), que es teórica y podría funcionar de manera autónoma en múltiples escenarios; y Superinteligencia Artificial (*Artificial Super Intelligence*, ASI), que es la forma más precisa de IA, ya que superaría la capacidad humana y será capaz de ser pionera en descubrimientos en los campos general, científico, académico, creativo y social, lo que podría conducir a la redundancia de seres humanos.

Existe un creciente interés por implementar la IA en educación, se distinguen diversas iniciativas internacionales recientes como la *AI+Education Summit* (2023) [35], organizado por el *Stanford Institute for Human-Centered Artificial Intelligence* (HAI) y el *Stanford Accelerator for Learning*, que reunió a líderes académicos y de la industria para explorar cómo las tecnologías de IA, como los modelos de lenguaje, la visión por computadora y la realidad aumentada/virtual, pueden transformar la enseñanza y el aprendizaje. También *EnlightED 2023* [36] celebrada en Madrid, donde se discutió el impacto de la IA en el aprendizaje y el mercado laboral. En 2023, en Argentina se llevaron a cabo eventos como el Congreso Internacional de Inteligencia Artificial y Educación (CIIAE)<sup>8</sup>, con un enfoque en las aplicaciones de la IA en la educación y la investigación; el Congreso Internacional de Tecnología Educativa y Educación a Distancia (CITEED)<sup>9</sup> donde se abordaron temas relacionados con la IA y la educación a distancia y las nuevas tecnologías en el aula. También,

<sup>3</sup> El 'normalismo' se refiere a la respuesta del Estado a la necesidad de establecer el Sistema Educativo Nacional, creando escuelas normales en distintos lugares del país, que brindaran titulación a maestros de enseñanza primaria.

<sup>4</sup> <https://program.ar/>

<sup>5</sup> <https://fundacionsadosky.org.ar/>

<sup>6</sup> <https://www.chicos.net/public/humania/formacion-webinars>

<sup>7</sup> <https://www.argentina.gob.ar/economia/conocimiento/argentina-programa>

<sup>8</sup> <https://ciiae.isistan.unicen.edu.ar/>

<sup>9</sup> <https://revistacolegio.com/events/iii-congreso-internacional-de-tecnologia-aprendizaje-y-educacion-citae-2023/>

el simposio de Innovación Educativa y Tecnología (SIET)<sup>10</sup>, centrado en innovaciones tecnológicas, incluyendo la IA, para mejorar los procesos educativos.

### 3.2. La inclusión de la Inteligencia Artificial en Nivel Medio

Se prevé que la IA penetre en la mayoría de las industrias; provocando un cambio significativo a nivel profesional, educativo, industrial y social. La Alfabetización en IA emerge como una habilidad esencial del siglo XXI, combinando ciencia de datos, pensamiento computacional y conocimiento multidisciplinario, con un enfoque en el uso ético y crítico de la tecnología. Según UNESCO [37] el término "Alfabetización en IA" abarca tanto la comprensión de cómo la IA maneja los datos, como el entendimiento de los algoritmos que detectan patrones en esos datos, estructurando así el campo emergente de la alfabetización en IA como una competencia crucial para la sociedad moderna. Según [38] se presentan diferentes enfoques para la enseñanza de la IA, que van desde perspectivas disciplinarias hasta enfoques interdisciplinares y transdisciplinares, reconociendo que la resolución de problemas complejos requiere la integración de diversos campos del conocimiento. Elementos clave para la implementación efectiva de estos contenidos incluyen la formación docente y la disponibilidad de materiales y herramientas adaptadas a los contextos escolares, respetando las culturas locales.

Por otro lado, a pesar de los desafíos que se presentan a la hora de pensar los usos en educación de la IA, ciertos agentes (chatbots educativos, robótica y plataformas de auto-aprendizaje, entre otros), ofrecen ventajas significativas, como la posibilidad de una mayor personalización del aprendizaje y la automatización de ciertos procesos. Además, [32] afirma que la IA puede transformar profundamente los métodos de enseñanza, los procesos pedagógicos y la formación docente. Entre los desafíos y oportunidades que presenta este tipo de tecnología se encuentran la necesidad de una alfabetización digital/tecnológica/computacional; y la necesidad de desarrollar competencias pedagógicas orientadas a construir un pensamiento científico y tecnológico. [33] señalan algunos retos a resolver como la formación del profesorado en el uso de nuevos recursos educativos, junto con el diseño de modelos pedagógicos innovadores que integren la realidad inmersiva. Esto debe ir acompañado del desarrollo de diseños curriculares de alfabetización en IA y de la consideración de desafíos éticos, como evitar sesgos, garantizar la protección de datos de los estudiantes, y abordar los derechos de propiedad intelectual. También, se requiere optimizar los recursos y redefinir el paradigma de la interacción máquina-humano, para ofrecer un seguimiento personalizado a los estudiantes.

Asimismo, las recomendaciones de UNESCO [39] sobre políticas para implementar la enseñanza de la IA en nivel medio, están orientadas a construir una estrategia integral que abarque varios aspectos clave:

- Visión y prioridades estratégicas integrales: es necesario establecer una visión clara y estratégica que defina los objetivos a largo plazo para la inclusión de la IA en la educación. Esto incluye identificar las áreas prioritarias y alinearlas con las necesidades del sistema educativo y del mercado laboral.
- Planificación interdisciplinaria y gobernanza intersectorial: la implementación de la IA en la educación requiere una colaboración interdisciplinaria entre diferentes sectores, incluyendo la educación, la tecnología, la economía, y la política. Es fundamental que haya una gobernanza sólida que coordine estos esfuerzos y asegure una integración coherente y eficaz de la IA en los currículos escolares.
- Equidad, inclusión y ética en el uso de la IA: las políticas deben garantizar que la IA se utilice de manera equitativa e inclusiva, respetando los principios éticos y evitando la exacerbación de desigualdades preexistentes. Esto implica diseñar políticas que promuevan el acceso igualitario a las tecnologías de IA y que minimicen los riesgos de sesgo algorítmico.
- Planes para la gestión, enseñanza, aprendizaje y evaluación: es crucial desarrollar planes detallados para la integración de la IA en la gestión educativa, en la enseñanza, y en los métodos de aprendizaje y evaluación. Estos planes deben incluir el uso de la IA para personalizar el aprendizaje y mejorar los resultados educativos, así como para capacitar a los docentes en nuevas metodologías.
- Pruebas piloto, monitoreo y evaluación: antes de una implementación a gran escala, es recomendable realizar pruebas piloto que permitan evaluar la eficacia de la IA en contextos educativos específicos. El monitoreo continuo y la evaluación rigurosa son necesarios para ajustar las políticas y estrategias según sea necesario, asegurando que la inclusión de la IA realmente beneficie al sistema educativo.
- Promoción de innovaciones locales: fomentar la innovación local en el desarrollo y la implementación de tecnologías de IA adaptadas a las necesidades específicas de cada región o comunidad es esencial. Esto incluye apoyar a los educadores y desarrolladores locales en la creación de herramientas y recursos que respondan a las realidades y desafíos locales.

<sup>10</sup> <https://www.lasalle.edu.ar/presentamos-las-memorias-del-congreso-iet-2023>

Estas recomendaciones subrayan la importancia de un enfoque estratégico y bien coordinado para la implementación de la IA en la educación, asegurando que los beneficios potenciales de esta tecnología se aprovechen al máximo mientras se mitigan los riesgos asociados.

## 4. Metodología de trabajo

Este estudio se desarrolló en base al **enfoque mixto cualitativo-cuantitativo**, orientado a la exploración de los sitios web oficiales de cada Ministerio de Educación provincial, seleccionando una **muestra no probabilística** de los Diseños Curriculares disponibles y propuestas Formación docente vigentes en la educación media. Su finalidad es visibilizar la presencia o ausencia de contenidos específicos sobre IA en cada jurisdicción educativa argentina. Para ello, dentro del universo de material disponible en línea en cada sitio web ministerial, se identificaron y seleccionaron los Documentos de Diseños curriculares del Nivel medio, y las Propuestas de capacitación docente, con la finalidad de construir las siguientes unidades de análisis:

- **Diseño curricular:** la muestra incluye los Diseños curriculares oficiales de nivel medio disponibles en los sitios web de los Ministerios de educación de las 24 jurisdicciones de Argentina (23 provincias y la Ciudad Autónoma de Buenos Aires), y el Ministerio de Educación nacional. Se seleccionaron documentos que contengan información relevante sobre la inclusión de la Inteligencia Artificial (IA) en materias de Ciclo Básico -como Tecnología (o equivalente)-, Orientaciones de Ciclo orientado como Informática (o similar), colegios técnicos de nivel medio y formación docente de nivel terciario para nivel medio.
- **Formación docente:** la muestra abarca los programas específicos de formación docente relacionados con la enseñanza de la IA, que estén publicados en los portales de los Ministerios de Educación provinciales. Esto incluye cursos, talleres, y módulos de actualización profesional.
- **Propuesta curricular:** incluye también una muestra de propuestas curriculares en IA que no se encuentran incluidas en los Diseños curriculares, pero sí presentes en los sitios web relevados.
- **Propuesta de recursos:** cualquier material didáctico o recurso educativo relacionado con la IA que esté disponible en los sitios web oficiales de las provincias, tales como manuales, guías de trabajo, y herramientas tecnológicas diseñadas para la enseñanza en el nivel medio.

Además, se detectó material complementario que se creyó oportuno incorporar a esta investigación por su aporte en el tema, creando las siguientes unidades de análisis:

- **Políticas públicas:** en caso de ser evidente en los sitios consultados, se seleccionaron las políticas públicas y marcos normativos provinciales que guían la implementación de IA en el currículo escolar y la formación docente.
- **Formación ciudadana:** se trata de formación orientada al público general, que puede ser o no aprovechada por los docentes.

Por otro lado, con motivo de lograr una mejor interpretación y organización de la información recolectada, se construyeron las siguientes **categorías de análisis y variables**:

- **Categoría 1: presencia de contenidos** relacionados con Inteligencia Artificial en los portales oficiales de los Ministerios de Educación de cada provincia.
  - Variable 1: presencia de contenidos IA en Sitios web.
- **Categoría 2: Grado de inclusión** de la Inteligencia Artificial en los Diseños Curriculares de la educación media en cada jurisdicción educativa.
  - Variable 2: inclusión de IA en Diseños curriculares.
- **Categoría 3: Disponibilidad y características** de los programas de capacitación docente sobre Inteligencia Artificial en cada jurisdicción.
  - Variable 3: capacitación docente en IA.

También, se diseñaron **instrumentos de exploración y recolección de datos**, con la finalidad de obtener la información necesaria para el análisis. Los principales métodos e instrumentos incluyeron:

- **Exploración en Internet:** la recopilación de documentación y fuentes de datos relevantes, fue realizada en el motor de búsqueda de Google, bajo los siguientes criterios de búsqueda:
  - Ministerio de Educación de la provincia de (nombre de la jurisdicción) en Argentina.

- Una vez identificada la página web oficial del Ministerio de educación de cada jurisdicción nacional, la búsqueda se orientó a explorar cada sitio en la búsqueda del siguiente material:

- Diseño curricular
- Orientaciones curriculares
- Propuestas curriculares
- Sugerencias curriculares
- Cuadernillos de acompañamiento docente
- Material complementario de los diseños curriculares
- Formación docente
- Capacitación docente

En este punto, cabe aclarar que se descartó la búsqueda con modelos de Inteligencia Artificial Generativa como *Gemini*, *ChatGPT* o *Copilot*; por notar un alto grado de ‘alucinación’<sup>11</sup> en las respuestas y datos erróneos.

- **Revisión Documental:** una vez identificado cada documento, como indica [41], se generó un procedimiento sistemático para examinar dentro de cada uno, a modo de revisar y evaluar la presencia o ausencia de las siguientes palabras clave: 'Inteligencia Artificial', 'IA', 'aprendizaje automático', 'machine learning', 'redes neuronales', 'procesamiento del lenguaje natural', 'automatización', 'tecnologías emergentes', 'algoritmos', y 'innovación educativa'. Dentro de las fases de análisis, se incluyó la lectura de los textos, la identificación de sus características y tipologías, su codificación en las unidades de análisis creadas, y cuantificación de temas.
- **Ficha de registro:** una vez finalizada la exploración de datos y recolectada la información, se ordenó de manera de facilitar su análisis en lo que [41] denomina “Ficha de registro”. En este caso, el instrumento fue creado en una planilla de cálculo con las distintas jurisdicciones educativas y las siguientes columnas: Región del país, Jurisdicción, ¿Presenta IA?, Categoría, Sitio, Acceso, Contenidos.

Asimismo, la recopilación de los datos se llevó a cabo a través de un **corte de tiempo transversal que abarcó el mes de julio de 2024**.

#### 4.1. Hallazgos de investigación

Para la presentación de los datos recolectados, se organizó la información en el siguiente instrumento de visualización gráfica: [ingresar aquí al instrumento de visualización de datos](#)<sup>12</sup>. Entre los hallazgos más importantes, mencionamos:

La Tabla 1, constata la cantidad de unidades de análisis encontradas por jurisdicción (Variable 1: presencia de contenidos IA en Sitios web):

Jurisdicción	¿Presenta IA?	Unidades de análisis							Total
		Formación Docente	Diseño Curricular	Propuesta de recursos	Propuesta curricular	Formación Ciudadana	Políticas Públicas		
Nación	Sí	11	3	1	1	3	0	19	
Ciudad Autónoma de Buenos Aires (CABA)	Sí	0	6	0	0	4	1	11	
Provincia de Buenos Aires	Sí	1	4	0	0	0	0	5	
Córdoba	Sí	6	4	0	0	1	0	11	
Entre Ríos	No	0	0	0	0	0	0	0	
Santa Fe	Sí	0	2	0	0	0	0	2	
Catamarca	No	0	0	0	0	0	0	0	
Jujuy	Sí	2	3	0	0	0	0	5	

<sup>11</sup> [40] definen a las alucinaciones en los modelos de lenguaje grande (LLM) como la generación de contenido que es sin sentido o infiel al contenido fuente proporcionado. Esto incluye una amplia gama de fallos en la fidelidad y veracidad, y se enfoca en un subconjunto llamado confabulaciones, donde los LLMs hacen afirmaciones que son incorrectas y arbitrarias.

<sup>12</sup> <https://sites.google.com/view/presenciadeia/planilla-datos>

La Rioja	Sí	1	2	0	0	0	0	3
Salta	Sí	2	1	0	0	0	0	3
Santiago del Estero	Sí	2	0	0	0	0	0	2
Tucumán	Sí	2	0	0	0	0	0	2
Chaco	Sí	1	1	0	0	0	0	2
Corrientes	Sí	2	0	0	0	0	0	2
Formosa	No	0	0	0	0	0	0	0
Misiones	Sí	4	1	0	0	0	0	5
Mendoza	Sí	2	1	0	0	0	0	3
San Juan	No	0	0	0	0	0	0	0
San Luis	Sí	0	3	0	0	0	0	3
Chubut	Sí	0	1	1	0	1	0	3
La Pampa	Sí	1	3	0	0	0	0	4
Neuquén	Sí	4	2	0	0	1	0	7
Río Negro	Sí	2	1	0	0	0	0	3
Santa Cruz	No	0	0	0	0	0	0	0
Tierra del Fuego, Antártida e islas del Atlántico Sur	No	0	0	0	0	0	0	0

Tabla 1: cantidad de unidades de análisis por jurisdicción

El siguiente gráfico, representa el total de unidades de análisis encontradas por cada jurisdicción. Se observa que Nación, CABA y Córdoba son las jurisdicciones con mayor diversidad y cantidad de unidades de análisis presentes. También, que Formosa, San Juan, Entre Ríos, Santa Cruz, Tierra del Fuego y Catamarca, no tienen presentan unidades de análisis categorías registradas.

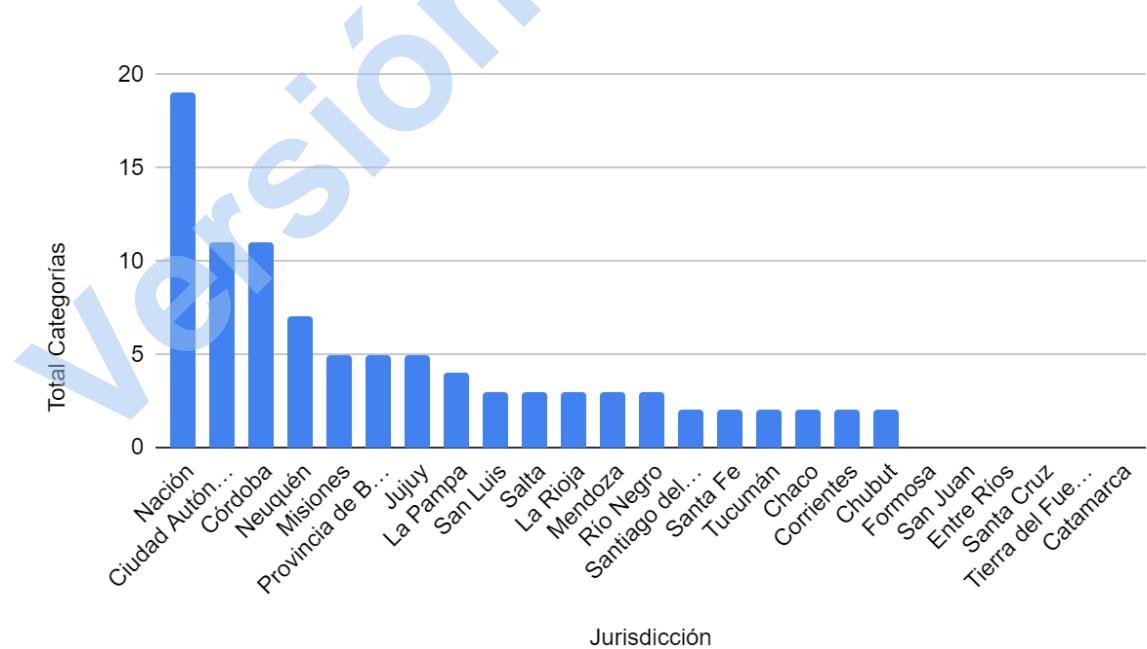


Gráfico 1: unidades de análisis por jurisdicción

Analizando el gráfico anterior, podemos agrupar las jurisdicciones en tres grupos según el nivel de presencia de IA (**Variable 1 - presencia de contenidos IA en Sitios web**):

- **Grupo Bajo (0-3%):** escasa o nula presencia de IA en los sitios web de sus Ministerios de Educación

- Entre Ríos: 0%
  - Catamarca: 0%
  - Formosa: 0%
  - San Juan: 0%
  - Santa Cruz: 0%
  - Tierra del Fuego, Antártida e islas del Atlántico Sur: 0%
  - Santa Fe: 2%
  - Santiago del Estero: 2%
  - Tucumán: 2%
  - Chaco: 2%
  - Corrientes: 2%
  - La Rioja: 3%
  - Salta: 3%
  - Mendoza: 3%
  - San Luis: 3%
  - Chubut: 3%
  - Río Negro: 3%
- **Grupo Medio (4-9%):** moderada presencia de IA en los sitios web de sus Ministerios de Educación
    - La Pampa: 4%
    - Provincia de Buenos Aires: 5%
    - Jujuy: 5%
    - Misiones: 5%
    - Neuquén: 7%
  - **Grupo Alto (10-20%):** mayor presencia de IA en los sitios web de sus Ministerios de Educación, que en el resto de los sitios consultados.
    - Ciudad Autónoma de Buenos Aires (CABA): 12%
    - Córdoba: 12%
    - Nación: 20%

Las diferencias observadas en la presencia de IA en los sitios web de los Ministerios de Educación de las distintas jurisdicciones reflejan:

- **Grupo Bajo (0-3%):** las jurisdicciones en este grupo presentan una escasa o nula presencia de contenidos relacionados con IA en sus portales educativos oficiales. Esto incluye a provincias como **Entre Ríos, Catamarca, Formosa, San Juan, Santa Cruz, y Tierra del Fuego, Antártida e islas del Atlántico Sur**, que muestran un 0% de contenidos de IA. Además, otras provincias como **Santa Fe, Santiago del Estero, Tucumán, Chaco, Corrientes, La Rioja, Salta, Mendoza, San Luis, Chubut, y Río Negro** tienen una presencia muy limitada, con porcentajes que oscilan entre el 2% y el 3%. Este grupo indica una necesidad mayor de desarrollo e inclusión de IA en los recursos educativos accesibles a través de los sitios web oficiales.
- **Grupo Medio (4-9%):** este grupo incluye a jurisdicciones con una presencia moderada de IA en sus portales web, con porcentajes que van del 4% al 7%. **La Pampa, Provincia de Buenos Aires, Jujuy, Misiones, y Neuquén** se encuentran en esta categoría. Estas provincias muestran un esfuerzo inicial por incluir IA en sus contenidos educativos en línea, aunque todavía tienen un margen significativo de mejora para alcanzar una inclusión más completa y robusta.
- **Grupo Alto (10-20%):** las jurisdicciones que conforman este grupo han logrado una presencia de IA en los sitios web de sus Ministerios de Educación, mayor que el resto. **Ciudad Autónoma de Buenos Aires (CABA) y Córdoba** alcanzan un 12% de contenidos relacionados con IA, mientras que **Nación** lidera con un 20%. Estas jurisdicciones están claramente en ventaja en términos de accesibilidad y promoción de contenidos de IA a través de sus portales educativos, lo que refleja un mayor compromiso en el tema.

La variable '**Inclusión de IA en Diseños Curriculares**' mide el nivel de incorporación de la IA en los Diseños curriculares en las distintas jurisdicciones de Argentina. En base a los resultados, se distinguen los siguientes niveles de Inclusión:

- **Alta inclusión:** **Ciudad Autónoma de Buenos Aires (CABA)** destaca con el mayor puntaje (6), lo que indica una mayor inclusión de contenidos de IA en los Diseños curriculares. La **Provincia de Buenos Aires y Córdoba** presentan un puntaje de 4, evidenciando un esfuerzo significativo para incorporar la IA en diversas áreas del currículo.
- **Moderada inclusión:** **Nación, Jujuy, San Luis, y La Pampa** tienen un puntaje de 3, lo que indica que han incluido en cierto nivel la IA en el currículo, aunque de manera menos extensa que las jurisdicciones del grupo

anterior. **Santa Fe, La Rioja, y Neuquén** con un puntaje de 2, también muestran una inclusión moderada, pero en menor nivel.

- **Baja o Nula inclusión:** varias provincias muestran una inclusión baja o nula de IA en sus currículos: **Salta, Chaco, Misiones, Mendoza, Chubut, y Río Negro** tienen un puntaje de 1, lo que sugiere una implementación mínima de IA en sus diseños curriculares. Por otro lado, **Santiago del Estero, Tucumán, Corrientes, Formosa, Santa Cruz, Entre Ríos, Catamarca, y San Juan y Tierra del Fuego** no han incluido la IA en sus currículos (puntaje 0).

La variable '**Capacitación Docente en IA**' mide el nivel de oferta y participación en programas de formación docente específicos sobre Inteligencia Artificial (IA) en las distintas jurisdicciones de Argentina. A continuación, se presenta un análisis de la distribución general de la oferta de Formación Docente en IA:

- **Alta Capacitación:** **Nación** lidera con 11 puntos, lo que refleja un fuerte enfoque en la capacitación docente en IA. **Córdoba** sigue con 6 puntos, mientras que **Misiones y Neuquén** alcanzan los 4 puntos cada una. Estas jurisdicciones están haciendo un esfuerzo significativo para preparar a sus docentes en IA.
- **Moderada Capacitación:** un grupo considerable de provincias se incluye en la categoría, con puntajes que van de 1 a 3. Las provincias de **Buenos Aires, Jujuy, Salta, Santiago del Estero, Tucumán, Chaco, Corrientes, Mendoza, La Pampa y Río Negro**; muestran que, si bien hay esfuerzos para capacitar a los docentes en IA, estos son limitados y podrían beneficiarse de una expansión y profundización.
- **Baja o Nula Capacitación:** varias provincias, incluidas **CABA, Entre Ríos, Santa Fe, Catamarca, Formosa, San Juan, San Luis, Chubut, Santa Cruz y Tierra del Fuego**; no muestran ninguna iniciativa de capacitación docente en IA (puntaje 0).

La siguiente matriz de correlación muestra la relación entre las diferentes variables del conjunto de datos. Como punto destacado de la correlación entre variables mencionamos:

- **Formación Docente y Diseño Curricular:** existe una correlación positiva moderada (0.57), lo que indica que a medida que aumenta la formación docente, también tiende a mejorar el diseño curricular.
- **Diseño Curricular y Políticas Públicas:** existe una correlación positiva (0.65), lo que sugiere que mejores diseños curriculares están asociados con mejores políticas públicas.

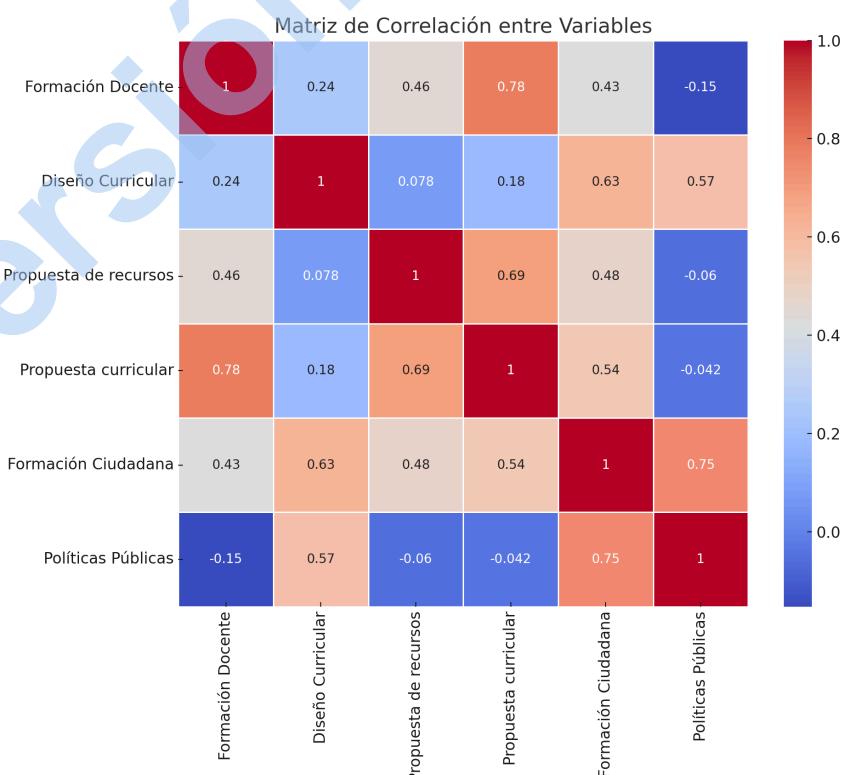
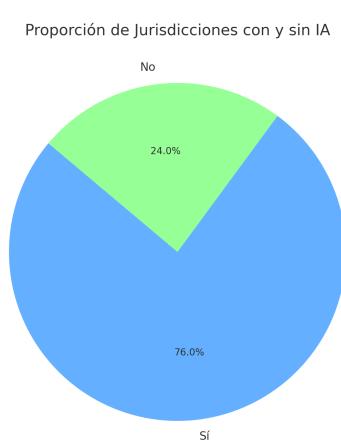


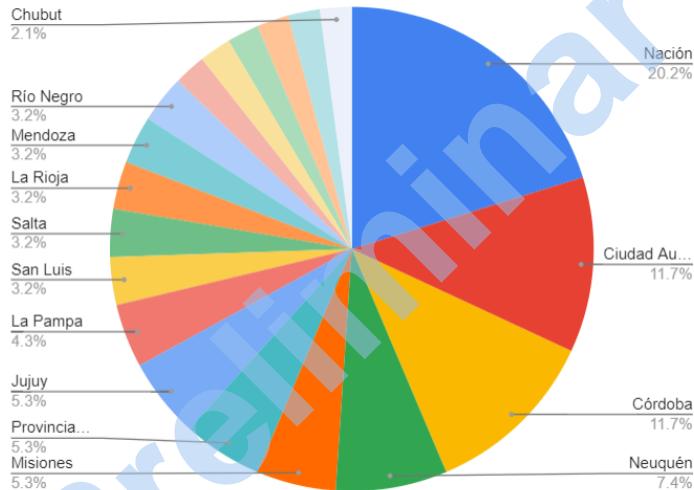
Gráfico 2: matriz de correlación

El siguiente gráfico circular muestra la proporción total de jurisdicciones con y sin presencia de IA. Según el Gráfico 3, la proporción total de presencia todas las jurisdicciones es:

- Sí (IA presente): 80%
  - No (IA no presente): 20%



*Gráfico 3: proporción total de jurisdicciones con y sin presencia de IA*



*Gráfico 4: proporción de cada jurisdicción*

A continuación, se expone un mapa de la distribución geográfica de unidades de análisis identificadas en cada provincia;

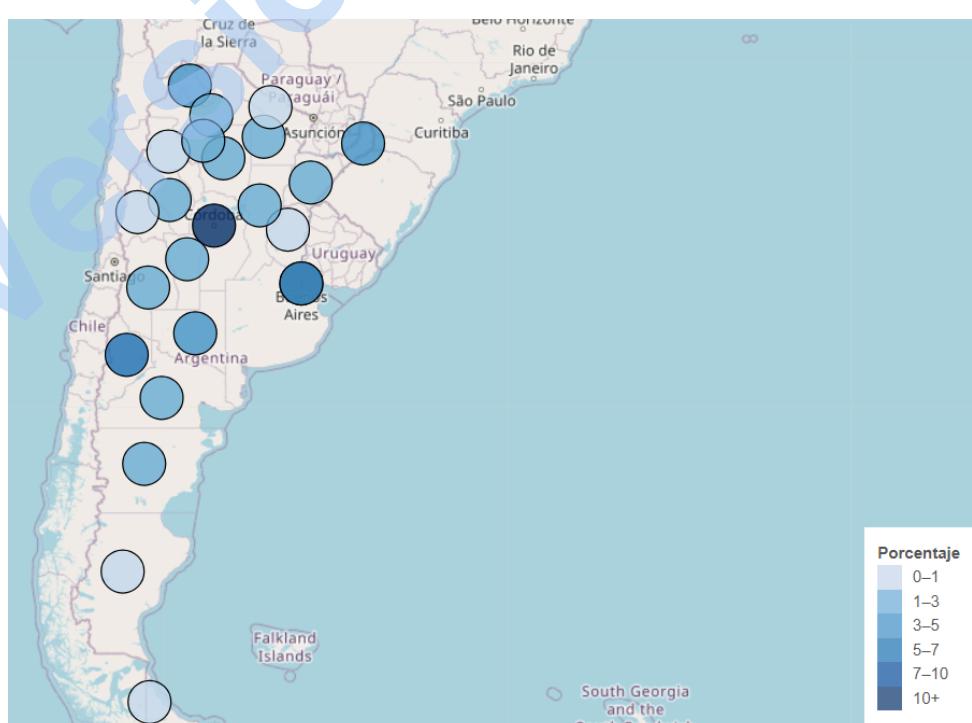


Gráfico 5: distribución geográfica de la presencia de JA

## 5. Conclusiones

Una manera de fortalecer la democracia de cada país, es educando a sus ciudadanos. La educación permite discutir, formar opinión, analizar un candidato y generar voto por líderes responsables que mejoren la vida de las personas. En este proceso, hoy la tecnología tiene incidencia directa en la formación de la ciudadanía, generando mecanismos que tienen impacto sobre las decisiones y opiniones. A través de los datos, las personas pueden analizar situaciones, descubrir hechos, recibir información; los anunciantes pueden focalizar anuncios, segmentar y polarizar audiencias. La brecha de acceso a la tecnología, refuerza la concentración del poder. El acceso equitativo a ella, fortalece la democracia y distribuye el poder en sus ciudadanos.

Si bien los países potencias se encuentran en real ventaja en el desarrollo tecnológico de IA, lo importante en este momento de pleno progreso, es el impulso que cada país le da a la materia. Es decisión política la velocidad con que comenzamos el desarrollo en IA. En el ámbito educativo, lo anterior se traduce en la necesidad de integrar lo tecnológico en los diseños curriculares formales y en la formación docente, de manera de preparar a los estudiantes no solo para el futuro laboral, sino también para participar de manera activa y crítica en la ciudadanía moderna. A pesar de los esfuerzos, se evidencia una necesidad imperante de equiparar en el mapa argentino el acceso a la formación en IA, tanto de alumnos como de sus educadores. Sin una distribución que iguale el acceso a los recursos educativos relacionados a lo tecnológico, las provincias en situaciones desfavorables -y que ya están rezagadas en esta área-, ampliarían rápidamente su brecha de acceso a una educación que prepare a sus estudiantes.

La finalidad de esta investigación, fue evidenciar el nivel de presencia del tema Inteligencia Artificial en los sitios web de los Ministerios de Educación de cada jurisdicción educativa argentina. Teniendo en cuenta que la jurisdicción 'Nación' es la encargada de brindar lineamientos generales a todas las provincias argentinas, decimos que los datos arrojaron un panorama diverso, con marcadas disparidades en el nivel de inclusión y capacitación en IA a lo largo del país. El análisis de la variable 'Inclusión de IA en Diseños Curriculares', destacó que sólo algunas jurisdicciones -como la Ciudad Autónoma de Buenos Aires (CABA), Buenos Aires, y Córdoba-; han logrado integrar en cierto nivel la IA en algunos de sus Diseños Curriculares. A pesar de este esfuerzo, la inclusión es parcial e incompleta y, además, marca una gran diferencia con el resto de jurisdicciones pertenecientes a la muestra, que presentan una adopción nula o mínima de IA en sus currículos.

Por otro lado, no se puede enseñar lo que no se sabe. La formación docente en IA es fundamental en un área de conocimiento incipiente e inédito, y que no estuvo presente en los contenidos de los profesorados. Al respecto, surge de los datos que jurisdicciones como Nación, Córdoba, Misiones y Neuquén; han comenzado a generar propuestas de programas de formación docente en IA. Provincias como CABA, Santa Fe, San Luis, y Chubut; no evidencian en sus portales programas de capacitación en el tema para sus profesores, lo que potencialmente dificultará la implementación del tema en sus Diseños curriculares y enseñanza en los establecimientos educativos.

La única manera de disminuir una brecha naciente en alfabetización en IA en el sistema educativo y sus niveles, es con el desarrollo de políticas educativas nacionales y provinciales firmes y bien direccionadas a promover la integración del área de conocimiento en los Diseños Curriculares. También, dotando de recursos y estrategias equitativas a todas las provincias, de manera de posibilitar el despliegue en todo el sistema, y una integración eficaz de esta tecnología.

## 6. Agradecimientos

A la Dra. Prof. María Cecilia Martínez, por su generosidad, guía y aportes a este artículo.

## Referencias

- [1] E. Brynjolfsson, "The Turing Trap: The Promise & Peril of Human-Like Artificial Intelligence," 2022. [Online]. Available: [https://www.researchgate.net/publication/357791053\\_The\\_Turing\\_Trap\\_The\\_Promise\\_Peril\\_of\\_Human-Like\\_Artificial\\_Intelligence](https://www.researchgate.net/publication/357791053_The_Turing_Trap_The_Promise_Peril_of_Human-Like_Artificial_Intelligence)
- [2] Jan, Zohaib, et al. "Inteligencia artificial para la industria 4.0: revisión sistemática de aplicaciones, desafíos y oportunidades" 2023. [Online]. Available <https://www.sciencedirect.com/science/article/pii/S0957417422024757>
- [3] A. Ng, "How to Build Your Career in AI," S/D [Online]. Available: <https://wordpress.deeplearning.ai/wp-content/uploads/2022/10/eBook-How-to-Build-a-Career-in-AI.pdf>
- [4] EDUCAUSE, "Horizon Report: Higher Education Edition," EDUCAUSE, Louisville, CO, 2019. [Online]. Available: <https://library.educause.edu/media/files/library/2019/4/2019horizonreport.pdf?la=en&hash=C8E8D444AF372E705FA1BF9D4FF0DD4CC6F0FDD1>. Accessed July 2023.

- [5] UNESCO, "Inteligencia artificial y educación: Guía para las personas a cargo de formular políticas," UNESCO, 2021. [Online]. Available: <https://unesdoc.unesco.org/ark:/48223/pf0000379376> . Accessed july 2023
- [6] González Angeletti, V. C. «Análisis Diseños Curriculares de Inteligencia Artificial en Educación Media», TEyET, n.º 37, p. e19, may 2024. [Online]. Available: <https://doi.org/10.24215/18509959.37.e19> Accessed july 2023
- [7] C. Martínez, E. P., M. J. Gómez, M. Borchardt, y M. Garzón, "Hacia un currículum emancipador de las Ciencias de la Computación," Revista Latinoamericana de Economía Y Sociedad Digital, vol. 3, 2022. [Online]. Available: <https://doi.org/10.53857/LBUS5649> Accessed july 2023
- [8] A. M. Chartier, "Historia de la educación," Sociedad Argentina de Historia de la Educación (SAHE) Anuario, 2019.
- [9] F. Fiorucci, "Maestros para el sistema de educación pública. La fundación de escuelas normales en Argentina (1890-1930)," *Revista Mexicana de Historia de la Educación*, vol. II, no. 3, pp. 25-45, 2014.
- [10] A. Puiggrós, *Sujetos, disciplina y currículum en los orígenes del sistema educativo argentino*, cap. 4, Buenos Aires: Galerna, 1990.
- [11] F. Acosta, "La sistematización estatal modelizadora y la segmentación en los orígenes y expansión de la escuela secundaria en la Argentina," Revista del IICE, vol. 47, pp. 23-40, ene.-jun. 2020. [Online]. Available: <http://revistascientificas.filc.uba.ar/index.php/iice/article/view/9637/8417> . Accessed 07-ago-2024. DOI: 10.34096/iice.n47.9637
- [12] Díaz Barriga, Ángel. Currículum. Tensiones conceptuales y prácticas. Revista Electrónica de Investigación Educativa, vol. 5, Nro 2, 2003. [Online]. Available: <https://biblioteca.clacso.edu.ar/Argentina/lpp/20100324015129/3.pdf> . Accessed july 2023
- [13] BERNSTEIN, B. Clase, código y control. Hacia una teoría de las transmisiones educativas, cap. 4 y 5, Madrid: Akal Universitaria, 1988.
- [14] A. Furlán y N. E. Ochoa, "Curriculum, las TIC y Subjetividades," en *Curriculum, subjetividades y nuevas tecnologías*, M. C. Barrón Tirado, Ed., México: IISUE educación, 2023.
- [15] D. Feldfeber y N. Gluz, "La Ley Federal de Educación y su impacto en el sistema educativo argentino," en *Educación y Sociedad en América Latina*, M. Moreno Olivos, Ed., Buenos Aires: Editorial Académica, 2017, pp. 343-356 Available: [https://ri.conicet.gov.ar/bitstream/handle/11336/168804/CONICET\\_Digital\\_Nro.0816f471-2b65-4d8c-a85d-381f2504bd37\\_A.pdf?sequence=2&isAllowed=y](https://ri.conicet.gov.ar/bitstream/handle/11336/168804/CONICET_Digital_Nro.0816f471-2b65-4d8c-a85d-381f2504bd37_A.pdf?sequence=2&isAllowed=y) Accessed july 2023
- [16] Terigi, F., "Sobre la cuestión curricular en la educación secundaria," en *La escolarización de los adolescentes, desafíos culturales, pedagógicos y de política educativa*, E. Tenti Fanfani, Coord., IIPE-UNESCO, Buenos Aires, 2012 [Online]. Available: <https://unesdoc.unesco.org/ark:/48223/pf0000218851> Accessed july 2023
- [17] N. Alterman, "La construcción del currículum escolar. Claves de lectura de diseños y prácticas," Revista Páginas, Escuela de Ciencias de la Educación, FFyH, UNC, vol. 10, no. 6, 2008.
- [18] D. Rodriguez y C. Cortez, "The Position of Computer Science in the Argentina Secondary School," Electronic Journal of SADIO, vol. 19, no. 2, pp. 136-150, 2020. [Online] Available: <https://publicaciones.sadio.org.ar/index.php/EJS/article/view/172/151> . Accessed july 2023
- [19] M. E. Echeveste y M. C. Martínez, "Desafíos en la enseñanza de Ciencias de la Computación," *Virtualidad, Educación y Ciencia*, vol. 7, no. 12, pp. 34-48, 2016. [Online] Available: [https://ri.conicet.gov.ar/bitstream/handle/11336/168804/CONICET\\_Digital\\_Nro.0e6d8ee4-7dca-450a-a069-da37c666625c\\_A.pdf?sequence=1&isAllowed=y](https://ri.conicet.gov.ar/bitstream/handle/11336/168804/CONICET_Digital_Nro.0e6d8ee4-7dca-450a-a069-da37c666625c_A.pdf?sequence=1&isAllowed=y) Accessed july 2023
- [20] M. Cecilia Martínez y M. Emilia Echeveste "Vista de Representaciones de estudiantes de primaria y secundaria sobre las Ciencias de la Computación y su oficio." RED. Revista de Educación a Distancia. Número 46 15-Sep-2015 [Online] Available: <https://revistas.um.es/red/article/view/240241/182961> Accessed july 2023
- [21] Jorge Rodríguez y Marcos Cortez "La posición de las Ciencias de la Computación el Diseño Curricular para la Escuela Secundaria Argentina: Una Revisión Sistemática" EJS 19 (2) 2020 136-150136 [Online] Available: <https://publicaciones.sadio.org.ar/index.php/EJS/article/view/172/151> Accessed july 2023
- [22] Sonia Sommer, María Eda Cornejo, Jorge Rodríguez, Laura Cecchi, "Diseño curricular e innovación en la enseñanza de la computación en la educación secundaria argentina," Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología, vol. 20, pp. 34-45, 2017. [Online] Available: [https://sedici.unlp.edu.ar/bitstream/handle/10915/62265/Documento\\_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y](https://sedici.unlp.edu.ar/bitstream/handle/10915/62265/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y) Accessed july 2023

- [23] M. Southwell, "Posiciones docentes: interrelaciones sobre la escuela y lo justo", INFoD, 2020.
- [24] D. F. Sarmiento, *Facundo: civilización y barbarie*, 1845.
- [25] D. Long y B. Magerko, "What is AI Literacy? Competencies and Design Considerations," en *Conference on Human Factors in Computing Systems*, abril 2020, pp. 1-16. [Online] Available: <https://doi.org/10.1145/3313831.3376727>. Accessed july 2023
- [26] UNESCO, *Qingdao Declaration: Seize Digital Opportunities, Lead Education Transformation*, 2015. [Online] Available: <https://unesdoc.unesco.org/ark:/48223/pf0000233352>. Accessed july 2023
- [27] UNESCO, *K-12 AI Curricula: A Mapping of Government-Endorsed AI Curricula*, 2022. [Online] Available: <https://unesdoc.unesco.org/ark:/48223/pf0000380602>. Accessed july 2023
- [28] UNESCO, *Beijing Consensus on Artificial Intelligence and Education*, 2019. [Online] Available: <https://unesdoc.unesco.org/ark:/48223/pf0000368303>. Accessed july 2023
- [29] S. Sommer *et al.*, "El lugar de las Ciencias de la Computación en el currículum de la Escuela Secundaria Argentina," en *XX Workshop de Investigadores en Ciencias de la Computación*, 2018, p. 470. [Online] Available: [http://sedici.unlp.edu.ar/bitstream/handle/10915/67562/Documento\\_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y](http://sedici.unlp.edu.ar/bitstream/handle/10915/67562/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y). Accessed july 2023
- [30] P. Wang, "On defining artificial intelligence," *Journal of Artificial General Intelligence*, vol. 10, no. 2, pp. 1-37, Jan. 2019, doi: 10.2478/jagi-2019-0002. [Online] Available: <https://intapi.sciendo.com/pdf/10.2478/jagi-2019-0002> Accessed july 2023
- [31] S. Kelly, S. A. Kaye, y O. Oviedo-Trespalacios, "What factors contribute to the acceptance of artificial intelligence? A systematic review," *Telematics and Informatics*, vol. 77, p. 101925, 2023. [Online] Available: [https://www.sciencedirect.com/science/article/pii/S0736585322001587/pdf?trk=public\\_post\\_comment-text](https://www.sciencedirect.com/science/article/pii/S0736585322001587/pdf?trk=public_post_comment-text) Accessed july 2023
- [32] R. D. M. Padilla, "La llegada de la inteligencia artificial a la educación," Dialnet, 2019. [Online] Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=7242777> Accessed july 2023
- [33] M. E. Zepeda Hurtado, E. O. Cardoso Espinosa, y J. A. Cortés Ruiz, "Influencia de la inteligencia artificial en la educación media y superior," *Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, vol. 14, no. 28, p. e679, ene.-jun. 2024. [Online] Available DOI: <https://doi.org/10.23913/ride.v14i28.1949>. Accessed july 2023
- [34] UNESCO, *Beijing Consensus on Artificial Intelligence and Education*, 2019. [Online] Available: <https://unesdoc.unesco.org/ark:/48223/pf0000368303>. Accessed july 2023
- [35] Stanford Institute for Human-Centered Artificial Intelligence, "AI+Education Summit: Advancing Human Learning with AI Technologies," 2023. [Online] Available: <https://hai.stanford.edu/events/aieducation-summit-advancing-human-learning-ai-technologies>. Accessed july 2023
- [36] IE University, "Rethinking education in the age of AI: insights from enlightED 2023," 2023. [Online] Available: <https://www.ie.edu/uncover-ie/rethinking-education-age-ai-insights-enlighted-2023/>. Accessed july 2023
- [37] UNESCO, "K-12 AI curricula: a mapping of government-endorsed AI curricula." [Online]. Available: <https://unesdoc.unesco.org/ark:/48223/pf0000380602> . Accessed: May, 2023
- [38] I. Kimura, C. Queiruga, y J. Díaz, "Iniciativas de la enseñanza de Inteligencia Artificial en la escuela. Enfoques globales, regionales y locales," en Memorias de las 52 JAIIO - SAEI, Córdoba, Argentina, 2024, pp. 171-184. [Online] Available [https://sedici.unlp.edu.ar/handle/10915/165628](http://sedici.unlp.edu.ar/handle/10915/165628) Accessed july 2023
- [39] UNESCO, Guía para Formular Políticas sobre Tecnologías Digitales en la Educación, 2019. [Online] Available: <https://unesdoc.unesco.org/ark:/48223/pf0000368303> Accessed july 2023
- [40] S. Farquhar, J. Kossen, L. Kuhn, y Y. Gal, "Detecting hallucinations in large language models using semantic entropy," *Nature*, vol. 630, pp. 625-630, 2024. [Online] Available: <https://www.nature.com/articles/d41586-024-03637-1> Accessed july 2023
- [41] G. A. Bowen, "Document Analysis as a Qualitative Research Method," *Qualitative Research Journal*, vol. 9, no. 2, pp. 27-40, Aug. 2009, doi: 10.3316/QRJ0902027 [Online] Available: <https://www.researchgate.net/publication/227311387> Accessed july 2023

[42] M. B. Miles, A. M. Huberman, y J. Saldaña, Qualitative Data Analysis: A Methods Sourcebook, 3rd ed. Thousand Oaks, CA: Sage Publications, 2013. [Online] Available: <https://vivauniversity.wordpress.com/wp-content/uploads/2013/11/milesandhuberman1994.pdf> Accessed july 2023

Versión Preliminar

# **Explorando la enseñanza de conceptos de programación utilizando diferentes lenguajes de programación**

José María Massa<sup>1</sup>, María Carmen Leonardi<sup>1</sup>, María Virginia Mauco<sup>1</sup>

<sup>1</sup> INTIA. Facultad de Ciencias Exactas. Universidad Nacional del Centro de la Provincia de Buenos Aires, Tandil, Argentina

{jmassa, cleonard, virginiamauco}@intia.exa.unicen.edu.ar

## **Resumen**

La elección de qué lenguajes utilizar para enseñar los conceptos básicos de la programación de computadoras en cursos introductorios de carreras de Ciencias de la Computación, es un tema de gran interés en la comunidad educativa. En cuanto a los contenidos, se ha considerado un conjunto mínimo común a partir de un relevamiento realizado en estos cursos introductorios en universidades de Argentina. En este artículo, la intención es explorar diferentes escenarios de enseñanza para algunos de esos contenidos de programación y proponer algunas guías acerca de cómo estos contenidos podrían ser desarrollados según el lenguaje que se haya seleccionado. Para la elección de los temas y los lenguajes utilizados en este trabajo, se consideró que, para cada tema, haya dos familias de lenguajes que difieran significativamente en la característica del lenguaje que se quería explorar. Finalmente, se presenta una discusión inicial acerca de cómo estas características influirían en los aspectos didácticos de los cursos.

**Palabras clave:** Materias de programación; Carreras de Ciencias de la Computación; Lenguajes de programación

## **1. Introducción**

La elección de los lenguajes de programación en cursos de programación de los primeros años de las carreras de grado del campo de la informática, se ha convertido en un tópico de gran interés en los últimos años. Una de las motivaciones de este interés es la cantidad y variedad de lenguajes que han surgido y las preguntas que se plantean los docentes acerca de la incorporación de estos lenguajes a la enseñanza [1, 2].

En [3] se definió un conjunto de contenidos mínimos de una materia introductoria de programación en las carreras de Ciencias de la Computación de Argentina, así como también se presentó un relevamiento de los lenguajes utilizados en las mismas, junto con un análisis de sus características. En este artículo, la intención es explorar diferentes escenarios de enseñanza para algunos de esos contenidos de programación y proponer algunas guías acerca de cómo estos contenidos podrían ser desarrollados según el lenguaje que se haya seleccionado. Se supone que el proceso de selección del lenguaje ya se ha realizado y por lo tanto el mismo queda fuera del alcance de este artículo. Los lenguajes de programación utilizados en este artículo para resolver las consignas provienen, en su mayoría, del mismo relevamiento.

La elección de los temas y de los lenguajes no fue al azar, sino que se seleccionaron en conjunto considerando que, para cada tema, haya dos familias de lenguajes que difieran significativamente en la característica del lenguaje que se quiere analizar. Esta consideración se basó en la suposición de que, en general, se pueden agrupar los lenguajes en dos conjuntos: por un lado, aquellos lenguajes en donde el programador debe explicitar manualmente las construcciones (lenguajes de bajo/medio nivel de abstracción) y, por otro lado, los lenguajes que automatizan implícitamente las construcciones (lenguajes de alto nivel de abstracción). Por ejemplo, si la característica ligadura de tipo influye en una consigna donde el tipo de datos es un elemento central para su resolución, se seleccionaron dos lenguajes, uno de los cuales tiene ligadura estática de tipos y el otro, ligadura dinámica.

Para el tratamiento de los temas y lenguajes, se eligieron cuatro temas y para cada uno de ellos se seleccionaron dos lenguajes de programación. Se propuso una consigna para cada tema, analizando los elementos que se ponen en juego en

cada consigna y se desarrollaron dos soluciones, una en cada lenguaje. Para cada una de estas resoluciones se analizaron las cuestiones que surgieron por haber utilizado el lenguaje seleccionado y un posible abordaje por parte del docente.

El artículo contiene una sección en la que se presentan algunos trabajos relacionados a la temática. Luego, la Sección 3 en la que se desarrollan los cuatro ejemplos de escenarios de enseñanza de los temas seleccionados, y finalmente una sección en la que se presentan algunas conclusiones y futuros trabajos.

## 2. Trabajos relacionados

La elección del lenguaje de programación a usar en cursos introductorios de programación en distintos niveles de educación es un tema que se viene abordando desde hace varios años por la comunidad de educación en computación. En particular, existen trabajos dedicados especialmente al nivel universitario, en los que se presentan y analizan distintos criterios a tener en cuenta para elegir un lenguaje de programación y un entorno de trabajo asociado. Algunos de ellos, se describen brevemente a continuación.

Existen trabajos que involucran estudios de largo plazo mostrando la evolución de los diferentes lenguajes de programación utilizados en los cursos mencionados y la influencia de los paradigmas de estos lenguajes en los enfoques de dichas materias [1, 2]. En estos artículos se analizan no sólo las características de los lenguajes utilizados sino la influencia de los enfoques imperativo, funcional, estructurado y orientado a objetos en la enseñanza de la programación.

En [4] se plantea la necesidad de decidir qué lenguaje de programación y entorno de trabajo usar considerando de una manera criteriosa las características de los lenguajes/entornos ya que éstas tienen una gran influencia en la didáctica de los contenidos a enseñar. Para eso, se proponen distintas dimensiones para analizar distintos entornos de enseñanza de programación (lenguaje de programación general o específico, lenguaje textual o entorno por bloques, entre otras). En este artículo se retoma la paradoja del lenguaje planteada en [5]: el lenguaje y el entorno a utilizar no son importantes, pero sin embargo son importantes. Esta paradoja establece que como lo importante son los conceptos a enseñar, el lenguaje o entorno que se elige no serían importantes. Pero como el lenguaje, a través del entorno, es el que permite expresar esos conceptos se hace necesario tener en cuenta de qué manera influye en la forma de transmitir esos conceptos.

En [6] se hace referencia a este tema como el dilema de la elección del primer lenguaje, y se define un conjunto preliminar de criterios educativos a tener en cuenta a la hora de elegir un lenguaje (tales como sintaxis simple, soporte de verificación para promover la escritura de código correcto, conceptos claros para ciclos y subrutinas), analizando seis lenguajes en base a estos criterios.

En [7] se presenta un análisis comparativo de dos investigaciones realizadas sobre cursos introductorios de programación en universidades de Reino Unido y Australia y Nueva Zelanda, en las que se abordaron los lenguajes de programación utilizados, los entornos de desarrollo y las razones de estas elecciones. Estos autores concluyen que “la guerra de los lenguajes”, como se la llamó en [8] para referirse a la gran divergencia e impacto de los distintos lenguajes de programación, difícilmente llegue a una solución definitiva y en consecuencia los departamentos de informática deberán considerar periódicamente qué lenguaje de programación utilizar en sus cursos introductorios.

En [9] se presenta un estudio de la evolución, a través de los años, de la elección de un primer lenguaje de programación, mostrando cuáles son los lenguajes más usados en el “mundo real”, así como también qué cuestiones se pueden tener en cuenta para hacer esa elección. En base a este estudio, la autora reflexiona que la elección del lenguaje de programación para la enseñanza introductoria debe acompañar la evolución, pero por ser el lenguaje que introduce a los estudiantes en la programación, la elección debe tener en cuenta tanto aspectos pedagógicos como de aceptación del mundo exterior.

Estos artículos coinciden en que la elección de un lenguaje de programación para un curso introductorio 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 [3, 10].

### 3. Ejemplos de Escenarios de Enseñanza

En esta sección se presentan un subconjunto de los temas del conjunto de contenidos mínimos de [3]. Para cada uno de ellos se plantea una consigna y se analizan dos posibles resoluciones<sup>1</sup> de las mismas con dos lenguajes de programación que presenten características bien diferenciadas en relación a los aspectos de los lenguajes que se ponen en juego para resolver la consigna.

#### Escenario 1: Enseñanza de tipos de datos

Consigna: Implementar una entidad Producto para representar un producto de un determinado negocio. Debe tener un código numérico, una descripción alfanumérica y un precio con decimales. Escribir un programa que dé de alta dos productos con valores y los muestre por pantalla.

Solución 1 (realizada en <a href="#">Java</a> )	<pre>class Producto {     private int codigo;     private String descripcion;     private double precio;      public int get_codigo() {         return codigo;     }      public String get_descripcion() {         return descripcion;     }      public double get_precio() {         return precio;     }      public Producto(int codigo, String descripcion, double precio) {         this.codigo = codigo;         this.descripcion = descripcion;         this.precio = precio;     }      public class EjemploModularizacion {         public static void main(String[] args) {             Producto producto1 = new Producto(101, "Manzana", 325.99);             System.out.println(producto1.get_codigo());             System.out.println(producto1.get_descripcion());             System.out.println(producto1.get_precio());         }     } }</pre>
Solución 2 (realizada en <a href="#">Python</a> )	<pre>class Producto:     def __init__(self, codigo, descripcion, precio):         self.codigo = codigo         self.descripcion = descripcion         self.precio = precio      def main():         producto1 = Producto(codigo=101, descripcion="Manzana", precio=325.99)         print (producto1.codigo)         print (producto1.descripcion)         print (producto1.precio)          producto2 = Producto(codigo="Banana", descripcion=102, precio=412.79)         print (producto2.codigo)         print (producto2.descripcion)         print (producto2.precio)      main()</pre>

<sup>1</sup> Cada una de estas soluciones está disponible a través del enlace a la plataforma <https://onecompiler.com/>

Este ejercicio se ubica en el contexto de enseñanza de estructurar variables de tipos heterogéneos en variables de un solo tipo para representar entidades de un problema. Si bien es esperable que el problema de la elección de los tipos de datos adecuados ya se haya planteado en la enseñanza de variables simples, se ha preferido unificar estos temas en un solo ejemplo. Se seleccionaron dos lenguajes que se comportan de forma opuesta para la característica “tipo” (uno es dinámico y el otro es estático).

Solución 1: El lenguaje seleccionado posee ligadura estática de tipos de datos y clases como mecanismo para estructurar datos de tipos heterogéneos. Teniendo en cuenta esto, introducir el tema implica que el docente hable del concepto de tipos como moldes que definen un dominio de valores y un conjunto de operaciones válidas para las variables de ese tipo. En este sentido, aprovechando que este lenguaje ofrece un menú de familias de tipos, el docente puede aprovechar para trabajar sobre la elección del tipo más adecuado para el dato que se quiera representar. También puede separar explícitamente la declaración de la variable con determinado tipo, de su primer uso, diferenciando claramente estos dos conceptos. Con este tipo de lenguajes, la verificación de compatibilidad de tipos se realiza en tiempo de compilación, por lo tanto, si por ejemplo el estudiante escribe la sentencia: “Producto producto2 = new Producto(“Banana”, 102, 325.99);” el compilador arrojará un error de incompatibilidad de tipos.

Solución 2: El lenguaje seleccionado utiliza clases como mecanismo para estructurar datos al igual que el lenguaje de la solución anterior y ligadura dinámica de tipos de datos, lo cual implica que la determinación del tipo de dato de las variables se realiza durante la ejecución y las variables se crean con su primer uso. Esto implica que el programador no debe realizar las declaraciones de variables estableciendo sus tipos, sino que esto lo realiza automáticamente el lenguaje. Aquí se plantean las siguientes cuestiones: ¿Qué significa que no sea necesario declarar explícitamente las variables? ¿Qué significa que no haya que escribir los tipos de datos de cada variable?, entre otras. En este punto el docente se encuentra frente a una situación de simplificación de la solución y puede plantearse diferentes enfoques. Uno de ellos consiste en no profundizar sobre los aspectos subyacentes y aprovechar la simpleza del lenguaje para abarcar otros temas de interés en la programación. Otro enfoque posible podría ser detenerse y considerar los mecanismos que realiza el lenguaje de manera subyacente. Así el estudiante, si bien no tiene control directo, puede tomar conciencia de lo que está ocurriendo en cada instrucción que escribe. En este caso, sería entender que el programa en Python posee declaraciones de variables y estas variables poseen sus correspondientes tipos, aun cuando estas cuestiones no sean visibles. Por otro lado, esta explicación no debe escaparse del hecho de situarnos en que estamos enseñando la lógica de la programación y cómo resolver problemas a través de programas, sin profundizar en los lenguajes en sí mismos como objetos de estudio. Continuando con el ejemplo, si al igual que en el caso de la Solución 1, un estudiante introduce las líneas que se observan al final del código, no sólo no se producirá ningún error en tiempo de escritura del programa, sino que, como consecuencia de la ligadura dinámica de tipos de Python, las variables “código” y “descripción” cambiarán de tipo, con lo cual cambiará el comportamiento del programa. Si posteriormente se quisiese realizar operaciones sobre estas variables, seguramente aparecerán, además, errores de incompatibilidad de tipos en tiempo de ejecución.

#### Escenario 2: Enseñanza de resolución de ámbito

Consigna: Dada una función que tiene una variable costo, escribir funciones que calculen, por ejemplo, los impuestos iva e ingresos brutos con las alícuotas del 21% y 3% respectivamente y los agreguen a la variable costo.

Solución 1 (realizada en <a href="#">Python</a> )	<pre>def agregar_impuestos():     costo = 100     def agregar_iva():         iva = 0.21         nonlocal costo         costo = costo + costo*iva     def agregar_ingresos_brutos():         iibb = 0.03         nonlocal costo         costo = costo + costo*iibb         print(costo)     agregar_ingresos_brutos()     print(costo)     agregar_iva()     print(costo) agregar_impuestos()</pre>
Solución 2 (realizada en <a href="#">Perl</a> )	<pre>sub agregar_impuestos() {     local \$costo = 100;     print(\$costo);     agregar_iva();     print(\$costo);</pre>

```

agregar_ingeros_brutos();
print($costo);
}

sub agregar_iva() {
    $iva = 0.21;
    $costo = $costo + $costo*$iva;
}

sub agregar_ingeros_brutos() {
    $iibb = 0.03;
    $costo = $costo + $costo*$iibb;
}
agregar_impuestos();

```

Este ejercicio se ubica en el contexto de enseñanza donde se quiere explicar cómo intercambian información las unidades de un programa. De esta manera, el enfoque pedagógico se debe adecuar al mecanismo de resolución de ámbito que provea el lenguaje en cuestión. Por un lado, se tienen los lenguajes con resolución de ámbito profundo que generalmente responden al paradigma imperativo y por otro lado los lenguajes con resolución de ámbito superficial que generalmente se corresponden al paradigma funcional. Si bien en el caso de estos últimos, no se han encontrado en el relevamiento realizado en [3], resulta interesante su análisis ya que hoy en día existe un resurgimiento de los aspectos funcionales que se manifiesta no solamente por la aparición de nuevos lenguajes de este tipo, sino también en la incorporación de algunas características de este paradigma en los lenguajes imperativos [11]. Por eso, se seleccionaron dos lenguajes que se comportan de forma opuesta para la característica “ámbito” (uno es dinámico y el otro es estático).

Solución 1: Si el lenguaje seleccionado tiene resolución de ámbito profundo, como es el caso de Python, el docente explica que las variables no locales de una unidad se buscan en la unidad que la anida sintácticamente y así sucesivamente. Este concepto es completamente visible en el mismo código, por lo que, por lo general, no requiere emplear mecanismos didácticos adicionales para la explicación. En esta solución, la estructura de la solución es simple de describir ya que las características de estructuración de unidades del lenguaje se ajustan bien a la consigna. De esta manera resulta natural que la función “agregar\_impuestos” posee la variable “costo” y que las funciones “agregar\_iva” y “agregar\_ingeros\_brutos” accederán a dicha variable definida como no local a estas funciones.

Solución 2: Si el lenguaje seleccionado tiene resolución de ámbito superficial, como es el caso de Perl, las variables no locales a una función no necesariamente se encuentran definidas en las unidades que las anidan, sino que el lenguaje tiene un mecanismo de búsqueda según el orden de ejecución. De esta manera, la función “agregar\_iva” puede utilizar la variable “\$costo” porque esta última fue definida en la función “agregar\_impuestos” que es la invocadora de “agregar\_iva”. Este mecanismo, si bien es trivial, no está visible a primera vista para el programador, con lo cual didácticamente sería deseable acompañar esta explicación con un diagrama o notación donde pueda visualizarse el flujo de ejecución del programa.

### Escenario 3: Enseñanza de memoria dinámica

Consigna: Realizar un programa para devolver la posición del producto que tiene el máximo costo dentro de una lista de productos

Solución 1  
(realizada en [C++](#))

```

#include <iostream>
#include <vector>
#include <string>

class Producto {
public:
    std::string Descripcion;
    long Código;
    double Costo;

    // Constructor
    Producto(std::string descripcion, long código, double costo) {
        Descripcion = descripcion;
        Código = código;
        Costo = costo;
    }
};

```

	<pre> int main() {     std::vector&lt;Producto*&gt; productos;      // Agregar productos a la lista utilizando memoria dinámica     productos.push_back(new Producto("Manzana", 101, 325.99));     productos.push_back(new Producto("Banana", 102, 412.79));     productos.push_back(new Producto("Pera", 103, 389.40));     productos.push_back(new Producto("Naranja", 104, 254.99));      int maxPos = 0;     int posicion;     double maxCosto = productos[0]-&gt;Costo;      for (posicion = 0; posicion &lt; productos.size(); posicion++) {         if (productos[posicion]-&gt;Costo &gt; maxCosto) {             maxPos = posicion;             maxCosto = productos[posicion]-&gt;Costo;         }         delete productos[posicion];     }     std::cout &lt;&lt; "Posicion donde se encuentra el máximo: \$" &lt;&lt; maxPos &lt;&lt;     std::endl;      return 0; } </pre>
Solución 2 (realizada en <a href="#">C#</a> )	<pre> using System; using System.Collections.Generic;  // Definición de la clase Producto public class Producto {     public string Descripcion { get; set; }     public long Codigo { get; set; }     public double Costo { get; set; }      // Constructor     public Producto(string descripcion, long codigo, double costo)     {         Descripcion = descripcion;         Codigo = codigo;         Costo = costo;     } }  public class Program {     public static void Main()     {         // Crear una lista para almacenar objetos Producto         List&lt;Producto&gt; productos = new List&lt;Producto&gt;();          // Agregar productos a la lista         productos.Add(new Producto("Manzana", 101, 325.99));         productos.Add(new Producto("Banana", 102, 412.79));         productos.Add(new Producto("Pera", 103, 389.40));         productos.Add(new Producto("Naranja", 104, 254.99));          int maxPos = 0;         int posicion = 0;         double maxCosto = productos[0].Costo;          while (posicion &lt; productos.Count-1) {             if (productos[posicion].Costo &gt; maxCosto) {                 maxPos = posicion;                 maxCosto = productos[posicion].Costo;             }             posicion++;         }     } } </pre>

```

        }
        posicion++;
    }
    Console.WriteLine(maxPos);
}

```

Este ejercicio se encuentra en el contexto de creación de variables en la memoria libre, en donde los estudiantes deben crear una lista que permita almacenar una cantidad arbitraria de elementos de tipo “Producto”, solicitando memoria al lenguaje para almacenar cada uno, luego implementar un recorrido de esta lista para devolver la posición del producto con el costo máximo y finalmente liberar la memoria solicitada. En este caso, se seleccionaron dos lenguajes que se comportan de forma opuesta para la característica “gestión de memoria” (uno es automático y el otro es manual).

Solución 1: En el código de ejemplo de esta solución, escrito en el lenguaje C++, se tiene una clase “Producto” con los atributos “codigo”, “descripción” y “costo”, luego se declara una variable de tipo Vector de productos llamada “productos” y se crean 4 variables de tipo “Producto” en memoria libre y se agregan a productos. Luego, se recorren los elementos del vector “productos” para obtener el máximo, liberando la memoria de cada producto ya utilizado y finalmente termina el programa luego de imprimir la posición del elemento máximo. En este punto el docente explica cómo la instrucción new solicita memoria libre al lenguaje, el mismo devuelve una dirección de memoria donde se encuentra el bloque solicitado (puntero) y esta dirección de memoria se almacena en la posición correspondiente del vector. Luego explica cómo se accede a la información de cada producto mediante el puntero a través de una notación especial ( $\rightarrow$ ). Finalmente se explica la liberación de memoria cuando ya no se necesita a través del borrado explicado de cada uno de los productos apuntados por las direcciones.

Solución 2: En el código de ejemplo de esta solución, escrito en el lenguaje C#, se tiene una clase “Producto” con los atributos “codigo”, “descripción” y “costo”, luego se declara una variable de tipo Lista de Productos llamada “productos” y se crean 4 variables de tipo “Producto” en memoria libre y se agregan a “productos”. Luego, se recorren los elementos del vector “productos” para obtener el máximo y finalmente termina el programa luego de imprimir la posición del elemento máximo. En este punto el docente explica cómo la instrucción new solicita memoria libre al lenguaje, el mismo devuelve una dirección de memoria donde se encuentra el bloque solicitado y esta dirección de memoria se almacena en la posición correspondiente del vector. Sin embargo, el manejo de esta dirección es implícito y por lo tanto no se visualiza en el programa. En este lenguaje, por la característica que posee de administrar la liberación automática de memoria a través de un algoritmo recolector de memoria libre del lenguaje, no es posible que el programador solicite explícitamente la liberación de memoria. El hecho de no poseer un operador explícito de memoria ni un mecanismo de liberación hace que el docente pueda adoptar un enfoque de no profundizar en esto y por otro lado, puede aprovechar este automatismo para explicar que el lenguaje implícitamente está trabajando con direcciones de memoria por más que lo oculte al usuario y además, profundizar en el mecanismo de recolección de memoria libre en los aspectos de manipulación del mismo que están al alcance de un programador de nivel inicial.

#### Escenario 4: Enseñanza de iteraciones

Consigna: Calcular el promedio de los costos de una lista de productos

Solución 1  
(realizada en [Pascal](#))

```

program CalcularPromedioCostos;

type
    TProducto = record
        codigo: Integer;
        costo: Real;
        descripcion: String;
    end;

var
    productos: array of TProducto;
    sumaCostos, promedio: Real;
    i, numProductos: Integer;

begin
    numProductos := 3;
    SetLength(productos, numProductos);

    productos[0].codigo := 101;

```

	<pre> productos[0].costo := 250.5; productos[0].descripcion := 'Manzana';  productos[1].codigo := 102; productos[1].costo := 405.0; productos[1].descripcion := 'Banana';  productos[2].codigo := 103; productos[2].costo := 353.0; productos[2].descripcion := 'Pera';  sumaCostos := 0.0; for i := 0 to numProductos - 1 do begin     sumaCostos := sumaCostos + productos[i].costo; end; promedio := sumaCostos / numProductos;  Writeln('El promedio de los costos es: ', promedio:0:2); end.</pre>
Solución 2 (realizada en <a href="#">Python</a> )	<pre> class Producto:     def __init__(self, descripcion, codigo, costo):         self.descripcion = descripcion         self.codigo = codigo         self.costo = costo      def __repr__(self):         return f"Producto(descripcion={self.descripcion}, codigo={self.codigo}, costo={self.costo})"  # Crear una lista de productos productos = [     Producto("Manzana", 101, 325.99),     Producto("Banana", 102, 412.79),     Producto("Pera", 103, 389.40),     Producto("Naranja", 104, 254.99) ] total_costos = mean(producto["costo"] for producto in productos) promedio = total_costos / len(productos)</pre>

Este ejercicio está en el contexto de enseñanza de iteraciones sobre colecciones de datos. Para resolverlo, se seleccionaron dos lenguajes que se comportan de forma opuesta para esta característica: el primero de ellos requiere que el programador ponga en juego varias habilidades para implementar todo lo necesario para realizar la iteración, como se verá en la solución 1. El otro lenguaje, soporta la construcción de iteradores automáticos, que no requieren explicitar más que la estructura sobre la cual se va a iterar y una variable que va a contener los elementos sucesivos de la iteración, liberando al programador del resto de las acciones necesarias para recorrer la colección.

Solución 1: En esta solución, realizada en Pascal, el docente debe explícitamente hablar de la necesidad de tener que definir una estructura de control como por ejemplo “for” para realizar la iteración sobre la colección, definir un índice con un tipo de datos entero, inicializarlo y definir valores de comienzo y fin. En este enfoque, el docente puede trabajar sobre la enseñanza de cada uno de los aspectos que intervienen para que una iteración se pueda realizar correctamente.

Solución 2: Esta solución se presenta codificada en Python, que cuenta con una construcción para iteraciones automáticas. Si bien este lenguaje también permite realizar iteraciones de forma manual, se prefiere analizar este enfoque como

contrapartida conceptual con el de la Solución 1. En este caso, la solución sería más sencilla ya que el propio iterador se encarga de automatizar no solo la definición del inicio y el fin de la iteración, sino el índice y su incremento. En este punto se plantean dos alternativas para el docente, una de ellas es omitir el detalle de lo que hace el lenguaje para realizar la iteración, aprovechando que estas características automáticas eliminan los errores de programación que se deben a definiciones incorrectas del índice, inicio y fin, mal manejo del incremento del índice entre otros. El otro enfoque consiste en profundizar en lo que hace el lenguaje automáticamente para hacer funcionar la iteración, lo cual no excede lo que debería hacer el programador en un lenguaje sin iteradores. Lo beneficioso de este segundo enfoque es que cuando los estudiantes se encuentren con un lenguaje sin iteradores automáticos, tendrán los elementos para hacerlo.

## 4. Conclusiones

En este trabajo se han presentado algunos ejemplos de problemas sencillos sobre contenidos de materias de programación de los primeros años de las carreras de Ciencias de la Computación de universidades de Argentina, y para cada uno de esos problemas, se han propuesto dos soluciones. Considerando características de los lenguajes de programación relacionadas a cada una de las consignas, se seleccionó un lenguaje en el cual el uso de esa característica requiere un trabajo explícito por parte del programador y otro lenguaje en el cual esa característica automatiza de forma implícita gran parte de ese uso. Si bien para cada consigna es posible pensar en un abanico grande de soluciones, las presentadas en los ejemplos se confeccionaron de modo de hacer visible el uso explícito e implícito de las características de los lenguajes. Los lenguajes utilizados en los ejemplos fueron seleccionados del conjunto del relevamiento realizado en [3], excepto en uno de los casos.

Es evidente que, si se considera la implementación de un programa que requiere un trabajo más exhaustivo por parte del estudiante debido a que las construcciones que se utilizan no son de un alto nivel de abstracción, el docente deberá estar más atento a los detalles del código fuente, ya que finalmente un programa que está bien planteado desde la lógica puede fallar por errores de programación en el uso de las construcciones de niveles bajos/medios de abstracción. En el caso de los lenguajes que ofrecen construcciones de niveles más altos de abstracción, como se mencionó anteriormente, por lo general éstos ocultan algunos detalles de la implementación, automatizándolos implícitamente. Esto hace que un código que están bien planteado desde la lógica tenga menos posibilidades de falla y requiera menos tiempo del docente para atender a los detalles del uso de las construcciones. Teniendo en cuenta esto y considerando el análisis de los ejemplos presentados en la Sección 3, se puede ver que los lenguajes de niveles más altos de abstracción ponen a disposición del docente un tiempo adicional.

El qué hacer con este tiempo dependerá de los objetivos a cumplir en estos cursos de programación en relación a la carrera en la cual están insertos, ya que puede haber un escenario donde lo que se busca es generar una mayor cantidad de programadores que puedan responder a las demandas actuales de trabajo. En este escenario, si se busca generar programadores de alto nivel de abstracción, el tiempo puede destinarse a incorporar más temas de programación en la materia. En otro escenario, posiblemente relacionado con carreras más largas, el objetivo de las materias de programación puede consistir en formar un pensamiento computacional para la programación, abarcando un mayor espectro de niveles de abstracción. En este caso, si se utilizó un lenguaje de alto nivel, el docente puede profundizar en los aspectos subyacentes de lo que realizan automáticamente estos lenguajes. Es importante en este punto aclarar que la profundización consistiría en el “qué está haciendo el lenguaje automáticamente” y no en el “cómo lo está haciendo detalladamente”, ya que esto último es un tópico del área de Lenguajes de Programación.

En consecuencia, sería beneficioso que el docente tenga información disponible para poder aprovechar al máximo el lenguaje que por alguna razón se haya decidido utilizar. Es por ello que este trabajo es un puntapié inicial para avanzar en la confección de una guía de los aspectos a atender considerando los temas que se dictan en los primeros cursos de programación y diferentes familias de lenguajes.

El lenguaje de programación inicial, primer paso de un largo recorrido, es aquel que permitirá al estudiante comenzar a adquirir y materializar los primeros conceptos de programación. Este lenguaje debería ser el medio para que el estudiante se introduzca en el pensamiento computacional y se “enamore” de la programación. En este sentido la pregunta de qué lenguaje de programación se utilizará en esta etapa introductoria, es una pregunta que estará siempre en vigencia debido a la permanente evolución de los lenguajes de programación y a la necesidad de lograr un balance entre las demandas de la industria y los aspectos didácticos de la enseñanza de la programación. Y más allá del lenguaje que se elija, lo importante es cómo el docente debe adaptar su enfoque didáctico al lenguaje en cuestión.

## Referencias

- [1] S. Sobral, “30 Years of CS1: Programming languages evolution”, Proceedings 12th Annual International Conference of Education, Research and Innovation ICERI 2019, pp. 9197-9205). IATED.
- [2] J. Swacha, “Programming Languages in Education: 50 Years of Evolution as Evidenced by Literature”, Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2, 2022, pp. 1385.

- [3] J. M. Massa, M. C. Leonardi, M. V. Mauco, L. Felice, J. Doorn, “Hacia un Análisis de Contenidos y Lenguajes de Programación en Materias Iniciales de Programación de Carreras de Ciencias de la Computación” Jornadas Argentinas de Didáctica de las Ciencias de la Computación JADiCC 2023.
- [4] M. J. Gómez, P. E. “Fidel” Martínez López, “¿Scratch, Python, o qué?”, Jornadas Argentinas de Didáctica de las Ciencias de la Computación JADiCC 2021, pp 101-114.
- [5] P. E. “Fidel” Martínez López, E. A. Bonelli, F. A. Sawady O’Connor, “El nombre verdadero de la programación: Una concepción de enseñanza de la programación para la sociedad de la información”, 10º Simposio sobre la Sociedad de la Información, JAIIo 2012.
- [6] M. Foughali, “Some thoughts on teaching introductory programming and the first language dilemma”, 23rd Koli Calling International Conference on Computing Education Research, 2023, Finland, pp 1-8.
- [7] Simon, R. Mason, T. Crick, J. Davenport, E. Murphy, “Language Choice in Introductory Programming Courses at Australasian and UK Universities”, Symposium on Computer Science Education, 2018, pp 852-857.
- [8] A. Stefik, S. Hanenberg “The Programming Language Wars. Questions and Responsibilities for the Programming Language Community”, Proceedings of the 2014 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming & Software, 2014, pp 283-299.
- [9] S. Sobral, “The old question: which programming language should we choose to teach to program?”, International Conference on Advances in Digital Science, 2021, Springer pp 351–364.
- [10] K. R. Parker, T. A. Ottaway, J. T. Chao, “Criteria for the selection of a programming language for introductory courses”, International Journal of Knowledge and Learning, vol 2, no 1-2, pp 119-139, 2006.
- [11] J. Doorn, J. M. Massa, *Closures: semántica y pragmática*. Editorial UNICEN, 2024.

# Recomendaciones acerca de los escenarios al momento de introducir en clases algoritmos de ordenamiento de secuencias de elementos

Tobías Valdés Castro<sup>1</sup>, Pablo Turjanski<sup>1,2</sup>, and Matías Lopez-Rosenfeld<sup>1,2</sup>

<sup>1</sup>Universidad de Buenos Aires. Facultad de Ciencias Exactas y Naturales. Departamento de Computación. Buenos Aires, Argentina.

<sup>2</sup>CONICET-Universidad de Buenos Aires. Instituto de Ciencias de la Computación (ICC). Buenos Aires, Argentina.

tvaldes@dc.uba.ar, pturjanski@dc.uba.ar, mlopez@dc.uba.ar

## Resumen

Entre los temas iniciales que se suelen incluir en la currícula de las carreras de Ciencias de la Computación se encuentra el ordenamiento. Una de las estrategias didácticas para abordarlo es la problematización del tema con diferentes escenarios, como pueden ser las secuencias de números, naipes, etc. Desarrollamos un videojuego para poder estudiar el comportamiento de estudiantes frente a distintos escenarios, y de esta manera analizar ventajas y desventajas de cada propuesta. En la caracterización de escenarios, se destaca que en algunos de ellos la información disponible y operaciones mínimas necesarias para resolver no son equivalentes a las usadas por los algoritmos que se presentan en los cursos, parte de la tarea es computada sin la necesidad de realizar explícitamente una de sus operaciones. Finalmente, se concluye que se debe prestar atención al momento de seleccionar un escenario, de manera tal que las operaciones e información disponibles sean similares a las que tienen y utilizan luego los algoritmos que se enseñarán al resolverlo. Esto es, que incluya los comandos básicos a utilizar por el algoritmo a introducir en clase.

**Palabras clave:** Ordenamiento; Juegos; Didáctica; Algoritmos.

## 1. Introducción

En el contexto de la enseñanza de las Ciencias de la Computación, los métodos de ordenamiento de secuencias de elementos suelen estar entre los temas incluidos en el programa de las materias relacionadas con algoritmos y estructuras de datos [1].

Existen diversos algoritmos de ordenamiento. Una de las formas de presentar el problema es el de ordenar de manera creciente, de izquierda a derecha, una secuencia  $A$  que posee  $n$  elementos ( $A[0], A[1], \dots, A[n-1]$ ). Un ejemplo es el algoritmo *Selection Sort* (en el Código 1 se incluye un pseudocódigo del mismo) cuya estrategia de resolución del problema consiste en repetir la acción de buscar el menor elemento de la secuencia y ubicarlo al principio de la misma. Luego se repite el procedimiento con los elementos aún no ordenados, y así sucesivamente.

```
1 selectionSort(A[0..n-1])
2     for i = 0:n-2
3         # Buscar el menor elemento (desde la posición i hasta n-1)
4         i_min = i # primer candidato a mínimo
5         for j = i+1:n-1
6             if A[j] < A[i_min]
7                 i_min = j # nuevo candidato mínimo encontrado
8             # Ubicar en la posición i al menor elemento hallado
9             swap(A[i],A[i_min])
10            # Retornar la secuencia de elementos ordenada
11        return A[0..n-1]
```

Código 1: Pseudocódigo correspondiente al algoritmo *Selection Sort*. Las secciones del código asociadas a los comandos **COMPARE** (línea 6) **SWAP** (línea 9) y **SUBMIT** (línea 11) se encuentran resaltadas en color amarillo.

*Selection Sort* no es el único algoritmo de ordenamiento que se suele introducir en estos cursos. Otros algoritmos que se suelen incorporar a los cursos son *Insertion Sort* y *Bubble Sort*. En el caso del *Insertion Sort* (ver Código

2), la estrategia consisten en partir la secuencia de elementos en una parte ordenada (izquierda, inicialmente sin elementos) y otra desordenada (derecha). En cada paso se toma un elemento de la parte desordenada y se lo inserta en la posición correcta de la parte ordenada. Este proceso se repite hasta que todos los elementos se encuentran ordenados.

```

1 insertionSort(A[0..n-1])
2   for i = 1:n-1
3     # Hacer espacio para ubicar en la posicion correcta, al elemento A[i],
4     # dentro de la parte ordenada (A[0], ..., A[i-1])
5     key = A[i]
6     j = i-1
7     while(j>=0 && A[j] <A[key] )
8       # Mueve a los elementos de A[0..i-1], que son mayores a la
9       # clave a una posicion por delante de su posicion actual
10      swap(A[j+1],A[j])
11      j = j-1
12    # Mover al elemento A[i] a la posicion final
13    A[j+1]=key
14  # Retornar la secuencia de elementos ordenada
15 return A[0..n-1]
```

Código 2: Pseudocódigo correspondiente al algoritmo *Insertion Sort*. Las secciones del código asociadas a los comandos **COMPARE** (línea 7) **SWAP** (línea 10) y **SUBMIT** (línea 15) se encuentran resaltadas en color amarillo.

El algoritmo *Bubble Sort* (ver Código 3) tiene como estrategia la comparación metódica de pares de elementos adyacentes y los intercambia si están en el orden incorrecto. Este proceso se repite hasta que todos los elementos quedan ordenados.

```

1 bubbleSort(A[0..n-1])
2   for i = 1:n-1
3     for j = 0:n-i
4       # Verificar si el vecino a derecha es menor
5       if A[j+1] <A[j]
6         # Intercambiar los vecinos
7         swap(A[j+1],A[j])
8   # Retornar la secuencia de elementos ordenada
9 return A[0..n-1]
```

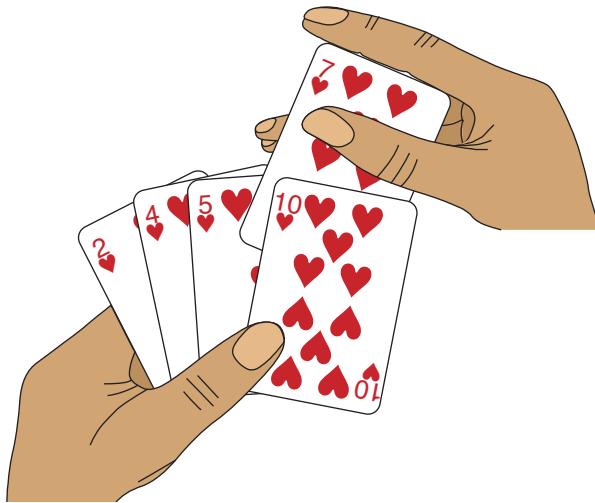
Código 3: Pseudocódigo correspondiente al algoritmo *Bubble Sort*. Las secciones del código asociadas a los comandos **COMPARE** (línea 5) **SWAP** (línea 7) y **SUBMIT** (línea 9) se encuentran resaltadas en color amarillo.

Esta familia de algoritmos ha ido creciendo, dando lugar a nuevos desarrollos más efectivos en complejidad temporal. Para más detalle sobre las características de los algoritmos previamente descriptos y estos nuevos algoritmos se pueden consultar los Capítulos I y II del libro *Introduction to Algorithms* [2].

Es importante resaltar que, por lo general, todos los algoritmos de ordenamiento tienen en común la necesidad de utilizar la comparación entre pares de elementos (que en este trabajo llamaremos **COMPARE**) y el intercambio entre pares de elementos (que llamaremos **SWAP**), y que finalmente devuelven la secuencia final ya ordenada (**SUBMIT**). En los Códigos 1, 2 y 3 correspondientes a *Selection*, *Insertion* y *Bubble Sort* hemos resaltado estos comandos en color amarillo de fondo.

En los cursos de programación, con el fin de motivar el problema de ordenar una lista de elementos, se suelen utilizar diversos escenarios. A modo de ejemplo, podemos citar el exponer simplemente a los alumnos a una lista de números enteros y solicitarles que la ordenen (ej. ordenar la secuencia de números enteros [3, 4, 8, -1, 5] de menor a mayor, de izquierda a derecha). Como escenario de la vida cotidiana, podemos mencionar el de una secuencia de cartas (ver Figura 1, tomada del libro *Introduction to Algorithms* [2]), las cuales deben ser ordenadas por los alumnos de menor a mayor valor. Vamos a ver más adelante que la selección del escenario puede afectar a la problematización del tema y ser un punto crítico al momento de introducir el tema de ordenamiento.

Con el fin de trabajar determinados contenidos, en múltiples cursos se suelen utilizar juegos de computadora basados en diversos escenarios. A partir del análisis de las interacciones de los humanos con las computadoras, en ciertos casos es posible reflejar el comportamiento asociado al funcionamiento de la mente, permitiendo inferir de esta manera eventos no observables. La exploración en base a la interacción humano-computadora fue utilizada por Luis Von Ahn para que individuos resuelvan tareas sencillas para ellos (y hasta divertidas),

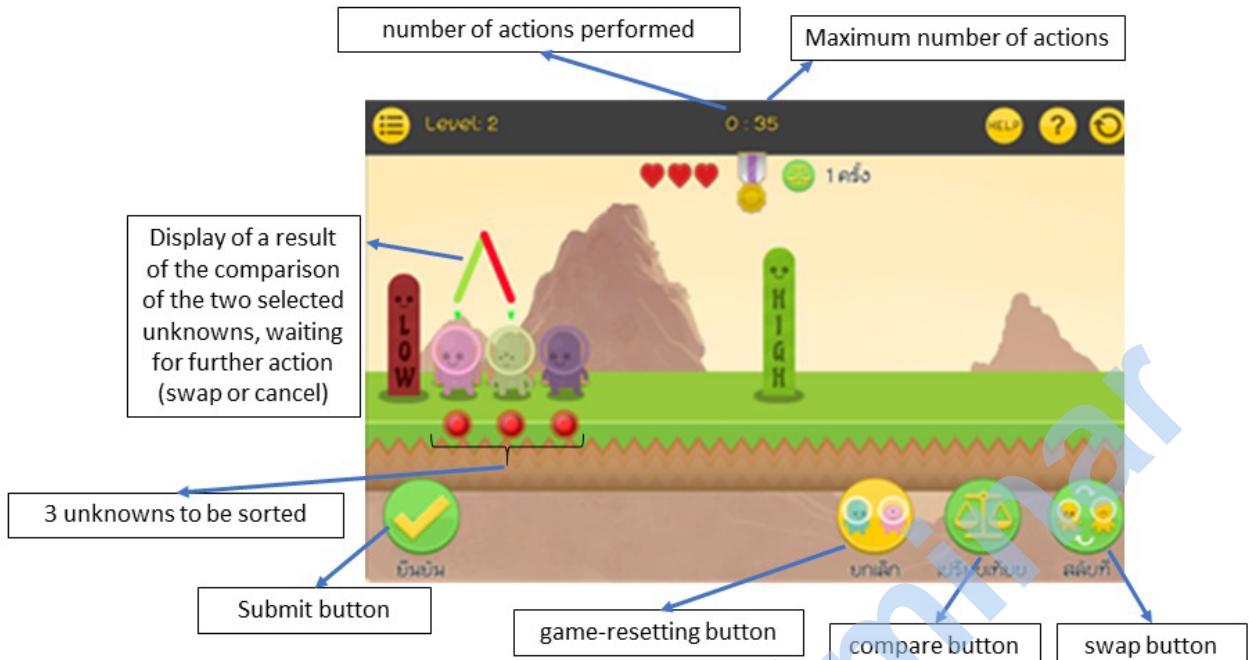


**Figura 1.** Figura 2.1 del libro *Introduction to Algorithms* de Thomas Cormen problematiza el problema de ordenar con la tarea de acomodar cartas [2].

pero complejas para una computadora [3, 4]. Con esta idea de capturar eventos ocultos mediante juegos, surge la propuesta de capturar las interacciones de personas en la tarea de ordenar una lista de elementos. Así, en principio, se podrían caracterizar los distintos procedimientos utilizados por los individuos para ordenar una secuencia de elementos.

A nivel internacional el problema de ordenar ha sido explorado por varios grupos [5–9], en particular nos interesa mencionar el trabajo de un grupo de investigación de Tailandia [10]. En dicho trabajo mencionan que el aprendizaje de los temas asociados a algoritmos de ordenamiento suponen un obstáculo para ciertos estudiantes debido al alto nivel de abstracción necesario para su comprensión. Para evitar el desánimo desarrollaron un juego digital denominado *Sorted* (al que en este trabajo referenciaremos como JUEGOTHAI), cuyo propósito es derribar dicha barrera. En dicho juego, los jugadores tienen la misión de ordenar, de menor a mayor, una secuencia de elementos. En el escenario propuesto, los elementos a ordenar son personajes extraterrestres: los jugadores no conocen los valores de los elementos a ordenar, sólo pueden seleccionarlos de a dos para compararlos (**COMPARE**) o intercambiárselos (**SWAP**) (ver Figura 2). Para la acción **COMPARE**, cuando los dos elementos se encuentran seleccionados, la pantalla muestra cuál es el mayor de ellos. Para la acción **SWAP**, se intercambian las posiciones de los dos elementos seleccionados. Una vez que se está seguro de que todos los elementos están ordenados, se puede presionar el botón de envío (**SUBMIT**). Utilizando la herramienta desarrollada, los autores realizaron un estudio en el cual participaron voluntariamente estudiantes que no tenían conocimientos previos sobre algoritmos de ordenamiento. Los estudiantes jugaron utilizando múltiples niveles (niveles más avanzados contenían secuencias con más elementos a ordenar). Para reflexionar posteriormente sobre el razonamiento de los estudiantes, las tomas de decisiones, etc., se registraron las acciones realizadas por los estudiantes durante el juego. El algoritmo de ordenamiento utilizado (en caso de existir) se puede deducir de las acciones secuenciales registradas (ver, a modo de ejemplo, Figura 3). Finalmente el artículo desarrolla una explicación teórica de cómo los estudiantes adoptan distintos algoritmos de ordenamiento en base a los resultados del juego digital.

Como antecedente local, en los años 2020 y 2021 durante el dictado de la materia Algoritmos y Estructura de Datos I en la FCEN-UBA –en tiempos de pandemia y en modalidad virtual–, desarrollamos un juego de computadora (al que denominaremos como JUEGOUBA) basado en un escenario de cartas (en línea con la bibliografía y la historia de la materia). El desafío era ordenar una secuencia de cartas de manera creciente, de izquierda a derecha, de forma interactiva [11]. De manera previa a dictar el contenido correspondiente al tema de ordenamiento se invitaba a los alumnos a ordenar una secuencia desordenada de cartas utilizando dicho juego (ver Figura 4). La únicas operaciones que podían realizar eran el intercambio entre dos cartas (**SWAP**) y el aviso de que todos los elementos estaban en su posición correcta (**SUBMIT**). El objetivo detrás de esta actividad era doble: i) hacer que cada estudiante experimente el problema de ordenar una secuencia de elementos desordenados previo a comenzar la clase (que identifique diferentes aspectos sobre el problema: su dificultad, casos especiales, posibles soluciones, etc.), y ii) desde el punto de vista de los docentes, poder indagar sobre el método de ordenamiento utilizado por cada estudiante previo a ver los temas, un poco con la intención de responder la pregunta de si los individuos utilizan algún método en particular para ordenar. Como resultado preliminar, los docentes pudieron observar que la estrategia más utilizada entre los estudiantes para ordenar era



**Figura 2.** Captura de pantalla del juego de ordenamiento JUEGOTHAI. Imagen tomada del artículo [10].

No.	Action	Alien Position						
		A	B	C	D	E	F	G
13	COMPARE	49	[1]	[76]	90	36	69	95
14	COMPARE	49	1	[76]	[90]	36	69	95
15	COMPARE	49	1	76	[90]	[36]	69	95
16	SWAP	49	1	76	[90]	[36]	69	95
17	COMPARE	49	1	[76]	[36]	90	69	95
18	SWAP	49	1	[76]	[36]	90	69	95
19	COMPARE	49	[1]	[36]	76	90	69	95

#### (b) Insertion Scheme

Finding the place for unsorted item by comparing the unsorted item to the adjacent item in the sorted list until finding the place to insert.

**Figura 3.** Ejemplo de registro del juego JUEGOTHAI que muestra un esquema de inserción utilizado por un participante. Imagen tomada del artículo [10].

similar a la seguida por el algoritmo *Selection Sort* (más información en [11]).

Al comparar las herramientas JUEGOTHAI y JUEGOUBA notamos que, en relación a los escenarios propuestos, esta última no contaba con la acción **COMPARE** de manera explícita, ya que los alumnos tenían a la vista el valor de todos los elementos (observar los valores de los naipes en la Figura 4). Sin embargo, se puede advertir que los algoritmos clásicos que son introducidos en clase (*Selection Sort*, *Insertion Sort*, *Bubble Sort*, etc.) requieren siempre de la utilización de la operación **COMPARE** (ver en los Códigos 1, 2 y 3 los comandos resaltados en amarillo).

El escenario (ej. extraterrestres en JUEGOTHAI, cartas en JUEGOUBA, secuencia de números, etc.) y las acciones que ofrece un juego de ordenamiento (tanto las disponibles como las necesarias: **COMPARE**, **SWAP**, **SUBMIT**), pensamos que podrían influenciar en la generación de una estrategia de ordenamiento propia de una persona. Generar un algoritmo propio, por parte del participante, puede facilitar la comprensión, en base a similitudes y diferencias, del aprendizaje de algoritmos clásicos.

Para estudiar dicha influencia nos planteamos responder en este trabajo las siguientes preguntas:



Ordenar la secuencia de cartas (la más chica a la izquierda, la más grande a la derecha)

**Checkear**

**Figura 4.** Videojuego JUEGOUBA, de ordenamiento de cartas utilizado en clase de Algoritmos y Estructuras de Datos I, FCEN-UBA. El estudiante debe ordenar las cartas en orden creciente, de izquierda a derecha, utilizando como único movimiento posible el intercambio (SWAP) entre dos de ellas. Al terminar debe presionar el botón “Chequear” (equivalente al comando SUBMIT) para notificar que finalizó la tarea. Se puede observar la notación de arreglos sobre las cartas en la parte superior. El objetivo de esto es que el estudiante identifique la posición de cada carta, y afiance el sistema posicional al momento de pensar en el algoritmo de ordenamiento.

1. ¿Cómo resuelven el ordenamiento aquellas personas expuestas a escenarios donde, por la información disponible a la vista, el uso del comando **COMPARE** ofrecido por la herramienta es **opcional**? ¿Existen diferencias con aquellas personas que son expuestas a escenarios donde el uso del comando **COMPARE** ofrecido por la herramienta es **necesario**?
2. ¿Qué sucede con las personas al momento de pasar a un escenario diferente en el que la información de la relación de orden sólo es observable a través del comando **COMPARE** (y antes no)?

Para responder estas preguntas decidimos desarrollar una herramienta nueva que combine lo visto en JUEGOTHAI y que nos permita registrar eventos como en JUEGOUBA. A esta nueva herramienta la denominamos JUEGOORDENAR.

A continuación, en la siguiente sección describiremos la herramienta implementada, luego enumeraremos los resultados obtenidos para, finalmente, realizar una breve discusión y conclusión.

## 2. Materiales y Métodos

Para responder las preguntas planteadas, hemos desarrollado una herramienta a la que denominamos JUEGOORDENAR. La misma tiene como objetivo desafiar a los usuarios a ordenar, de menor a mayor, una secuencia finita de elementos. Dentro de los requerimientos se incluyó que: i) los comandos a disposición deban ser siempre los siguientes: **COMPARE**, **SWAP** y **SUBMIT**; ii) los elementos deben poder ser disfrazados según el escenario propuesto (escenarios donde el uso del **COMPARE** ofrecido por la herramienta es opcional, y otros donde dicho comando -ofrecido por la herramienta- es necesario para resolver la tarea); iii) debe permitir jugar múltiples niveles (secuencias de distintos largos, escenarios distintos, etc.); iv) debe permitir registrar las decisiones tomadas por los usuarios en cada interacción con la herramienta para un posterior análisis. A continuación detallamos algunos puntos del desarrollo.

## 2.1. Juegos

Dentro de la herramienta diseñamos 3 tipos de escenario: J-NAIPES, J-CUADROS y J-PERFUMES. Todos ellos usan la misma progresión de 6 niveles. A continuación se detallan las secuencias a ordenar según el nivel:

- **Nivel 1:** [3, 2, 1].
- **Nivel 2:** [3, 2, 1, 4].
- **Nivel 3:** [4, 1, 6, 3, 2, 5].
- **Nivel 4:** [5, 6, 4, 2, 1, 3].
- **Nivel 5:** [7, 6, 2, 8, 3, 4, 1, 5].
- **Nivel 6:** [4, 1, 6, 3, 2, 5].

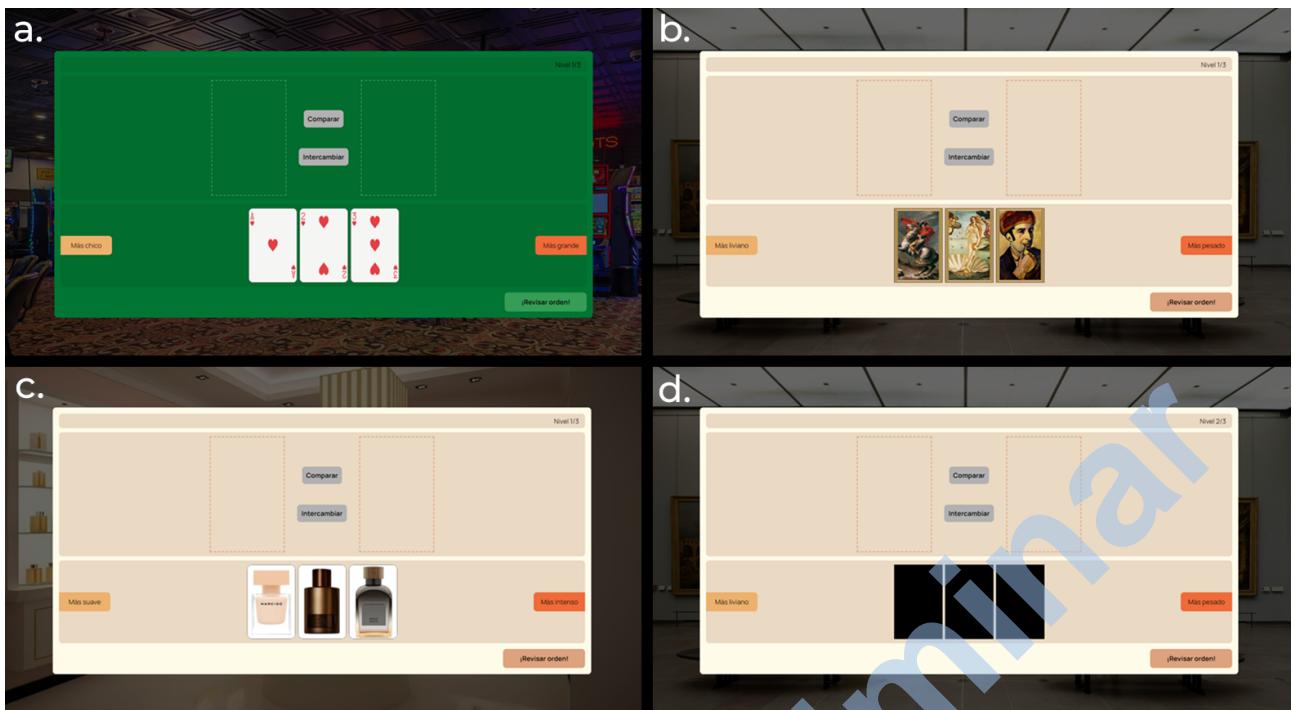
Cada valor numérico del elemento representa a la posición final donde deberá ubicarlo el usuario. En el Nivel 1, el primer elemento hay que ubicarlo al final (posición 3), y el tercero al principio (posición 1), mientras que el elemento del medio ya se encuentra ordenado (posición 2).

La diferencia entre los 3 tipos de juegos es el escenario que le hemos aplicado a los primeros 5 niveles. En J-NAIPES los primeros 5 niveles corresponden a un escenario donde los elementos son disfrazados de naipes; en J-CUADROS los elementos de dichos niveles son representados como cuadros; y en J-PERFUMES se utilizan perfumes para representar a los elementos. En cuanto al último nivel (Nivel 6) en los 3 juegos, corresponde a un mismo escenario: elementos indistinguibles. Este nivel tiene como objetivo intentar responder nuestra segunda pregunta de cuál es el impacto de pasar de un escenario donde el uso del comando **COMPARE** ofrecido por la herramienta es opcional (ej. naipes) a uno donde dicho comando (ofrecido por la herramienta) es necesario para cumplir el objetivo de ordenar los elementos (ej. elementos indistinguibles).

## 2.2. Escenarios

Como hemos mencionado, la herramienta implementada permite contar con diversos escenarios. A continuación un detalle de cada uno de ellos.

- **Naipes (baraja francesa).** En este ambiente los elementos son cartas francesas (ver Figura 5a). Este escenario fue introducido para que el usuario pueda optar si usar o no el comando **COMPARE** ofrecido por la herramienta, ya que el valor de los naipes se visualizan a simple vista (los valores de ambas cartas a comparar son visibles para el usuario), y por lo tanto no es obligatorio su uso.
- **Cuadros.** En este caso los elementos son pinturas (ver Figura 5b) y el valor asociado corresponde al peso del cuadro (ej. en kilogramos). En este escenario se quiere explotar el hecho de que el usuario desconoce el valor del elemento y tiene que recurrir necesariamente al uso del comando de comparación de elementos (utilizar el comando **COMPARE** ofrecido por la herramienta) para conocer su valor relativo. Si bien en una primera instancia pensamos en utilizar el valor monetario (ej. \$) del cuadro a modo de valor del elemento, desistimos de dicha idea ya que los individuos podrían tener alguna noción del costo de ciertos cuadros y estaríamos en un escenario similar al de los naipes (donde se conoce el valor y no es necesario el uso del comando **COMPARE** ofrecido por la herramienta).
- **Perfumes.** En este escenario los elementos son perfumes (ver Figura 5c) y el valor asociado corresponde a la intensidad de su aroma. Este caso es similar al escenario de los cuadros y tiene como fin ser utilizado a modo de escenario alternativo, para evaluar si el cambio de contexto modifica las respuestas de los individuos.
- **Elementos indistinguibles.** En este ambiente cada elemento es un rectángulo de color negro, y la característica visual principal es que los rectángulos son indistinguibles entre sí (ver Figura 5d). De esta manera el usuario requiere necesariamente el uso del comando de comparación entre elementos (**COMPARE** ofrecido por la herramienta) para conocer la relación de orden entre ellos, ya que el valor de cada uno de ellos no es posible visualizarlo.



**Figura 5.** Se pueden observar los 4 escenarios disponibles en la herramienta JUEGOORDENAR: a. Naipes, b. Cuadros, c. Perfumes, d. Indistinguibles

En el caso de los cuadros y los perfumes, los juegos cuentan con una cantidad suficiente de cuadros distintos para que nunca se repitan durante todo el juego. El motivo de esto es evitar que los usuarios asocien un valor a cada cuadro (y perfume) puntual, ya que si dicho cuadro se utiliza en varios niveles, los individuos potencialmente podrían llegar a recordar relaciones de orden de escenarios previos.

### 2.3. Encuesta

Con el fin de poder agrupar a los jugadores en una futura etapa de análisis, la herramienta recolecta datos básicos a través de una encuesta inicial. La misma se realiza como primer paso, al iniciar el juego. Entre los datos solicitados se encuentran la edad y la cantidad de años de experiencia programando (entre otros). También se realiza una encuesta final, luego de terminado el juego, con comentarios sobre la estrategia utilizada (para verificar si le asignan un nombre, por ejemplo *Insertion Sort*).

### 2.4. Tutorial

Al terminar la encuesta inicial, y previo al primer nivel, el juego se inicia con un tutorial autoguiado donde el usuario puede comprender el uso de la herramienta. Con el fin de evitar influenciar al usuario, se tuvo especial cuidado en que los pasos seguidos por el tutorial no se asemejen a los realizados por los algoritmos de ordenamiento.

### 2.5. Implementación de la herramienta

La herramienta implementada se puede acceder a través de un navegador web. En cuanto a la lógica de los 3 juegos, los elementos a ordenar han sido representados como números enteros. Al momento de la visualización, cada uno de esos números es asociado a algún elemento visual.

Algunos requerimientos funcionales que se tuvieron en cuenta fueron los siguientes:

- **Asignación del juego.** El acceso a la herramienta debía ser a través de una única dirección web para

todos los usuarios. De manera independiente, la herramienta debía ser la encargada de seleccionar de manera aleatoria (siguiendo una distribución uniforme<sup>1</sup>) el juego para cada participante, es decir, una opción entre: J-NAIPES, J-CUADROS y J-PERFUMES.

- **Cantidad de elementos a ordenar.** Se decidió contar con un mínimo de 3 elementos para no tener un escenario trivial, y un máximo de 8 elementos para que los usuarios no demoren demasiado en la tarea de ordenar.
- **Exclusión de reglas.** A continuación, detallamos algunas de las reglas que se decidieron excluir y el motivo.
  - **Limitar la máxima cantidad de acciones (COMPARE, SWAP, SUBMIT) que puede realizar el jugador.** El motivo para excluir esta regla es el no inducir a los jugadores a tomar ningún camino puntual (en este caso, aquel que esté dentro de la mínima/máxima cantidad de acciones posibles).
  - **Limitar el tiempo disponible para cumplir con el objetivo.** Esto también intenta evitar la presión que puede llevar a tomar decisiones únicamente por una restricción temporal.
  - **Incluir elementos repetidos en la lista a ordenar.** Se decidió no incluir elementos repetidos en la lista a ordenar (orden total) dado que si aceptábamos utilizar orden parcial, la acción de comparación entre dos elementos (COMPARE) iba a requerir de un tercer resultado como respuesta (*igualdad*), con lo cual preferimos mantener un orden total estricto para simplificar el análisis posterior y concentrarnos en escenarios donde las repuestas sean únicamente *menor/mayor*.

Si bien estas reglas fueron excluidas de la lógica de los juegos, el código implementado permite incluir de manera simple a cada una de ellas.

## 2.6. Infraestructura

Para que la herramienta se encuentre disponible en línea en un sitio público de internet, decidimos utilizar el servicio de Google denominado Firebase, puntualmente Firebase Hosting<sup>2</sup>, que permite publicar sitios con un diseño one-pager<sup>3</sup>, como es el caso de esta herramienta, de una forma muy sencilla y de manera gratuita.

Para la persistencia de los datos recolectados nos orientamos por la solución ofrecida por el Firebase Hosting. Puntualmente utilizamos el Firebase Realtime Database<sup>4</sup> que brinda una base de datos no relacional clave-valor básica y de uso gratuito. Dicha infraestructura se ajustaba a nuestros requerimientos. El resultado parcial o total de cada juego y los metadatos asociados son persistidos en esta base de datos. Adicionalmente, permite exportar de manera trivial los datos a un archivo en formato JSON para procesarlos posteriormente.

Las decisiones tomadas en cuanto a la infraestructura nos permitieron liberarnos de la necesidad de contar con un servidor propio para poner en línea la herramienta y persistir los datos, haciendo la implementación más rápida y sencilla.

## 2.7. Métricas

Para responder a la pregunta 1. *¿Cómo resuelven el ordenamiento aquellas personas expuestas a escenarios donde, por la información disponible a la vista, el uso del comando COMPARE ofrecido por la herramienta es opcional? ¿Existen diferencias con aquellas personas que son expuestas a escenarios donde el uso del comando COMPARE ofrecido por la herramienta es necesario?* decidimos seleccionar el Nivel 3 y contabilizar, para cada usuario, la cantidad de veces que utilizaron en nuestra herramienta el comando COMPARE. Adicionalmente, analizaremos el tiempo utilizado por cada persona para resolver el Nivel 3.

En referencia a la pregunta 2. *¿Qué sucede con las personas al momento de pasar a un escenario diferente en el que la información de la relación de orden sólo es observable a través del comando COMPARE (y antes no)?* decidimos seleccionar el Nivel 6 (posee la misma cantidad de elementos que el Nivel 3 y en el mismo

<sup>1</sup>Random en Javascript y su distribución: <https://tc39.es/ecma262/#sec-math.random>

<sup>2</sup>Firebase Hosting: <https://firebase.google.com/docs/hosting>

<sup>3</sup>Diseño one-pager: Un sitio web de una sola página (one page website) es aquel que presenta todo su contenido en una misma página web, en lugar de tener varias páginas separadas. En este tipo de sitio web, no se incluyen páginas adicionales como "Acerca", "Contacto", etc.

<sup>4</sup>Firebase Realtime Database: <https://firebase.google.com/docs/database>

orden inicial, pero en este caso todos los usuarios se enfrentan al escenario de elementos indistinguibles), y nuevamente contabilizar, para cada usuario, la cantidad de veces que utilizaron el comando **COMPARE**. Al igual que en la pregunta anterior, buscaremos ver qué sucede con el tiempo utilizado por cada persona para resolver ambos niveles.

## 2.8. Participantes

La herramienta JUEGOORDENAR se puso a prueba de manera anónima con estudiantes de las siguientes materias, dictadas durante el primer cuatrimestre de 2024:

- Laboratorio de Datos, dictado por el Departamento de Computación, FCEN-UBA (comisión única)
- Biometría I, dictado por el Departamento de Ecología, Genética y Evolución, FCEN-UBA (2 comisiones)

Los estudiantes fueron invitados a participar al término de una clase en el mismo laboratorio donde estaban. La participación consistió en utilizar una única vez la herramienta y los juegos a los que fueron expuestos (J-NAIPES, J-CUADROS y J-PERFUMES) fueron asignados al azar (por la misma herramienta), finalizando:

- 27 estudiantes que utilizaron el juego J-NAIPES
- 29 estudiantes que utilizaron el juego J-CUADROS
- 33 estudiantes que utilizaron el juego J-PERFUMES

Aquellas personas que decidieron no completar la prueba fueron excluidas del análisis (n=4).

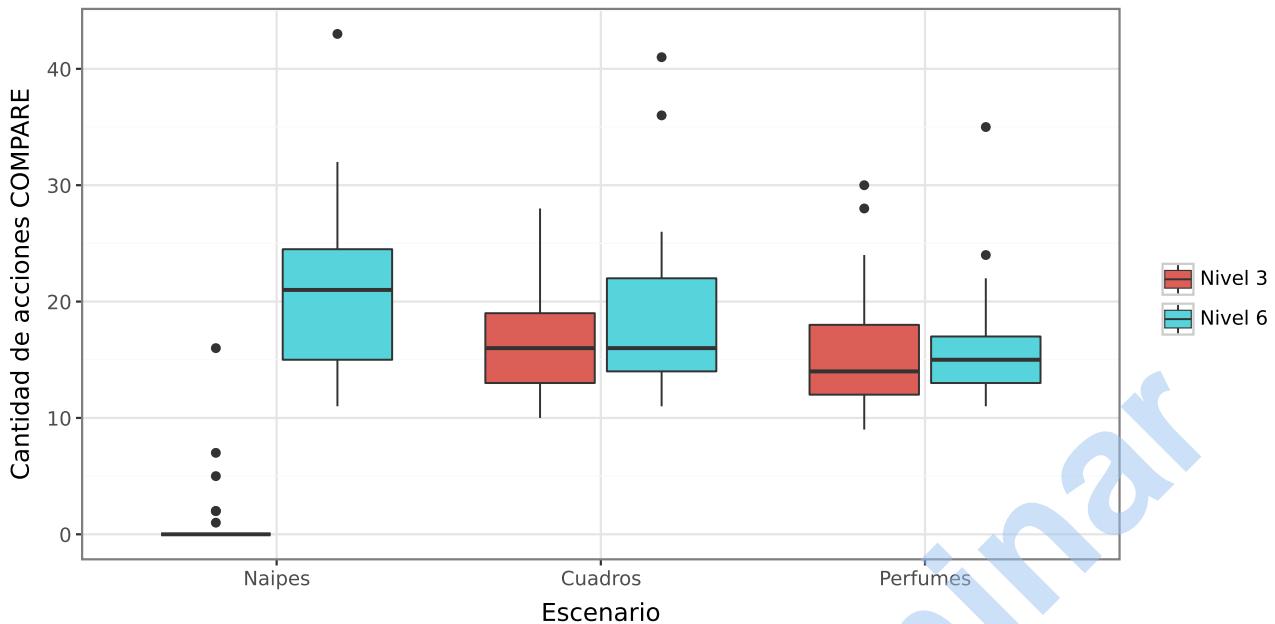
## 3. Resultados y Discusión

Para un mejor entendimiento de los resultados obtenidos es importante recordar que el Nivel 3 y el Nivel 6 tienen exactamente la misma cantidad de elementos a ordenar y se encuentran inicialmente en la misma posición (ver dichos niveles en subsección 2.1 Juegos).

En la Figura 6 se pueden observar la distribución de los datos correspondientes a la cantidad de veces que los usuarios utilizaron el comando **COMPARE** ofrecido por la herramienta. Los datos correspondiente al Nivel 3 se encuentra en color rojo y Nivel 6 en celeste. Si nos centramos sólo en el Nivel 3 (rojo), se puede observar que en el caso de los Naipes, donde el uso del comando **COMPARE** ofrecido por la herramienta es opcional, la mayoría de los estudiantes decidió no utilizarlo. En cambio, en los casos donde el uso del comando **COMPARE** era necesario para resolver el problema (Cuadros y Perfumes), se observa la utilización del mismo. Se observa también que la cantidad de veces que fue utilizado parece ser similar tanto en el escenario de cuadros como en el de perfumes.

La variación en la cantidad de veces que el comando **COMPARE** (ofrecido por la herramienta) fue usado por los distintos usuarios puede estar asociada a la estrategia que utilizó cada estudiante para resolver el problema. Queda para una investigación futura analizarlo.

A la pregunta 1. *¿Cómo resuelven el ordenamiento aquellas personas expuestas a escenarios donde, por la información disponible a la vista, el uso del comando COMPARE ofrecido por la herramienta es opcional? ¿Existen diferencias con aquellas personas que son expuestas a escenarios donde el uso del comando COMPARE ofrecido por la herramienta es necesario?*, encontramos evidencia de que los estudiantes que tienen disponible el comando **COMPARE**, pero no es necesario para resolver el problema, no lo utilizan. Esto se observa en la Figura 6, en la caja roja correspondiente a Naipes en Nivel 3, que presenta una mediana en cero, y su rango intercuartil es nulo. De manera contraria, aquellos estudiantes que fueron expuestos a un escenario donde el comando **COMPARE** era necesario, es el caso de los Cuadros y los Perfumes, lo han utilizado. Esto último se puede observar en las cajas rojas correspondientes a Cuadros y Perfumes Nivel 3, las cuales presentan una mediana superior (16 para cuadros y 14 para perfumes, con un rango intercuartil de 6 en ambos casos). Se puede notar que estos dos últimos escenarios no presentan casos de valores menores a diez. Queda a futuro analizar si las comparaciones realizadas por los usuarios fueron las indicadas para reconocer el orden final (o simplemente utilizaron varias veces el comando **COMPARE** sin un fin determinado).



**Figura 6.** Cantidad de acciones COMPARE (ofrecidas por la herramienta) fueron utilizadas en cada escenario. Escenarios con valores a la vista: Naipes en Nivel 3; sin valores a la vista: Cuadros y Perfumes en Nivel 3, y en todos los casos Nivel 6 (objetos indistinguibles).

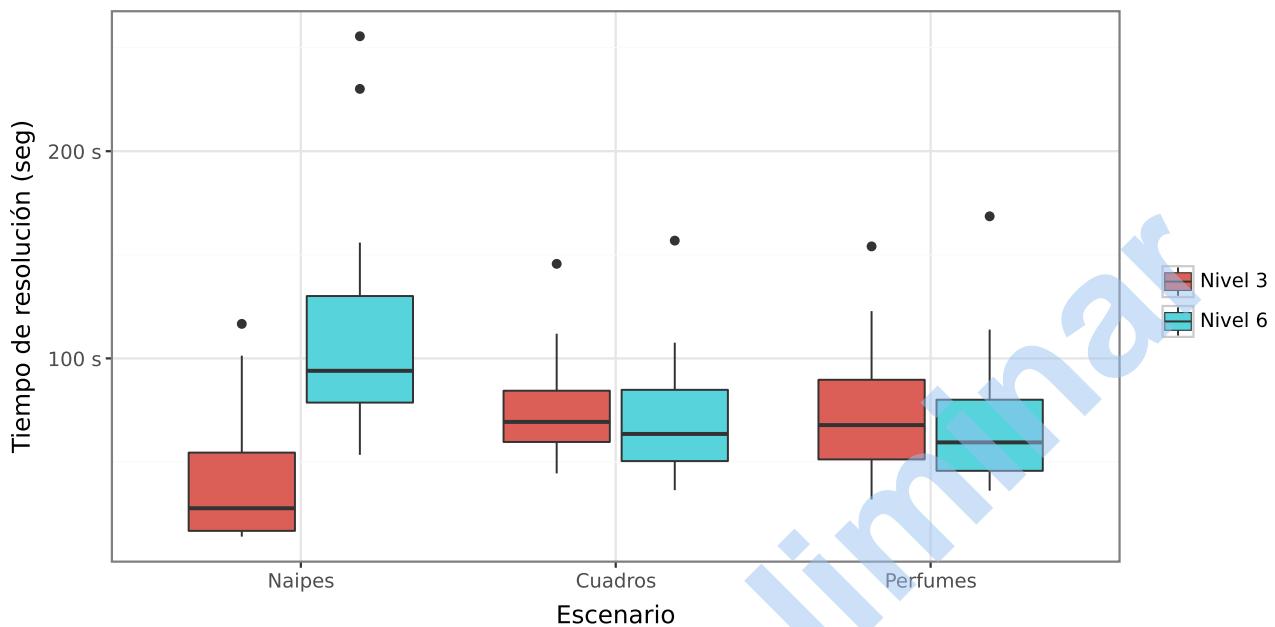
Este primer hallazgo es muy importante ya que, como nos ha ocurrido en JUEGOUBA (y también como se observa en la Figura 1), solemos exponer a alumnos a escenarios en los cuales no se encuentran con la necesidad de utilizar conceptos que luego queremos introducir (algoritmos donde es necesario el uso del COMPARE) y por lo tanto puede impactar a la hora de querer introducir analogías.

En cuanto a la segunda pregunta que nos planteamos en el presente trabajo: *2. ¿Qué sucede con las personas al momento de pasar a un escenario diferente en el que la información de la relación de orden sólo es observable a través del comando COMPARE (y antes no)?*, en la Figura 6 podemos observar también la distribución de los datos correspondientes al Nivel 6 (cajas celestes). Podemos observar que en el caso de los estudiantes expuestos en un principio a un escenario de Naipes, al exponerlos luego a un escenario en el cual requieren del uso del comando COMPARE terminan utilizandoarlo. Esto se puede observar en el cambio sufrido en la cantidad de veces que utilizan el comando COMPARE, que ronda el cero en el Nivel 3 (caja roja), pero que en el Nivel 6 este número se eleva a una mediana de 21 (caja celeste). Si comparamos la cantidad de comparaciones que realizaron los estudiantes que ya venían utilizando el comando COMPARE (J-CUADROS y J-PERFUMES) con los que no lo hacían (J-NAIPES), podemos observar que la mediana de los datos correspondiente a los primeros (Cuadros y Perfumes en Nivel 6) es menor que la de los segundos (Naipes en Nivel 6). Esto creemos que puede estar asociado a la falta de entrenamiento de los que utilizaron originalmente Naipes. Este resultado podría pensarse que puede extrapolarse a un ambiente educativo, donde a los estudiantes que no han sido expuestos al uso del comando COMPARE de manera explícita les puede dificultar un poco más comprender la necesidad de su uso. Finalmente, podemos observar que aquellos que en el Nivel 3 habían sido expuestos a escenarios de Cuadros y Perfumes, en el Nivel 6 las medianas correspondientes a la cantidad de comandos COMPARE utilizados se mantiene similar al Nivel 3, si bien en el caso de los Cuadros el tercer cuartil tiende a ser un poco mayor.

En relación a los tiempos de resolución de los niveles, en la Figura 7 se pueden observar los resultados para los niveles 3 y 6 en los 3 juegos (J-NAIPES, J-CUADROS y J-PERFUMES). En relación a la pregunta 1, el tiempo necesario para resolver el Nivel 3 es menor en el escenario que cuenta con la información del orden a la vista (J-NAIPES) que en los otros escenarios (J-CUADROS y J-NAIPES). En línea con lo que vimos en la figura anterior para el Nivel 3 (Figura 6), las métricas de tiempos de J-CUADROS y J-PERFUMES son muy similares entre sí, mientras que lo capturado del grupo J-NAIPES es menor que a estos dos juegos. Esto es esperable ya que los estudiantes que jugaron J-NAIPES utilizaron muchas menos veces el comando COMPARE ofrecido por la herramienta. Queda a futuro analizar el uso, tanto en cantidad como en tiempo, de los otros dos comandos (SWAP y SUBMIT).

Con respecto a la pregunta 2, quienes participaron en el escenario de J-NAIPES en el Nivel 3, al pasar al Nivel

6 el mismo aumenta notoriamente, incluso por sobre los otros dos juegos (J-CUADROS y J-PERFUMES). Esto es razonable si pensamos que los que jugaron originalmente J-NAIPES deben, recién en este nivel, utilizar de manera obligatoria el comando **COMPARE** ofrecido por la herramienta y desarrollar una nueva estrategia de solución, ya que la utilizada en niveles anteriores no son de utilidad en presencia de este nuevo escenario.



**Figura 7.** Tiempo total utilizados en cada escenario. Escenarios con valores a la vista: Naipes en Nivel 3; sin valores a la vista: Cuadros y Perfumes en Nivel 3, y en todos los casos Nivel 6 (objetos indistinguibles).

## 4. Conclusiones

Al hacer una revisión de los escenarios utilizados para introducir algoritmos de ordenamiento, notamos que algunos escenarios no eran propicios para tal fin debido a que no requerían del uso explícito de uno de los comandos más usados por los algoritmos clásicos: el comando **COMPARE**.

Para corroborar nuestra hipótesis, construimos una herramienta computacional que captura eventos de un juego cuyo objetivo es ordenar elementos por parte del usuario. Los elementos a ordenar son camuflados con distintos disfraces. Contamos con disfraces distinguibles (ej. Naipes) para los cuales no es necesario el uso del comando **COMPARE** ofrecido por la herramienta, y hay disfraces que tornan a los elementos en indistinguibles (cuadros, perfumes y rectángulos negros) donde el uso del **COMPARE** ofrecido por la herramienta es imprescindible.

A partir de los datos recolectados pudimos observar que en los escenarios donde el uso del comando **COMPARE** ofrecido por la herramienta es opcional, los estudiantes optaron por no utilizarlo. Luego, al introducir un escenario donde el uso del comando **COMPARE** ofrecido por la herramienta no es opcional, los estudiantes tuvieron ciertos inconvenientes en resolver el problema, utilizando más **COMPARE** y más tiempo en la resolución que los estudiantes que habían sido expuestos a escenarios contrarios.

A modo de conclusión final, y en base a la experiencia relatada en este trabajo, recomendamos al momento de planificar una actividad para el tema de ordenamiento utilizar escenarios donde la información y acciones disponibles sean similares a las que tiene y utilizan luego los algoritmos al resolverlo. Esto es, que incluya los comandos básicos a utilizar por el algoritmo a introducir en clase.

## Referencias

- [1] Amruth N. Kumar, Rajendra K. Raj, Sherif G. Aly, Monica D. Anderson, Brett A. Becker, Richard L. Blumenthal, Eric Eaton, Susan L. Epstein, Michael Goldweber, Pankaj Jalote, Douglas Lea, Michael

Oudshoorn, Marcelo Pias, Susan Reiser, Christian Servin, Rahul Simha, Titus Winters, and Qiao Xiang. Computer Science Curricula 2023. Association for Computing Machinery, New York, NY, USA, 2024.

- [2] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. Introduction to algorithms. MIT press, 2022.
- [3] Luis Von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. recaptcha: Human-based character recognition via web security measures. Science, 321(5895):1465–1468, 2008.
- [4] Luis Von Ahn and Laura Dabbish. Designing games with a purpose. Communications of the ACM, 51(8):58–67, 2008.
- [5] Gordon Rugg and Peter McGeorge. The sorting techniques: a tutorial paper on card sorts, picture sorts and item sorts. Expert Systems, 22(3):94–107, 2005.
- [6] Manargul Mukasheva, Zukhra Kalkabayeva, and Nurbek Pussyrmmanov. Visualization of sorting algorithms in the virtual reality environment. In Frontiers in Education, volume 8, page 1195200. Frontiers Media SA, 2023.
- [7] Beth Simon, Tzu-Yi Chen, Gary Lewandowski, Robert McCartney, and Kate Sanders. Commonsense computing: what students know before we teach (episode 1: sorting). In Proceedings of the second international workshop on Computing education research, pages 29–40, 2006.
- [8] Joshua Schrier, Michael F Tynes, and Lillian Cain. Determining the activity series with the fewest experiments using sorting algorithms. Journal of Chemical Education, 98(5):1653–1658, 2021.
- [9] Olivia Witanowska. The use of virtual manipulatives in teaching sorting algorithms. 2020.
- [10] Suparat Chuechote, Artorn Nokkaew, Apichat Phongsasithorn, and Parames Laosinchai. A neo-piagetian analysis of algorithmic thinking development through the “sorted” digital game. Contemporary Educational Technology, 12(1):1–15, 2020.
- [11] Dario Ocles, Pablo Turjanski, and Matías Lopez-Rosenfeld. A ordenar, cada cosa en su lugar: una experiencia de migración a la virtualidad de contenidos sobre algoritmos de ordenamiento en la universidad. In Simposio Argentino de Educación en Informática - Jornadas Argentinas de Informática, 2021.

# **Impacto de un módulo de formación docente en las opiniones sobre arquitectura de computadoras, inteligencia artificial y problemáticas asociadas**

Julián Dabbah<sup>1</sup>, Tomás Caballero<sup>1</sup>, Marcos J. Gómez<sup>1,2</sup>.

<sup>1</sup> Fundación Sadosky, CABA, Argentina

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

[jdabbah@fundacionsadosky.org.ar](mailto:jdabbah@fundacionsadosky.org.ar), [tcaballero@fundacionsadosky.org.ar](mailto:tcaballero@fundacionsadosky.org.ar), [marcos.gomez@unc.edu.ar](mailto:marcos.gomez@unc.edu.ar)

## **Resumen**

En este trabajo presentamos la implementación de un módulo parte de una actualización docente a distancia cuyos contenidos giran en torno a la arquitectura de computadoras, la inteligencia artificial y temas sociales puntuales relacionados a estos contenidos. Para conocer el impacto de las actividades y contenidos del módulo, recuperamos la opinión de las y los docentes sobre algunas afirmaciones relacionadas con estos tres ejes temáticos en un esquema de pre pos test. Después de analizar las diferencias entre estas dos instancias podemos concluir que impacta de forma estadísticamente significativa en opiniones asociadas a los tres ejes. En el trabajamos presentamos herramientas, diseño de estudios y resultados, para comenzar a discutir próximas líneas de acción.

**Palabras clave:** Formación Docente; Inteligencia artificial; Arquitectura de computadoras; Ciudadanía digital; Misconceptions.

## **1. Introducción**

La vida en el mundo contemporáneo está -desde las últimas décadas- cada vez más atravesada por las tecnologías computacionales y particularmente, en el último tiempo, por las denominadas de “inteligencia artificial”. Por lo tanto, para poder participar de este mundo, de manera activa, crítica e informada se requiere comprender estas tecnologías y habilitar la posibilidad de construirlas. Según Dabbah et al [1] “Las [Ciencias de la Computación] CC entonces son vistas como una disciplina que resulta fundamental tanto para comprender un mundo cada vez más organizado y dependiente de la tecnología como para poder participar activa y soberanamente en debates actuales sobre la interacción entre las computadoras y la sociedad, y por extensión sobre la ciudadanía democrática y en pleno ejercicio de derechos y responsabilidades.”

En esta línea, la escuela -como responsable de la formación de esta ciudadanía crítica y democrática- demanda la incorporación de contenidos de CC como parte de la currícula obligatoria para ofrecerlos universal y equitativamente a las y los ciudadanos. Democratizar estos conocimientos, entonces, requiere la formación de docentes, el desarrollo de material didáctico e implementación de diseños curriculares específicos.

En línea con el reconocimiento de la importancia de formar docentes en contenidos de CC, la Iniciativa Program.AR de la Fundación Sadosky, en convenio con el Instituto Nacional de Formación Docente (INFoD), dictaron durante 2022 y 2023 el programa de actualización docente “Actualización Académica en Educación Tecnológica con foco en la Programación”. La formación fue enteramente virtual con un total de 200hs de las cuales 40 correspondían al módulo “La ciudadanía en un mundo atravesado por computadoras”, en el que se abordaban contenidos de arquitectura de computadoras, redes e internet e inteligencia artificial, tanto desde una mirada disciplinar como política y ciudadana.

El módulo estuvo acompañado de diferentes herramientas de evaluación, una de ellas vinculada específicamente a recuperar conocimientos previos y opiniones sobre las computadoras y la inteligencia artificial. El presente estudio

evalúa el impacto de esta parte de la formación en las concepciones previas de las y los docentes sobre la construcción y organización de las computadoras, la neutralidad y las limitaciones de las aplicaciones de IA y temas sociales puntuales relacionados con estos contenidos: el problema de la obsolescencia programada, la recolección de datos por parte de las plataformas de uso gratuito y la aplicación indiscriminada de soluciones basadas en IA para cualquier tipo de problema.

Para abordar el impacto del módulo definimos dimensiones vinculadas a las opiniones de las y los docentes sobre estos temas. Para cada una de las dimensiones, planteamos hipótesis no direccionales para evaluar el impacto en sus opiniones de las actividades y contenidos abordados en el módulo.

“No existe diferencia estadística significativa en la opinión de las/os docentes antes y después de la participación del módulo sobre que...

- H1(nula) ... para considerar a un dispositivo como computadora es imprescindible que tenga procesador y memoria RAM.”
- H2(nula) ... todas las computadoras tienen los mismos componentes que una computadora personal.”
- H3(nula) ... la memoria RAM y los dispositivos de almacenamiento tienen funciones distintas.”
- H4(nula) ... no siempre es necesario contar con los mejores componentes de hardware para que el funcionamiento de la computadora sea el que esperamos.”
- H5(nula) ... las personas influyen en los resultados de las aplicaciones de IA.”
- H6(nula) ... la IA puede aplicarse en cualquier ámbito.”
- H7(nula) ... la IA puede cometer errores.”
- H8(nula) ... las aplicaciones con IA son neutrales y pueden discriminar o prejuzgar.”
- H9(nula) ... la IA es superior a las personas para resolver problemas.”
- H10(nula) ... parte de la estrategia comercial de las empresas es que los dispositivos que fabrican queden desactualizados rápidamente.”
- H11(nula) ... los datos que generamos al usar las plataformas de uso gratuito son altamente valiosos económicamente.”
- H12(nula) ... es necesario regular para qué puede usarse la IA y para qué no.”

Para evaluar el impacto del módulo, diseñamos un pre y un pos test, en el cual las y los docentes indicaron su grado de acuerdo o desacuerdo con diferentes afirmaciones que reflejan las hipótesis planteadas y analizamos estadísticamente las diferencias entre las respuestas de ambas instancias.

El artículo está organizado de la siguiente manera: en la Sección *Trabajo previo*, recuperamos trabajo previo vinculado a formación docente a nivel mundial. Luego, en *Descripción del curso de formación* describimos el módulo implementado, sus actividades y objetivos. En *Diseño del estudio* introducimos características generales de los grupos de docentes que participaron del estudio y las herramientas de recolección de datos. En *Resultados* presentamos y analizamos los resultados obtenidos a partir de las intervenciones, y en *Conclusiones*, resumimos los hallazgos y planteamos posible trabajo futuro.

## 2. Trabajo Previo

En los últimos 20 años muchos países, incluido Argentina, han avanzado en la introducción de contenidos vinculados a las Ciencias de la Computación (CC), en particular de programación, como parte de la currícula escolar obligatoria. Durante los procesos de diseño e implementación, un tema de debate entre académicos, funcionarios públicos y la comunidad educativa es quién enseñará estos contenidos en las escuelas, ya que en la mayoría de los casos, no se contaba con docentes con formación específica en contenidos de programación ni tampoco existían carreras terciarias o universitarias para formar docentes en el área. Un punto crucial a tener en cuenta es que las y los docentes de CC que forman parte del sistema educativo actual no tienen necesariamente conocimientos previos en la disciplina, ya que muchos de ellos provienen de áreas relacionadas a las Tecnologías de la Información y la Comunicación (TIC), por lo que hacen foco en enseñar a utilizar programas en vez de centrarse en enseñar contenidos específicos de Computación y en particular de programación [2, 3, 4].

Si bien la mayoría de los investigadores/as e impulsores/as de políticas públicas coinciden en que las carreras terciarias o universitarias son la mejor opción para formar docentes calificados [2], ante la falta de docentes con formación específica muchos países ofrecen formaciones para docentes que ya están en actividad (Reino Unido, Nueva Zelanda, Estados Unidos y Alemania) [2, 3, 4, 5]. Esta oferta de cursos a corto plazo está particularmente motivada por la ausencia de profesorados de CC, la formación heterogénea de los docentes de programación en actividad y la falta de profesionales de la industria del software interesados en la enseñanza.

Si bien la mayoría de las investigaciones realizadas [2,4] parecen mostrar que las y los docentes se sienten más seguros con respecto a la enseñanza de la programación luego de participar de cursos de formación docente, no es claro cómo incorporan en sus clases los contenidos aprendidos en los cursos y qué relación existe entre las actividades propuestas en los cursos y las prácticas de enseñanza que luego llevan a sus aulas.

En Argentina se han implementado diferentes experiencias para formar docentes en contenidos vinculados a las CC, en particular de programación. Encontramos experiencias de cursos cortos (de entre 40 y 120 horas) [7, 8, 9], pero también

formaciones extensas como especializaciones vinculadas a la didáctica de las CC [9, 10, 11]. Dentro de los cursos cortos, Martínez et al [7] implementaron una formación de 50hs en la Universidad Nacional de Córdoba enfocada en conceptos fundamentales de programación para docentes en actividad y de diversas especialidades. En [7] se comparten los datos de dos cohortes del curso. Por un lado, observaron que las y los docentes cambiaron sus percepciones sobre la enseñanza de la programación, logrando diferenciar a la programación de enseñar ofimática y uso de las TICs. Sin embargo, el curso fue insuficiente para preparar a las y los docentes sin formación previa en programación ya que abordaron sin profundidad, y con errores importantes, los conceptos de programación en sus aulas. Con respecto al cambio de la percepción sobre qué es la programación, la iniciativa Program.AR, implementó 7 ediciones del curso de formación docente “La programación y su didáctica”, en convenio con universidades nacionales, alcanzando 23 provincias y más de 4500 participantes [19]. Scasso et al [8] describen: “*A partir de la experiencia del curso, se manifiesta en los docentes participantes un cambio de paradigma, un corrimiento desde una mirada más “tradicional” de las TIC, basado en el modelo de la ofimática, a un posicionamiento anclado en el desarrollo del pensamiento computacional a partir de la enseñanza de la programación*”. Con respecto a los contenidos vinculados a la programación, las y los docentes estuvieron interesados pero no hay datos sobre la incorporación conceptual.

En cuánto a cursos de formación extensos, la iniciativa Program.AR también impulsó la apertura de ofertas de especialización docente en didáctica de CC, promoviendo una articulación entre universidades e institutos de formación docente, estableciendo como condición la postulación conjunta de ambas instituciones [9, 10, 11]. 8 universidades fueron elegidas para diseñar e implementar especializaciones de 400hs en contenidos de CC. Cada universidad definió contenidos y módulos a abordar. Cura et al [11], analizaron el impacto de la especialización con respecto a la incorporación de contenidos específicos de CC y la estrategia pedagógica en 3 universidades. Con respecto a los contenidos específicos de CC evaluaron los ejes de **programación y sistemas computacionales**. Según Cura et al [11] la gran mayoría ha adquirido conocimientos de contenido y considera incorporar contenidos de Ciencias de la Computación en diferentes escenarios.

Teniendo en cuenta las experiencias previas, consideramos que lograr un cambio de opinión en percepciones erróneas sobre los contenidos puede ser el primer paso para lograr una incorporación conceptual en profundidad. Con base en esta situación, diseñamos un estudio para evaluar el impacto de un módulo de formación docente continua en opiniones de las y los docentes con respecto a la arquitectura de computadoras, la inteligencia artificial, la obsolescencia programada, el valor de los datos y la responsabilidad ética al desarrollar sistemas de IA.

### 3. Descripción del curso de formación

En esta sección describimos el módulo implementado para la formación docente. En la subsección *Contexto de implementación del módulo* presentamos el programa del cual forma parte el módulo, su modalidad, objetivos generales y ejes de contenidos. En la subsección *Clases y actividades* presentamos y describimos las clases y actividades que forman parte del módulo.

#### 3.1. Contexto de implementación módulo

El curso en cuestión fue uno de los módulos de la *Actualización Académica de Educación Tecnológica con foco en programación*, ofrecido como parte del programa *Nuestra Escuela* por el Instituto Nacional de Formación Docente (INFoD).

“El Programa Nacional de Formación Permanente (PNFP) Nuestra Escuela es un instrumento de formación docente federal, universal y gratuito (...) es de carácter federal y alcance universal, gratuito, de implementación gradual y progresiva, con propuestas acordes a las distintas responsabilidades institucionales y de los puestos de trabajo. El programa reconoce el derecho a la formación permanente del colectivo docente, sus propuestas serán acreditables y, según su carga horaria, tendrán asignación de puntaje para hacer efectivo el progreso en la carrera docente.”<sup>1</sup>

La oferta era completamente a distancia, con una carga horaria de “210 horas reloj, 190 horas asincrónicas y 20 horas sincrónicas obligatorias para las actividades de reflexión y puesta en común y presentación del trabajo final integrador de la actualización.”<sup>2</sup> El público objetivo estaba conformado por: “Docentes que ejercen en establecimientos educativos de nivel secundario en espacios curriculares de educación tecnológica, tecnología, informática o afines a las Ciencias de la Computación en sus distintas modalidades; referentes de TIC de establecimientos educativos de nivel secundario; y docentes de la Formación Docente o Formación Técnica, del Sistema Educativo Argentino.”<sup>3</sup> La actualización estaba conformada por cinco módulos, cada uno con una carga horaria de 40hs, de las cuales el 10% se dedicaba para dos encuentros virtuales sincrónicos de 2hs. El plan de estudios estaba compuesto por tres módulos dedicados a la programación y su enseñanza (*Estrategias para la enseñanza de la programación I, II y III*), uno dedicado a precisar epistemológicamente las Ciencias de la Computación y la importancia de su abordaje en la escolaridad obligatoria, introduciendo la relación de los saberes computacionales con el ejercicio de una ciudadanía crítica y plena (*Marco para la enseñanza de las Ciencias de la Computación en la educación obligatoria*) y por el módulo objeto del presente estudio

<sup>1</sup> <https://www.conectarigualdad.edu.ar/postulados-infod>

<sup>2</sup> Acta de dictamen, CFE-ME No 2698/2022

<sup>3</sup> <https://red.infod.edu.ar/actualizacion-academica-en-enseñanza-de-educacion-tecnologica-con-foco-en-programacion/>

(*La ciudadanía en un mundo atravesado por computadoras. Hacia una ciudadanía digital informada*). Este módulo tenía un doble objetivo: por un lado, abordar nociones disciplinares que permitieran adentrarse en debates ciudadanos para construir algunas nociones centrales de la Computación y, por otro, ilustrar activamente el postulado del módulo *Marco para la enseñanza de las CC en la educación obligatoria* que relacionaba Ciudadanía y Computación. Algunos de sus contenidos mínimos fueron:

“Modelo abstracto de computadora, definición general e interdependencia de hardware y software. CPU, memoria RAM, dispositivos de almacenamiento y de entrada/salida, características de cada uno e incidencia en el desempeño del sistema. Recolección y análisis de datos y nociones de aprendizaje automático. Aplicaciones de algoritmos de inteligencia artificial basados en datos y problemas éticos asociados.”

Cada clase, prevista para completarse en 10hs reloj de trabajo asincrónico a lo largo de dos semanas y con un encuentro sincrónico virtual mensual, consistía en la lectura de material intercalado con actividades disparadoras y la elaboración de una actividad final con entrega obligatoria.

### 3.2. Clases y actividades

La clase *¿Eso también es una computadora?* [12] tenía como primer objetivo ampliar la noción de computadora más allá de las computadoras de uso personal a otros dispositivos computacionales (como celulares pero también dispositivos automatizados) para poner de manifiesto la enorme presencia de estos dispositivos en el mundo contemporáneo. Esto derivaba en la presentación de los componentes fundamentales de todas las computadoras (procesador, memoria RAM, dispositivos de entrada y salida) y, en consecuencia, la identificación de aquellos que solo forman parte de alguna configuración en particular (como el teclado, el monitor o la pantalla táctil). Como actividad integradora, las y los participantes debían elegir componentes para construir un celular ideal, teniendo en cuenta necesidades y restricciones que imponían un perfil de uso particular (por ejemplo, un/a investigador/a de campo o alguien que trabaja con las redes sociales). Esto motivaba la presentación de algunas especificaciones técnicas de los componentes y, sobre todo, relativizar la idea de que el mejor dispositivo es el del último modelo, el más vistoso o el que tiene los componentes con especificaciones más altas (aún desconociendo qué mide cada una). Para oponer a esta idea, se presentaban algunas especificaciones técnicas generales (como la frecuencia del procesador o la capacidad de la memoria RAM) en relación con el desempeño del dispositivo, en un contexto restringido (por el tamaño del celular y la duración de la batería) que requería que las y los cursantes priorizaran los componentes y argumentaran el criterio con el que lo hacían. Esta actividad es una adaptación del proyecto *¿Existe la mejor computadora?* [15].

En la clase *¿Qué tan inteligente es la inteligencia artificial?* [13], se acompañaba a las y los cursantes a experimentar con un modelo de clasificación de imágenes en una plataforma en línea para la enseñanza de aprendizaje automático (TeachableMachine<sup>4</sup>), con el propósito de que conocieran, además de la herramienta, algunas nociones fundamentales (clases, conjunto de entrenamiento, entrenamiento, puesta a prueba), pudieran asociar el comportamiento exhibido por el modelo final con los datos que se usaron para entrenarlo y, en particular, reconocieran la noción de sesgo como un problema particularmente preocupante de estos sistemas. Esta mirada se profundizaba con el análisis de “casos en los que sesgos en sistemas de IA terminaron teniendo consecuencias graves en el plano humano.” [13]. Por ejemplo, el desarrollo de un clasificador de postulaciones laborales que descartaba sistemáticamente las enviadas por mujeres, la aplicación de un modelo de riesgo que resultó en la quita de asistencia a familias inmigrantes o un detector de sueño que impedía que las personas con ojos rasgados enciendan el auto. Como actividad integradora de entrega obligatoria de esta clase, debían seleccionar uno de los casos y analizarlo para identificar la motivación por la cual se había construido el sistema y cuál era la supuesta ventaja que traería, cuál era el sesgo que había finalmente exhibido el modelo y, a partir de ahí, especular con qué datos habría sido entrenado y si sería posible arreglar este sesgo (y si sí, cómo, si no, por qué). Esta actividad es una adaptación de las secuencias didácticas “*¿Qué necesita la IA para ser IA?*” [16], “*Un modelo de aprendizaje automático por dentro: entre datos y opiniones*” [17] y “*Hay una vaca en mi departamento: ¿entrené bien mi modelo? Errores y sesgos en el aprendizaje automático*” [18] elaboradas por los autores para la Iniciativa Program.AR

El trabajo final [14] proponía pensar explícitamente el cruce entre computación y sociedad. Consistía en la elaboración del trabajo final integrador del módulo en el que se presentaban nuevos casos asociados a los temas abordados en las otras clases y sus consecuencias en el plano humano. Estos casos se organizaban en ejes y para cada uno, se presentaba una serie de artículos o materiales de divulgación para que las y los cursantes profundizaran en él. Los títulos de los ejes eran los siguientes:

- ¿Cada cuánto hay que cambiar el celular? - Obsolescencia programada, derecho a reparar, expectativas de consumo y extractivismo.
- ¿Dónde está el negocio de las plataformas gratuitas de internet? - Recolección y extractivismo de datos
- ¿Inteligencia artificial para todo? - Ética e IA

Para reforzar la mirada técnica y contextualizada, la consigna del trabajo final proponía elaborar un breve ensayo a partir de analizar uno de los ejes. Indicaba, explícitamente, tener en cuenta: “Tensiones entre el impacto positivo y negativo sobre la sociedad y el mundo (impacto ambiental, autonomía o soberanía/dependencia, consecuencias sociales, igualdad/desigualdad, perpetuación de hegemomías, inclusión/discriminación); sus principales responsables, nuestro rol

<sup>4</sup> <https://teachablemachine.withgoogle.com/>

en ese impacto como ciudadanas/os y usuarias/os para repensar consumos y usos que hacemos de los dispositivos y artefactos computacionales. Responsabilidades de las grandes corporaciones, responsabilidad de los estados y organismos nacionales e internacionales de regulación.; ¿Existe una solución para el problema? ¿De quién depende?”. [14]

## 4. 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 módulo a partir de los datos demográficos de las y los docentes que participaron del estudio. Luego, describimos las herramientas de recolección de datos utilizados.

### 4.1. Participantes de la muestra

A continuación presentamos datos demográficos para caracterizar a la población de docentes que participó del estudio. Se trata de 79 de las 94 personas que completaron el pre y el post test, de las cuales contamos con información demográfica adicional.

En cuanto a la distribución geográfica, las y los participantes del estudio están distribuidos en 17 de las 24 las jurisdicciones del país, como se ve en la Figura 1. Merece mención especial la ausencia de participantes de C.A.B.A ya que institucionalmente no participó del programa de formación. En cuanto al género, el 38% de las personas se reconocieron como de género masculino, 62% como femenino y ninguna como “Otro”.

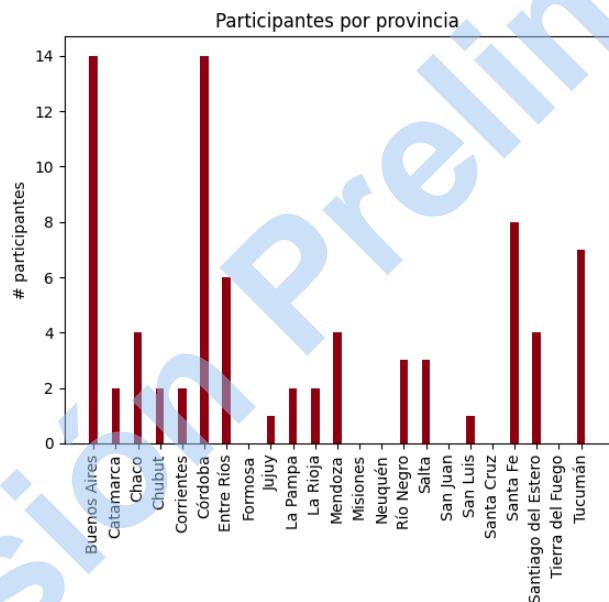
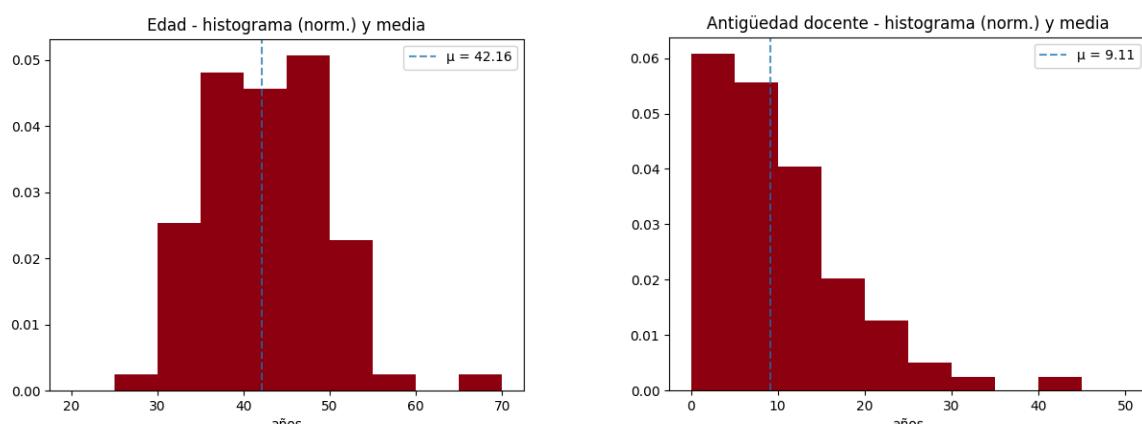


Figura 1. cantidad de participantes del estudio por provincia. Se excluye CABA por no participar de la formación.

Se indagó también en el ejercicio docente de las y los participantes. 93% ejerce la docencia y lo hace, en promedio, hace 9.11 años. La edad promedio de esta población es 42.16 años. Las figuras 2 y 3 muestran esta distribución.



Figuras 2 (izq.) y 3 (der.). Edad y antigüedad docente de las y los participantes.

De quienes ejercen la docencia, el 23% lo hace en el nivel *Primario*, el 76% en el nivel *Secundario* y 26% en el nivel *Superior no universitario*; el 42% ejerce en más de un nivel. Además, de los y las participantes en ejercicio, 95% son *docentes a cargo del aula*, de los cuales el 79% indicó hacerlo materias relacionadas a la Informática y las TIC y, más específicamente, un 24% indicó que se trata de materias de programación.

## 4.2. Herramientas de recolección

Para conocer el impacto del módulo en las y los docentes utilizamos encuestas como herramientas de recolección de datos. Una de las encuestas fue diseñada para ser completada dos veces por las y los docentes: como pre test (antes de comenzar el módulo) y como pos test (una vez finalizado el módulo). Denominaremos a esta encuesta como *pre-pos test*. Con esta herramienta intentamos capturar el impacto del módulo en las opiniones de las y los docentes con respecto a las hipótesis planteadas en la *Introducción* acerca de los componentes fundamentales y accesorios de las computadoras, la neutralidad y las limitaciones de las aplicaciones de IA, las estrategias comerciales de las empresas de tecnología y la resolución de cualquier problema por medio de la IA. El pre-pos test está compuesto por 15 afirmaciones vinculadas a estos temas que fueron abordados durante el desarrollo del módulo, para las que las y los docentes debían indicar su grado de acuerdo utilizando una escala Likert de 5 puntos – de 1 (nada de acuerdo) a 5 (muy de acuerdo) –. Para corresponder los pre y pos test las y los cursantes completaron ambos tests con su DNI.

En la Tabla 1 presentamos las afirmaciones con la etiqueta que utilizaremos en el resto del artículo para referenciarlas. Además, asociamos cada una de las hipótesis presentadas en la *Introducción* con las afirmaciones correspondientes.

Tabla 1: Asociación entre hipótesis y afirmaciones del formulario.

Hipótesis	Etiqueta	Afirmación
H1	ARQ_CPU	Si un dispositivo es una computadora entonces es necesario que tenga CPU
	ARQ_RAM	Si un dispositivo es una computadora entonces es necesario que tenga memoria RAM
H2	ARQ_TECL	Si un dispositivo es una computadora entonces es necesario que tenga teclado
	ARQ_MONI	Si un dispositivo es una computadora entonces es necesario que tenga monitor o pantalla
	ARQ_GABI	Si un dispositivo es una computadora entonces es necesario que tenga gabinete
H3	ARQ_ALMAC	La memoria RAM y otros dispositivos de almacenamiento como disco duro, pendrives o cds, cumplen la misma función en una computadora.
H4	ARQ_COMPRA	Al momento de comprar una nueva computadora conviene elegir los componentes más potentes
H5	IA_PERS	Las decisiones o recomendaciones que realiza la Inteligencia Artificial no dependen de las personas
H6	IA_DECID	La IA puede usarse para tomar cualquier tipo de decisión
H7	IA_ERROR	La aplicaciones que utilizan IA no cometan errores
H8	IA_OPINION	La IA, como está implementada en una computadora, no tiene opinión y por lo tanto no puede discriminar ni prejuzgar.
H9	IA_MEJOR	La IA resuelve los problemas mejor de que lo harían las personas
H10	CD_OBSOL	Se renuevan los dispositivos muy seguido porque los desarrollos tecnológicos cambian todo el tiempo.
H11	CD_DATOS	Los datos que almacena Google cuando utilizamos sus servicios gratuitos les dan mas ganancia de lo recaudado por sus servicios pagos
H12	CD_ETIA	Esta bien limitar qué cosas pueden realizar los sistemas basados en IA y cuáles deben ser sus alcances.

## 5. Resultados

En esta sección presentamos los resultados obtenidos del procesamiento de la prueba de pre-pos test.

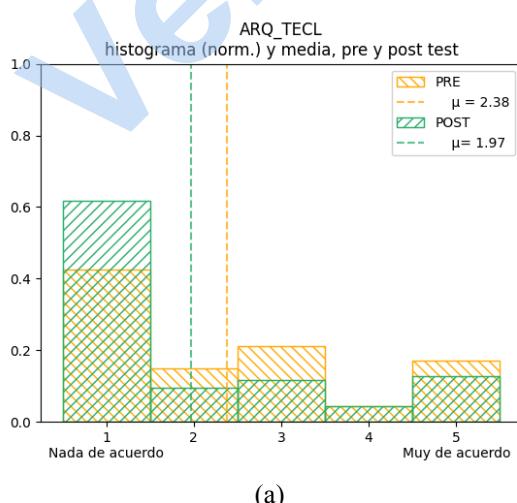
La población se compone de 94 participantes que completaron ambos cuestionarios. Se compararon las medias de las respuestas para cada afirmación en el pre y en el pos test y se evaluó la representatividad estadística mediante un T-Test apareado, con la hipótesis alternativa de que la media del pos test era mayor (o menor, dependiendo de la afirmación en cuestión) que en el pre test. Los resultados se presentan en la tabla 2.

*Tabla 2.* Resultados del T-Test apareado para cada una de las afirmaciones formuladas. Se presentan las medias observadas en el pre y el pos test para cada pregunta y el p-valor del T-Test apareado. Los valores que corresponden a diferencias estadísticamente significativas ( $p\text{-valor} < 5.00e-02$ ) aparecen en negrita y los que corresponden a diferencias estadísticamente extremadamente significativas ( $p\text{-valor} < 1.00e-02$ ) aparecen sombreados.

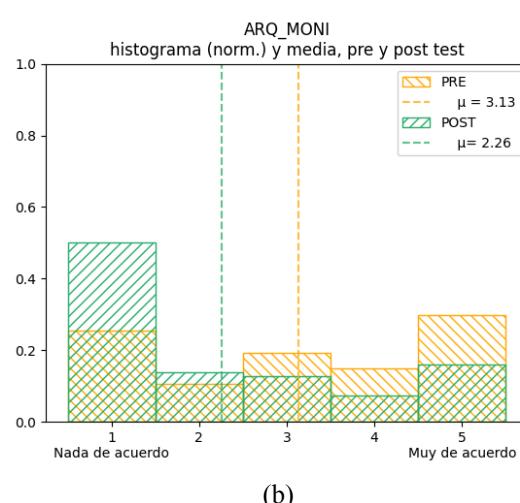
	Afirmación	Media - Pre	Media - Post	P-valor		Afirmación	Media - Pre	Media - Post	P-valor
Arq	ARQ_TECL	<b>2.38</b>	<b>1.97</b>	<b>3.04e-03</b>		IA_PERS	<b>2.60</b>	<b>2.29</b>	<b>3.62e-02</b>
	ARQ_MONI	<b>3.13</b>	<b>2.26</b>	<b>6.89e-08</b>		IA_DECID	<b>2.91</b>	<b>2.61</b>	<b>2.26e-02</b>
	ARQ_GABI	2.39	2.24	1.68e-01		IA_ERROR	<b>2.19</b>	<b>1.57</b>	<b>3.85e-06</b>
	ARQ_CPU	4.07	4.07	5.00e-01		IA_OPINION	<b>3.51</b>	<b>2.76</b>	<b>6.01e-05</b>
	ARQ_RAM	4.40	4.36	6.23e-01		IA_MEJOR	<b>2.80</b>	<b>2.18</b>	<b>1.07e-06</b>
	ARQ_ALMAC	2.30	2.05	6.71e-02		CC_CD_OBSOL	4.06	3.82	6.55e-02
	ARQ_COMPRA	<b>3.44</b>	<b>2.99</b>	<b>1.99e-04</b>		CC_CD_DATOS	<b>3.74</b>	<b>4.13</b>	<b>9.03e-04</b>
Ciud						CC_CD_ETIA	<b>4.11</b>	<b>4.52</b>	<b>2.94e-04</b>

A continuación, analizaremos la distribución de las respuestas y su variación entre pre y pos test por ejes temáticos. Para cada afirmación, presentaremos el histograma de respuestas (normalizado) y la media de la muestra; se superponen los datos para el pre y pos test con color y trazo diferenciado.

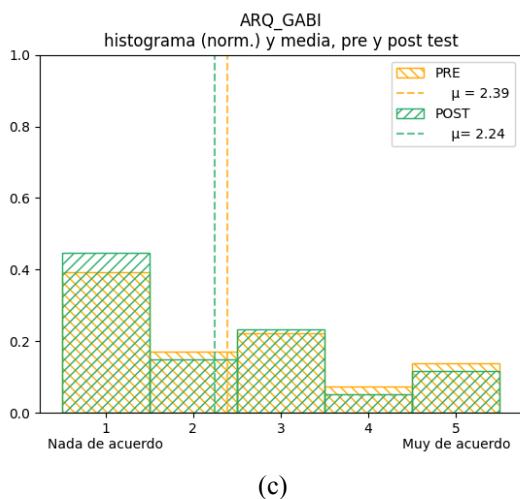
De las afirmaciones que indagaban sobre el conocimiento acerca de componentes básicos de una computadora y de la existencia de dispositivos computacionales además de las computadoras personales, las mayores diferencias (y las únicas significativas) son las correspondientes a las afirmaciones ARQ\_TECL y ARQ\_MONI (que expresan que una computadora necesariamente debe tener teclado o monitor, resp.). El test apareado muestra que existe una diferencia estadísticamente extremadamente significativa para estas afirmaciones entre el pre y el pos test. En el caso de ARQ\_GABI, si bien no se observa una diferencia significativa, la media en el pos test expresa un nivel de acuerdo similar al de ARQ\_MONI. Estos resultados permiten rechazar la hipótesis H2 (nula) en favor de la alternativa: el módulo impactó de forma extremadamente significativa en la opinión de las y los docentes sobre qué todas las computadoras tienen los mismos componentes que una computadora personal. Esto se muestra en las figuras 4.a, 4.b y 4.c.



(a)



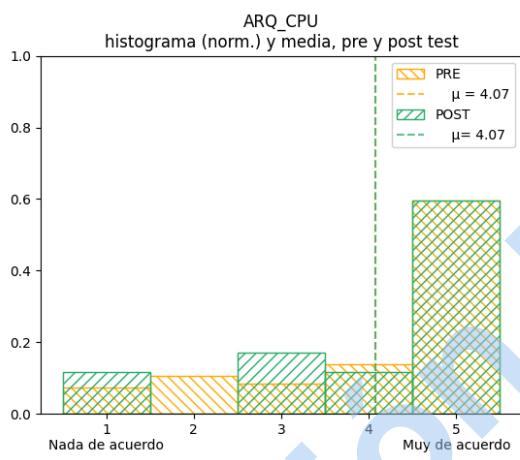
(b)



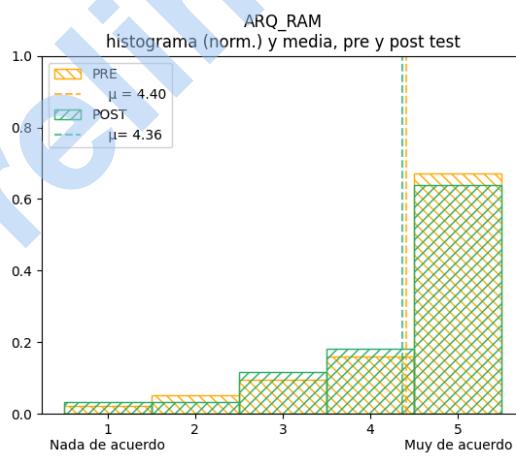
(c)

Figura 4. Distribución de las respuestas del pre y pos test para afirmaciones asociadas a la hipótesis H2.

Para las afirmaciones ARQ\_CPU y ARQ\_RAM no se observan diferencias significativas. En base a los valores observados podemos decir que la H1 (nula) se cumple. Observando los gráficos de las figuras 5.a y 5.b, se ve que la opinión de las y los docentes ya está bastante informada antes de comenzar el módulo en cuanto a la importancia del CPU y la memoria RAM para el funcionamiento de una computadora.



(a)



(b)

Figura 5: Distribución de las respuestas de pre y pos test para afirmaciones asociadas a la hipótesis H1.

Para la hipótesis H3, que evalúa si las y los docentes diferenciaron las funciones de la memoria RAM de los dispositivos de almacenamiento, no se observa un impacto significativo, por lo que la H3(nula) se cumple. Sin embargo, el nivel de acuerdo expresado tanto en el pre como en el pos test, permite inferir que las y los docentes son capaces de diferenciar las funciones de estos dispositivos de antemano (ver figura 6.a).

Otro propósito era generar una mirada más crítica sobre la evaluación y elección de dispositivos que valore las especificaciones técnicas en relación con el uso esperado más que la novedad o el marketing. A esto apuntaba la hipótesis H4 y la correspondiente afirmación ARQ\_COMPRA (que afirmaba que siempre era necesario comprar los dispositivos “más potentes”), para la que se observa una diferencia extremadamente significativa, lo que permite rechazar la hipótesis a favor de la alternativa (ver figura 6.b).

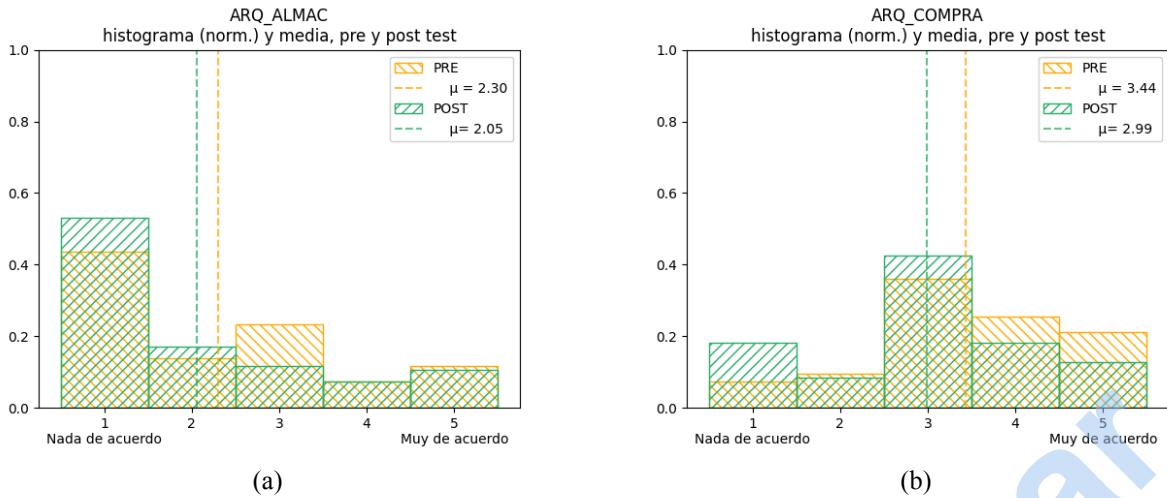


Figura 6. Distribución de las respuestas de pre y pos test para las afirmaciones asociadas a las hipótesis H3 (fig. a) y H4 (fig. b).

El otro eje temático del módulo abordado en el estudio es el de inteligencia artificial. El test apareado muestra que existe una diferencia estadísticamente significativa en la afirmación IA\_PERS entre el pre y el pos test. Estos resultados permiten rechazar la hipótesis H5 (nula) en favor de la alternativa. Lo mismo sucede con la afirmación IA\_DECID y la hipótesis H6 (nula). La figura 7 muestra las distribuciones de estas respuestas.

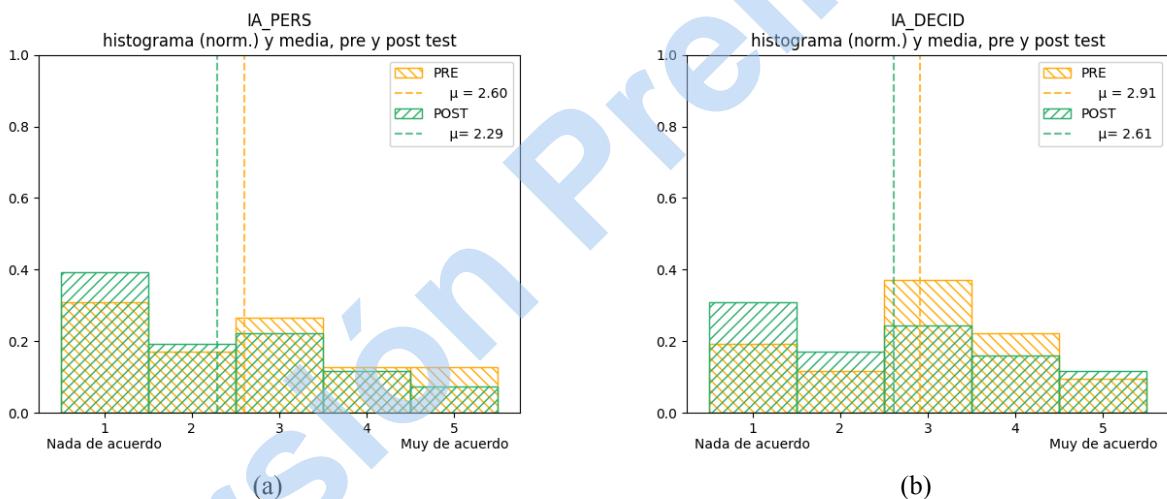


Figura 7. Distribución de las respuestas de pre y pos test para las afirmaciones asociadas a las hipótesis H5 (fig. a) y H6 (fig. b).

Además, el test apareado muestra que existe una diferencia estadísticamente extremadamente significativa para las afirmaciones IA\_ERROR, IA\_OPINION e IA\_MEJOR, lo cual permite rechazar las hipótesis H7 (nula), H8 (nula) y H9 (nula) a favor de sus alternativas. Las distribuciones para estas respuestas aparecen en la Figura 8.

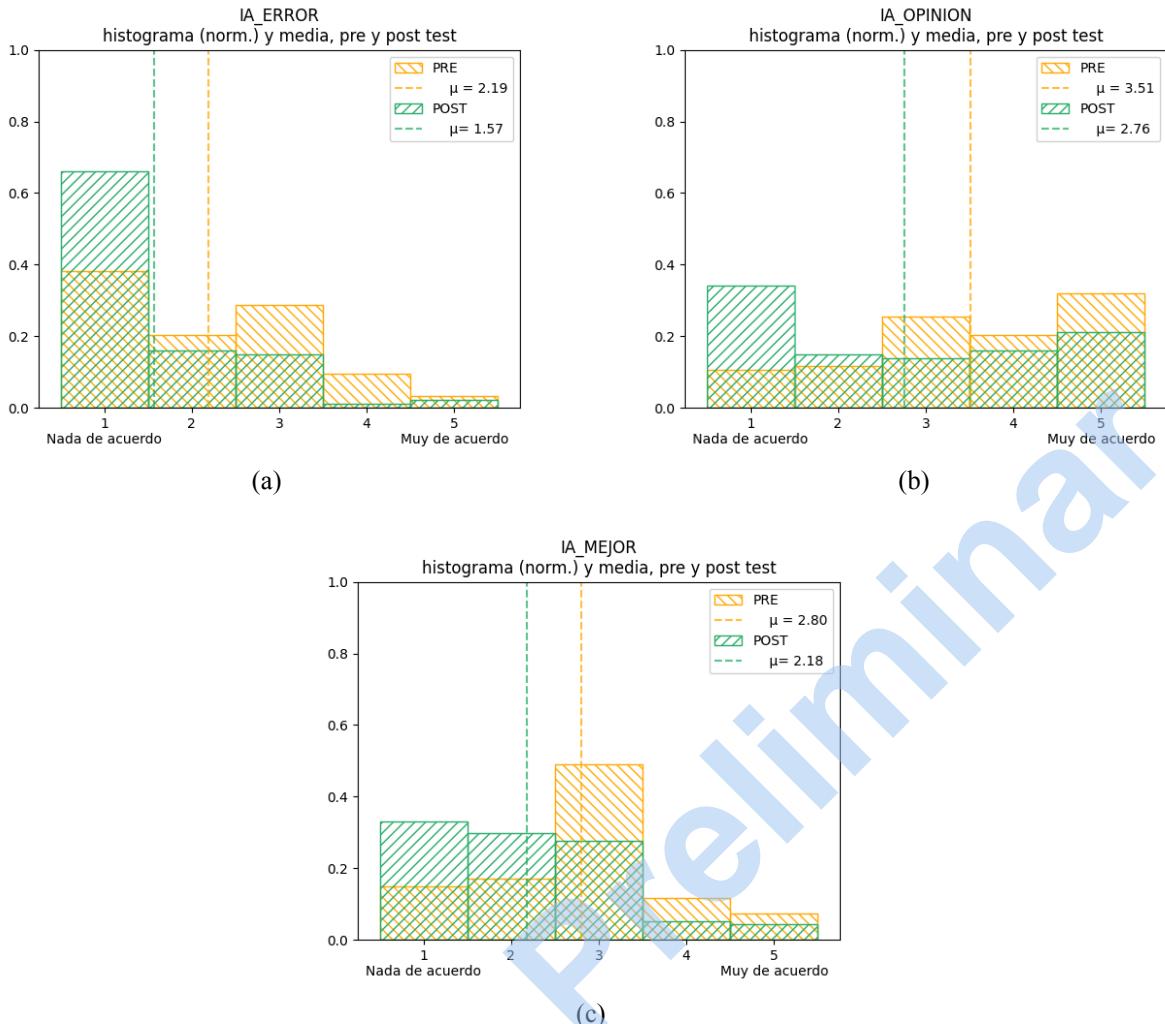
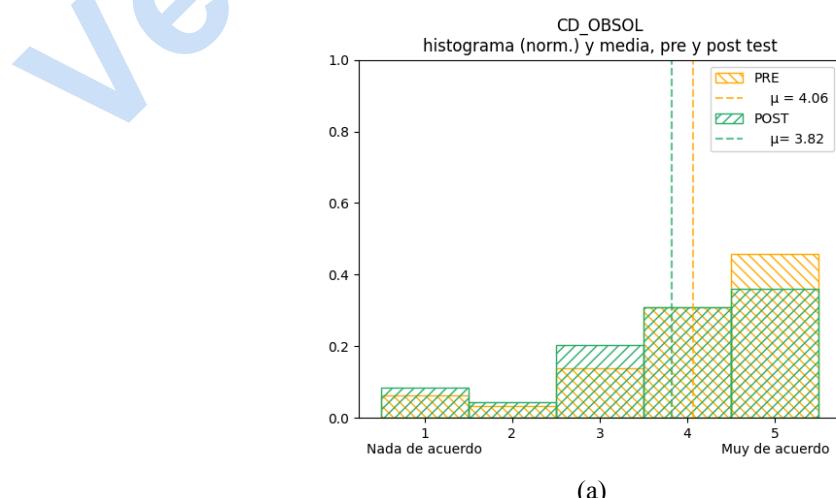


Figura 8. Distribución de las respuestas de pre y pos test para las afirmaciones asociadas a las hipótesis H7 (fig. a), H8 (fig. b) y H9 (fig. c).

Las hipótesis H10, H11 y H12 buscaban evaluar el impacto del trabajo final, reforzando la relación entre Computación y Ciudadanía. Para la afirmación CD\_OBSOL no se observó diferencia significativa, razón por la cual no podemos rechazar la hipótesis H10(nula) (ver figura 9.a). En cambio, las afirmaciones CD\_DATOS y CD\_ETIA mostraron diferencias estadísticamente extremadamente significativas lo que permite rechazar las hipótesis H11 (nula) y H12 (nula), respectivamente. La distribución de las respuestas muestran que el curso refuerza las opiniones previas, concentrando la distribución hacia el lado “Muy de acuerdo”, como se ve en las figuras 9.b y 9.c.



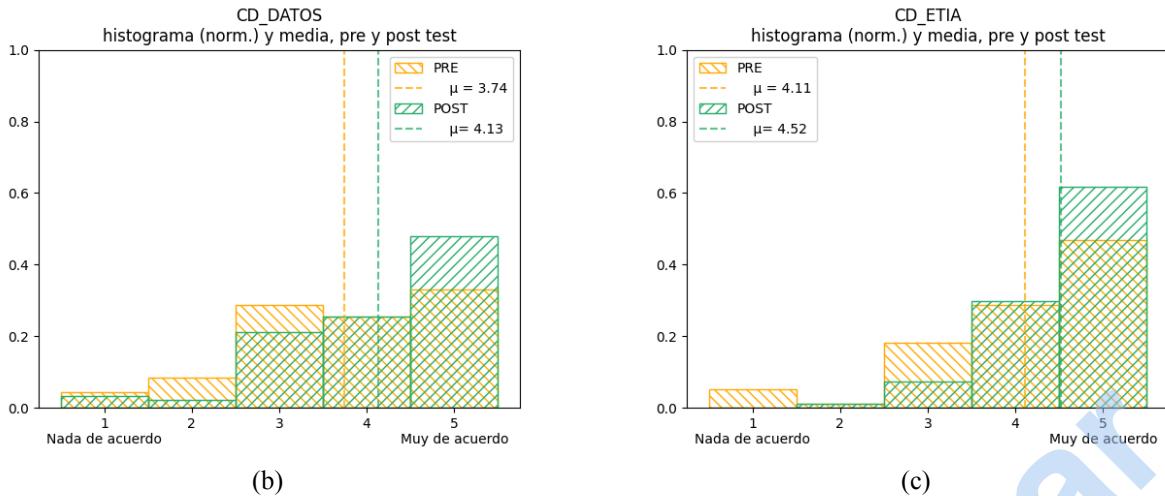


Figura 9: Distribución de las respuestas de pre y pos test para las afirmaciones asociadas a las hipótesis H10 (fig. a), H11 (fig. b) y H12 (fig. c).

## 6. Conclusiones y trabajo futuro

A partir de los resultados obtenidos en el pre y pos test, y presentados en la sección *Resultados*, observamos que la formación tuvo un impacto en las concepciones y opiniones de las y los docentes en relación a determinados contenidos del módulo. Cada una de las hipótesis planteadas en la sección *Introducción* fueron vinculadas con afirmaciones evaluadas utilizando el pre pos test.

En particular, para las hipótesis-afirmaciones H2-ARQ\_TECL, H2-ARQ\_MONI (vinculadas a las computadoras), H7-IA\_ERROR, H8-IA\_OPINION, H9-IA\_MEJOR (vinculadas a la inteligencia artificial) H11-CD\_DATOS y H12-CD\_ETIA (vinculadas a aspectos de ciudadanía digital) encontramos diferencias estadísticamente extremadamente significativas al aplicar el T-Test apareado, que permiten rechazar las hipótesis nulas a favor de las alternativas. Estos resultados ponen en evidencia un impacto de la formación en las opiniones de las y los docentes quienes, luego de la cursada, expresan reconocer: una más amplia variedad de computadoras; la no neutralidad, la falibilidad de la IA; y el valor que representan los datos que generamos al usar las plataformas de uso gratuito.

Respecto de las afirmaciones H5-IA\_PERS y H6-IA\_DECID, encontramos una diferencia estadísticamente significativa en los resultados. Esto aparece reflejado en la distribución de las respuestas, aunque de manera más moderada que para las afirmaciones anteriores: disminuyen las respuestas que expresan un alto nivel de acuerdo en el extremo no esperado y las que expresan duda. El análisis estadístico indica evidencia de un impacto en las concepciones de las y los docentes, que después del curso ponen de relevancia la participación de las personas en los desarrollos de IA y relativizan el uso de estos sistemas para la toma de decisiones.

Para el resto de las afirmaciones no se observan diferencias estadísticamente significativas pero podemos mencionar que para las hipótesis-afirmaciones H1-ARQ\_CPU, H1-ARQ\_RAM, H2-ARQ\_GABI, H3-ARQ\_ALMAC y H10-CD\_OBSOL las concepciones previas observadas en el pre test ya expresaban el nivel de acuerdo esperado para la afirmación.

Como trabajo futuro, teniendo en cuenta los resultados alentadores observados con respecto a la opinión de las y los docentes, podría analizarse específicamente la incorporación conceptual de los contenidos mínimos del curso, a partir de recuperar las respuestas a las actividades y ejercicios que se fueron desarrollando.

## Referencias

- [1] Dabbah, J., Fleitas, D., Garzón, M., Gómez, M., Martínez, M.C., Martínez López, P. E. & Peretti, G. (2024), “Propuesta curricular para la inclusión de las Ciencias de la Computación en la educación obligatoria de la Argentina”. Fundación Sadosky, Buenos Aires. Versión digital: <https://curriculum.program.ar/>
- [2] D. Thompson, T. Bell, “Adoption of new computer science high school standards by New Zealand teachers.”, in proceedings of the 8th Workshop in Primary and Secondary Computing Education, 2013.
- [3] N. Ragonis, O. Hazzan, J. Gal-Ezer, “A survey of computer science teacher preparation programs in Israel tells us: Computer science deserves a designated high school teacher preparation!”, in Proceedings of the 41st ACM technical symposium on Computer science education, 2010, pp. 401-405.
- [4] B. Ericson, M. Guzdial, M. Biggers, “Improving secondary CS education: progress and problems”, in Proceedings of the 38th SIGCSE technical symposium on Computer science education, 2007, pp. 298–301

- [5] A. Mühling, P. Hubwieser, T. Brinda, "Exploring teachers' attitudes towards object oriented modeling and programming in secondary schools", in Proceedings of the Sixth international workshop on Computing education research (ICER '10). Association for Computing Machinery, New York, NY, USA, 2010, pp. 59–68. <https://doi.org/10.1145/1839594.1839606>
- [6] C. Martinez, M. J. Gomez, M. Moresi, L. Benotti, "Lessons Learned on Computer Science Teachers Professional Development", in Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '16). Association for Computing Machinery, New York, NY, USA, 2016, pp. 77–82. <https://doi.org/10.1145/2899415.2899460>
- [7] M. Scasso, V. Marino, A. Colobini, G. Bortolotto, "Evaluación de la iniciativa Program.AR. - Cursos de Didáctica de la programación y talleres en escuelas secundarias", Buenos Aires, Fundación Sadosky, 2019, disponible en: <https://program.ar/investigaciones/>
- [8] G. Yansen, C. Cossio Mercado, A. Dolcemáscolo, P. Turjanski, L. Dughera. "La Programación y su Didáctica: resultados preliminares del seguimiento del curso en el Departamento de Computación de la FCEN-UBA", in II Simposio Argentino de Educación en Informática (SAEI 2020)-JAIIO 49, 2020.
- [9] M .Scasso, D. Cura, V. Marino, L. Kaplan, "Especializaciones en didáctica de las ciencias de la computación. Evaluación de procesos y resultados", Buenos Aires, Fundación Sadosky, 2020, disponible en: <https://program.ar/investigaciones/>.
- [10] A. Casali, N. Monjelat, P. San Martín, D. Zanarini, "Primary Level Teachers Training in Computer Science: Experience in the Argentine Context", in Computer Science – CACIC 2019. CACIC 2019. Communications in Computer and Information Science, vol 1184. Springer, Cham, 2020. [https://doi.org/10.1007/978-3-030-48325-8\\_25](https://doi.org/10.1007/978-3-030-48325-8_25)
- [11] D. Cura, H. Czemerinski, V. A. Marino, M. G. Scasso, F. Schapachnik, "A teacher training program in argentina analysed by profiles", in Proceedings of the 15th Workshop on Primary and Secondary Computing Education (WiPSCE '20). Association for Computing Machinery, New York, NY, USA, Article 24, 2020, pp. 1–2. <https://doi.org/10.1145/3421590.3421664>
- [12] Fundación Sadosky (2022). Clase 1: ¿Eso también es una computadora?. La ciudadanía en un mundo atravesado por computadoras. Hacia una ciudadanía digital informada. Actualización Académica en Enseñanza de Educación Tecnológica con foco en programación. Buenos Aires: Ministerio de Educación de la Nación.
- [13] Fundación Sadosky (2022). Clase 3: ¿Qué tan inteligente es la inteligencia artificial? La ciudadanía en un mundo atravesado por computadoras. Hacia una ciudadanía digital informada. Actualización Académica en Enseñanza de Educación Tecnológica con foco en programación. Buenos Aires: Ministerio de Educación de la Nación.
- [14] Fundación Sadosky (2022). Clase 4: la ciudadanía en un mundo atravesado por computadoras. Trabajo final. Actualización Académica en Enseñanza de Educación Tecnológica con foco en programación. Buenos Aires: Ministerio de Educación de la Nación.
- [15] UNICEF, PlaNEA: Proyecto 2, 1er año – Primer ciclo nivel secundario, Buenos Aires, julio 2020.
- [16] Fundación Sadosky (2023), "¿Qué necesita la IA para ser IA?", disponible en: <https://curriculum.program.ar>
- [17] Fundación Sadosky (2023), Un modelo de aprendizaje automático por dentro: entre datos y opiniones, disponible en: <https://curriculum.program.ar>
- [18] Fundación Sadosky (2024), Hay una vaca en mi departamento: ¿entrené bien mi modelo? Errores y sesgos en el aprendizaje automático, disponible en: <https://curriculum.program.ar>
- [19] C. Llambí, M. Borchardt, V. Klinkovich (Coords), Aprendizajes y desafíos para la enseñanza de las Ciencias de la Computación en las escuelas. La iniciativa Program.AR de Argentina. CAF - banco de desarrollo de América Latina y el Caribe - y Fundación Sadosky. 2023.

# Criterios de evaluación en las materias de programación de las carreras del Módulo de Informática de la FCEQyN UNaM

Daniela Alejandra Martínez<sup>1</sup>

<sup>1</sup> Universidad Nacional de Misiones, Apóstoles, Argentina  
[dalejmartinez@fceqyn.unam.edu.ar](mailto:dalejmartinez@fceqyn.unam.edu.ar)

## Resumen

Este trabajo explora concepciones sobre la evaluación en la enseñanza de la programación en el contexto universitario. El artículo inicia con un recorrido histórico sobre el desarrollo de las computadoras y los lenguajes de programación, destacando cómo esta evolución ha influido en las prácticas de enseñanza y evaluación. Se analizan diversos enfoques evaluativos, como el análisis de portafolios, entrevistas basadas en artefactos, diseño de escenarios y aprendizaje basado en proyectos, enfatizando la importancia de una evaluación integral que considere tanto el proceso como el resultado final. El trabajo plantea interrogantes sobre cómo los docentes construyen las instancias de evaluación, qué fundamentos utilizan para definir los criterios, y cómo estos se operacionalizan en la práctica. Se destaca la necesidad de considerar las rationalidades específicas del campo de la programación en la configuración de las prácticas evaluativas. El estudio propone una investigación futura para caracterizar los criterios de evaluación en las materias de programación de la FCEQyN, buscando mejorar los procesos de enseñanza y evaluación en este campo.

**Palabras clave:** Programación; Evaluación; Ciencias de la computación; Criterios de evaluación.

## 1. Introducción

En las últimas décadas, las ciencias de la computación han tenido un impacto determinante en el desarrollo de la denominada sociedad del conocimiento, y paralelamente, emergieron propuestas de formación afines al campo en todos los niveles educativos, marcando la importancia de contar con sujetos capacitados y profesionales calificados en el área.

Particularmente en el ámbito universitario se han creado diferentes carreras a lo largo de todo el país, con diversos perfiles de egreso. La formación de técnicos, analistas, licenciados, ingenieros y profesores en computación incorpora trayectos en programación, siendo relevante comprender cómo las características de la disciplina y el contexto configuran las propuestas de enseñanza y prácticas evaluativas particulares.

La evaluación en la educación superior es un proceso fundamental para medir el aprendizaje de los estudiantes y garantizar la calidad de la enseñanza. Los docentes universitarios evalúan a sus estudiantes desde una didáctica específica que constituye su objeto de estudio e interés, marcando con relación al proceso (teorías, métodos) y a las dimensiones de evaluación (eficiencia, eficacia, relevancia, equidad) aquellos enfoques que consideran más relevantes y oportunos a las circunstancias [1].

Según Giannattasio et al. [1], la evolución histórica de la evaluación de estudiantes se encuentra ligada a los cambios en los criterios empleados. Se enfatiza que los criterios son transitorios y que los elementos de continuidad son el contexto, el evaluado y el evaluador, siendo la experiencia de este último determinante en la creación de criterios de evaluación.

Teniendo en cuenta lo anterior, se entiende que, en las carreras de informática, los criterios de evaluación en programación se organizan en función de las rationalidades de las ciencias de la computación, los paradigmas de programación, el lenguaje de programación estudiado, el perfil de la propuesta educativa y la experiencia del enseñante.

Algunos autores han descrito las competencias necesarias en un profesional de la programación. Hamming [2] enfatiza que por cada estudiante en computación, de nivel superior, se debería priorizar las prácticas de laboratorio donde puedan

realizar tareas de diseño, construcción y documentación de programas. Estas prácticas permiten evaluar el desempeño del estudiante en función del estilo de programación, la eficacia práctica, la ausencia de errores y la documentación. “A la hora de juzgar su trabajo, debemos distinguir claramente entre la astucia superficial y la comprensión genuina. La inteligencia era esencial en el pasado, pero ya no es suficiente” [2, p.6].

Por otro lado, Brennan y Resnick [3], en un estudio donde desarrollan un framework para evaluar el desarrollo del pensamiento computacional en programación con Scratch, reiteran que las estrategias utilizadas buscan propiciar el pensamiento computacional, la resolución de problemas, la creatividad y autonomía en los estudiantes.

Evaluar competencias a partir de artefactos producidos por los estudiantes, donde la construcción de saberes es andamiaada desde la propia práctica de codificación, evidencia cierta complejidad en las instancias de evaluación en materias de programación, donde la generación de dispositivos propios, véase por ejemplo el caso local PRENDER[4], evidencia el valor de determinar en determinado contexto cuáles son los criterios de evaluación de la programación y cómo se construyen estos criterios.

Teniendo en cuenta lo anterior, se considera de gran relevancia elaborar un trabajo de investigación que permita comenzar a explorar esta temática. Dicha investigación se plantea en el contexto del Módulo de Informática de la Facultad de Ciencias Exactas, Químicas y Naturales (FCEQyN) de la Universidad Nacional de Misiones (UNaM), se cuenta con tres carreras, cada una con un perfil profesional diferente, que incluyen diversas materias abocadas a la enseñanza de la programación. Se destaca que, al ser la computación un campo de reciente formalización en el nivel superior, y que el Módulo de Informática es un espacio de joven conformación, no se cuentan con estudios que recopilen y analicen los criterios de evaluación utilizados en estas carreras para la enseñanza de la programación.

El trabajo de investigación buscará identificar e interpretar los criterios de evaluación utilizados por los docentes en las materias de programación de las carreras de informática del Módulo Apóstoles de la FCEQyN de la UNaM. A través de este, se pretende caracterizar estos criterios y entender el proceso mediante el cual son construidos, proporcionando una visión integral de los enfoques y consideraciones que los docentes emplean en la evaluación de sus estudiantes.

En este sentido, este trabajo presenta una aproximación al contexto conceptual de la investigación. Para esto, primeramente se considera crucial indagar sobre el origen de la programación, el devenir histórico y tendencias en prácticas evaluativas en el sistema universitario argentino. Es por esto que en este escrito se realiza el siguiente recorrido: La primera sección presenta una revisión de los orígenes de la computación, seguida del surgimiento de la programación y finalizando con la caracterización del proceso de evaluación desde las racionalidades de la didáctica específica.

## 2. Historia de las máquinas que computan

El campo de la computación tiene por componente fundamental al artificio tecnológico, la máquina que computa y permite materializar digitalmente lo que ideamos [2]. Reconociendo que el campo de la computación configura sus orígenes en el desarrollo de artefactos para el tratamiento de información.

La historia de la computación se encuentra estrechamente asociada a la historia del cómputo, ya que la búsqueda de automatización y eficiencia en el cálculo ha impulsado el desarrollo de la computación y la tecnología digital como la conocemos en la actualidad.

Se considera que las primeras herramientas de cómputo usadas fueron los “tally sticks”, el cual era un palo alargado hecho de madera, hueso u otro material, y que contiene un sistema de marcas para el registro de documentos numéricos, cantidades o incluso mensajes. Estos dispositivos tienen más de 35.000 años de antigüedad y representan una característica clave de los ordenadores: el almacenamiento y la recuperación de datos [5].

En la Antigüedad y la Edad Media, se pueden encontrar máquinas más complejas, conocidas como computadoras analógicas. Estas eran máquinas que, utilizando aspectos de los fenómenos físicos, lograban modelar el problema que se requería resolver. Una de estas es la máquina de Anticitera, de la antigua Grecia la cual se estima que fue creada aproximadamente entre 150 y 50 a.C. Se cree que este dispositivo, cumplía la función de ser un reloj astronómico que determinaba la posición de los cuerpos celestes con extraordinaria precisión. Esta máquina se considera un importante antecedente del campo, precursor temprano de los dispositivos mecánicos de cálculo, debido a su complejidad técnica y su capacidad para realizar cálculos astronómicos avanzados utilizando principios mecánicos [5].

Inicialmente las computadoras analógicas no realizaban los cálculos de forma autónoma, sino que ayudaban en el cómputo manual pero, con el paso del tiempo, estas comenzaron a ser capaces de realizar operaciones matemáticas básicas. Se identifica la creación de la primera calculadora mecánica desarrollada por Wilhelm Schickard en 1623, y en 1645, Blaise Pascal inventó la Pascaline, una calculadora mecánica capaz de realizar operaciones aritméticas simples mediante el uso de ruedas dentadas y engranajes [6].

Otro gran avance de las computadoras se produjo en 1801, cuando Joseph-Marie Jacquard diseñó y desarrolló un telar mecánico que utilizaba tarjetas perforadas para controlar automáticamente el patrón de tejido. Cada tarjeta perforada representaba una línea de tejido y contenía instrucciones codificadas que determinaban qué hilo levantar o bajar en cada paso del proceso de tejido. Este sistema permitió automatizar complejos diseños de tejido y fue una de las primeras

aplicaciones prácticas de control automático mediante tarjetas perforadas. Este telar introdujo el concepto de utilizar tarjetas perforadas como medio de almacenamiento de información y control de procesos [5].

En 1880, Herman Hollerith inventó métodos para producir tarjetas perforadas y mecanismos para que las máquinas pudieran leer los datos almacenados. Estas mejoras permitieron a las computadoras pasar de cálculos sencillos a ecuaciones diferenciales más complejas.

Históricamente, el desarrollo de computadoras más complejas fue impulsado desde diferentes campos, la ciencia, las finanzas y también la guerra. La era de las computadoras digitales comenzó alrededor de la Segunda Guerra Mundial. Durante este periodo Alan Turing, matemático inglés, contribuyó a la construcción de una de las primeras computadoras digitales, el Colossus, que se utilizó en los procesos de cifrado, siendo esta una máquina de gran tamaño y asombrosamente rápida para la época [5].

A partir de aquí se refiere a los sucesivos avances tecnológicos del ordenador como generaciones de computadoras [7], denominación que identifica diferentes épocas y rasgos en el desarrollo de la tecnología computacional, cada una caracterizada por avances significativos en hardware, software y telecomunicación:

Primera generación: entre 1937 - 1946 las computadoras sólo podían realizar una tarea y no tenían sistema operativo. Estaban construidas con electrónica de válvulas de vacío, se programaban en lenguaje máquina y utilizaban condensadores para almacenar datos.

Segunda generación: en los años 1947 - 1962 se desarrollaron componentes electrónicos más avanzados, como los transistores. Las computadoras tenían memoria, sistemas operativos y su almacenamiento se realizaba en cintas y discos, así como impresoras.

Tercera generación: entre 1963 - 1975, los circuitos electrónicos de las computadoras redujeron tamaño con la aparición de los circuitos integrados que las hicieron más potentes, fiables y capaces de ejecutar muchos programas diferentes al mismo tiempo.

Cuarta generación: entre 1975 - al presente el tamaño de los ordenadores se redimensiona a lo que llamamos Computadoras Personales, PC. Se crea el microprocesador y el chip. Esta generación se caracteriza por los avances en tecnología de microprocesadores con mayor capacidad de procesamiento, el auge de Internet y la World Wide Web.

Quinta generación: del presente al futuro, esta generación hace presencia de la Integración de tecnologías emergentes como inteligencia artificial, machine learning y computación en la nube.

Este recorrido evidencia que la evolución del cómputo persigue el mejoramiento de los componentes tangibles, con el afán de encontrar nuevas y mejores formas de representar e interactuar con la información, determinando el hardware los recursos para desarrollar un software pero, como se abordará en la siguiente sección, condicionando los lenguajes de programación qué y cómo podemos construir con el computador, factor de vital interés en las propuestas formativas del área.

### 3. El devenir histórico de la programación

Tras la introducción de las computadoras digitales, que ejecutaban diferentes programas, fue necesario encontrar una forma de programar estos dispositivos. Creándose el primer lenguaje de programación, el cual se denominó lenguaje máquina en base al sistema binario, empleando una serie de ceros y unos, que permite dar órdenes de cómputo, siendo a la actualidad el único lenguaje que estas pueden entender directamente.

En el inicio, las computadoras básicas contaban con largas filas de interruptores, cada uno de los cuales representaba un valor digital: 0 o 1. La máquina leía el estado de un interruptor a otro y procesando las instrucciones en función de si el interruptor estaba encendido o apagado. Para esto, el lenguaje máquina emplea el sistema binario formulado por Gottfried Leibniz en 1679.

Leibniz era un polímata cuyo pensamiento filosófico quedaría patente en su pensamiento binario. Él sosténía que cada acción que realizamos, ya sea una simple pregunta o un proceso de pensamiento más largo podría simplificarse en una decisión binaria, un sí o un no, pudiendo esto refinarse una y otra vez [5].

Aunque el sistema binario de Leibniz, permitía realizar ecuaciones matemáticas complejas, se comenzó a necesitar una estructura para unir las distintas operaciones binarias. En 1832, Boole tiene la idea de que el razonamiento humano podía deducirse a frases sencillas y luego combinarse junto con un conjunto de expresiones matemáticas, como "sí", "o", "no", para formar un lenguaje lógico o lenguaje del pensamiento. Si entendemos a las computadoras como máquinas lógicas que ejecutan y analizan una serie de ceros y unos (verdadero o falso), a partir de estas es posible crear frases o funciones complejas utilizando el álgebra de Boole y otros teoremas lógicos [5].

Cuando los enunciados lógicos se agrupan en instrucciones complejas que expresan una resolución en secuencia, se denominan algoritmos. Un algoritmo puede definirse como una lista de instrucciones que lleva a su usuario a una respuesta o salida determinada basándose en la información provista. Los algoritmos son componentes esenciales de todo programa informático.

La palabra algoritmo se le atribuye al persa Al-Khwarizmi, matemático de renombre que vivió entre los años 780 y 850. Pero, el primer caso registrado de uso de algoritmos se remonta a Shuruppak, antigua ciudad próxima a la actual Bagdad, donde los sumerios empleaban tablillas de arcilla, que datan aproximadamente del 2500 a.C., para ilustrar un método repetible para dividir equitativamente una cosecha de grano entre un número variable de hombres. El método descrito utilizaba pequeñas herramientas de medición; siendo útil para los vendedores de la época porque no tenían balanzas lo bastante grandes para pesar [5].

La ciencia que hay detrás de los algoritmos fue desarrollada y perfeccionada a lo largo de los siglos por grandes matemáticos como Euclides (circa 300 a.C.), el cual describió varios algoritmos para la construcción de figuras geométricas y la resolución de problemas matemáticos, Arquímedes (287-212 a.C.), que desarrolló métodos para calcular áreas y volúmenes utilizando técnicas geométricas, Leonardo Fibonacci y Pierre de Fermat desarrollaron métodos algorítmicos para resolver problemas matemáticos durante los siglos XV-XVII, Carl Friedrich Gauss, que inventó un sistema para predecir qué factores de los algoritmos son más importantes y cómo eliminar los que no lo son, entre muchos otros.

Cuando los componentes electrónicos se hicieron más eficaces y asequibles, las computadoras se volvieron más rápidas y la programación en lenguaje máquina se convirtió en un trabajo lento y difícil debido a que una serie de ceros y unos resulta complejo de entender y recordar para los humanos, por lo cual se tardaba mucho tiempo en escribir y entender incluso los algoritmos más sencillos [5]. Es por eso que surgió la necesidad de diseñar nuevos lenguajes, más adelante denominados de alto nivel, que hicieran la tarea de programar más accesible.

El primer caso de creación de lenguaje de programación “más humano” fue el lenguaje ensamblador, con el cual, en lugar de escribir ceros y unos, el programador podía escribir sencillas instrucciones lógicas, que luego se compilaban en lenguaje máquina. Este es un lenguaje sencillo y rudimentario que podría describirse como una simple receta ejecutada de arriba abajo. Fue utilizado por primera vez en 1948 en la máquina denominada EDSAC y era capaz de trabajar con aritmética de punto flotante, operaciones aritméticas y trigonométricas, vectores y matrices [6].

El rápido desarrollo de los circuitos electrónicos más eficientes permitieron la creación de computadoras capaces de multiprocesamiento, habilitando el cómputo de diferentes datos en una misma ventana de tiempo. La complejidad de codificar dicho funcionamiento en lenguaje ensamblador impulsó la necesidad de crear nuevos lenguajes de programación [5].

Estos nuevos lenguajes se construyen sobre las bases de funcionamiento de los lenguajes anteriormente mencionados, razón por los que se denominan lenguajes de alto nivel, y para que la computadora ejecute un código escrito en lenguaje de alto nivel primero lo traduce a lenguaje ensamblador y luego a lenguaje máquina.

Lovrenčić et al. [6] presentan una revisión detallada de los antecedentes de los lenguajes de alto nivel, incluyendo desarrollos significativos que abarcan desde la década de 1940 hasta finales del siglo XX. Entre ellos se destacan:

- El desarrollo de Plankalkül en 1948, creado por el científico alemán Konrad Zuse durante la Segunda Guerra Mundial, considerado el primer lenguaje de programación de orden superior. Aunque este trabajo pasó desapercibido en su época, sentó las bases para los lenguajes modernos al permitir asignaciones, subrutinas, matrices y otras características fundamentales.
- En 1954, se desarrolló FORTRAN como una alternativa a los complicados ensambladores para cálculos numéricos. FORTRAN introdujo tipos de datos enteros y de coma flotante, junto con sentencias aritméticas como IF, bucles DO y operaciones de entrada y salida. Versiones posteriores, como FORTRAN 66 y FORTRAN 77, refinaron el lenguaje y mejoraron su funcionalidad.
- LISP, desarrollado en 1958 en el M.I.T., introdujo nuevos paradigmas de programación orientados hacia la inteligencia artificial. Basado en el  $\lambda$ -cálculo de Church, LISP adoptó la notación prefija y permitió funciones recursivas y estructuras de datos basadas en listas.
- ALGOL, a partir de 1958, se diseñó para superar las deficiencias percibidas en FORTRAN, introduciendo conceptos como la programación estructurada y la gramática BNF que influyeron en lenguajes posteriores como Pascal y C.

- COBOL, surgido en 1959, se centró en la legibilidad y las capacidades de agregación de datos para aplicaciones empresariales. Introdujo un enfoque estructurado con divisiones claras en los programas.
- BASIC surgió en 1963 para simplificar la programación en comparación con lenguajes como FORTRAN y COBOL. Evolucionó incorporando subrutinas, bucles y operaciones con archivos, y con la introducción de Visual Basic en 1991, adoptó la programación orientada a eventos y conceptos de programación orientada a objetos.
- LOGO, desarrollado en 1967, fue un lenguaje de programación educativo con un enfoque procedimental y gráficos de tortuga para facilitar la visualización de conceptos. Introdujo características únicas como la recursividad, listas y llamadas a funciones basadas en la pila.
- Surge en este período el paradigma de la programación estructurada el cual introdujo estructuras de control sistemáticas y organización de datos, promoviendo la legibilidad y el diseño descendente de programas.
- Pascal, desarrollado en 1969, fue el primer lenguaje alineado con la programación estructurada, destacándose por su uso de bucles, punteros, recursividad y estructuras de datos robustas. Aunque considerado rígido en la organización de datos y secciones procedimentales, su impacto fue significativo, popularizado por el entorno integrado de desarrollo (IDE) Turbo Pascal de Borland.
- El lenguaje de programación C, desarrollado en 1969, fue creado para el sistema operativo UNIX, lo que influyó en su estructura y enfoque hacia los recursos computacionales. Se considera a C como lenguaje de bajo nivel ya que el permite acceso directo a la memoria, lo que facilita optimizaciones de código.
- La década de 1970 marcó la aparición de varios paradigmas de programación, como la programación funcional (Lisp, ML) y la programación lógica (Prolog), cada uno con enfoques distintos en la forma de describir problemas y construir soluciones.
- La programación orientada a objetos (POO) revolucionó la programación con el encapsulamiento de datos y métodos en objetos, introducido por lenguajes como Prolog y Smalltalk.
- SQL, concebido a finales de los años 60 como modelo de datos relacional, se convirtió en un lenguaje integral para la manipulación de bases de datos relacionales.
- Ada, desarrollado en la década de 1970, fue un lenguaje estructurado y orientado a objetos enfocado en la seguridad y la modularidad, pero enfrentó desafíos de adopción.
- C++, desarrollado a partir de C en 1979, expandió las capacidades de C con la adición de clases y características de programación orientada a objetos.

Estos antecedentes evidencian el desarrollo continuo en lenguajes de programación de alto nivel, denominados coloquialmente lenguajes de programación, los cuales comprenden una amplia gama de paradigmas, aplicaciones y enfoques singulares de desarrollo que estructuran el estado actual y la evolución de la programación, entendida como campo de estudio y aplicación de las ciencias de la computación.

## **La evaluación en la programación**

“Dime qué piensas de la evaluación y te diré qué tipo de profesional eres” [8, p. 10]. Esta afirmación subraya que los sentidos que atribuimos a la evaluación son un reflejo de nuestras concepciones fundamentales sobre la enseñanza, indicando también que estas interpretaciones están ligadas a nuestro modelo de sociedad y al papel de la educación en ella.

En relación, se reconoce que, a la hora de pensar en la evaluación, el principal desafío es “...construir criterios que nos permitan obtener información válida y confiable” [9, p. 30]. En este sentido, los docentes usan estos criterios como parámetros o indicadores de logros de aprendizaje, buscando que se reduzca el riesgo de que se apliquen en los exámenes apreciaciones subjetivas.

Giannattasio et al. [1] argumentan que la evolución histórica de la evaluación de estudiantes está marcada por la transformación en los criterios empleados, destacando la importancia del contexto, el evaluado y, especialmente, la experiencia del evaluador en la configuración de estos criterios. En el caso específico de la programación, estos criterios se ven además influenciados por las racionalidades propias de las ciencias de la computación, los paradigmas de programación vigentes, el lenguaje de programación utilizado y el perfil educativo de la carrera.

Hamming [2] enfatiza que en la formación de estudiantes de computación, universitarios o graduados, se debería priorizar las prácticas de laboratorio donde ellos puedan realizar tareas de diseño, construcción y documentación de programas. Estas prácticas son las que permitirían evaluar el desempeño del estudiante en función del estilo de programación, la eficacia práctica, la ausencia de errores y la documentación. Esto implica que la evaluación debe

considerar tanto el dominio técnico como la profundidad conceptual y la capacidad para resolver problemas complejos de manera creativa y crítica, haciendo hincapié en la capacidad de comprensión genuina del estudiantado como criterio fundamental de estos procesos formativos.

Teniendo en cuenta lo anterior, surge la incógnita de ¿cómo se piensan estos criterios en diversas estrategias y dispositivos de evaluación? Brennan y Resnick [3] en un estudio donde desarrollan un framework para evaluar el desarrollo del pensamiento computacional en programación con Scratch, presentan tres de los enfoques más empleados en estos espacios formativos, considerando sus fortalezas y debilidades respecto a la intención de que los educandos desarrollen el pensamiento computacional:

- Análisis del portafolio: donde los estudiantes recopilan los trabajos realizados durante todo el cursado con la finalidad de ser evaluados, planteándose una revisión sistemática del progreso del estudiante a lo largo del tiempo. El enfoque permite visualizar cómo resultaron los diversos trabajos de codificación, al contrario de un examen sumativo de un único proyecto final pero, por otro lado, se ve limitado al no recuperar la voz del estudiante, ya que se orienta a la revisión particular de cada producto sin revelar nada sobre el proceso de desarrollo de este y, a su vez, nada sobre las prácticas concretas y pensamiento computacional del estudiante durante la experiencia de construcción.
- Entrevistas basadas en artefactos construidos: instancia que implica interrogar/entrevistar al estudiante sobre sus antecedentes y conocimientos previos, las metas en la tarea de programación resuelta, su participación en comunidad y cómo contribuye lo construido a su desarrollo. Permite centrarse no solo en el código fuente sino también en la experiencia del estudiante.
- Diseño de escenarios: enfoque que emplea un conjunto de proyectos desarrollados previamente para abordar conceptos y prácticas a trabajar. Estos software son presentados a grupos de estudiantes, donde se solicita, en base al mismo, que los educandos expliquen qué hacen, comenten cómo podrían ampliarse, corrijan errores de programación y modifiquen su comportamiento añadiendo una o más funcionalidades. Se proponen estos cuatro tipos de preguntas para ofrecer la oportunidad de explorar sistemáticamente diferentes construcciones y experiencias. Adicionalmente, se reconoce que la naturaleza de las preguntas y el uso de proyectos seleccionados externamente pueden no conectar con los intereses personales y el sentido de motivación intrínseca del estudiante.

En el contexto argentino, Davila [10] destaca el uso de proyectos integradores en la evaluación, en el marco de propuestas de enseñanza centradas en el aprendizaje basado en proyectos. Este enfoque, propuesto inicialmente por Kilpatrick [11], sugiere que la educación debe centrarse en actividades significativas y prácticas que involucren directamente a los estudiantes, permitiendo evaluar su desempeño en un contexto auténtico y dinámico.

En la programación, esta concepción de proyecto permite enfatizar la evaluación formativa y continua en el contexto de proyectos significativos, permitiendo a los estudiantes demostrar su comprensión y habilidades a través de la aplicación práctica. La evaluación se centra en el proceso de desarrollo del proyecto, considerando criterios como la resolución de problemas, la colaboración y la creatividad, así como también proporcionando retroalimentación constructiva que guía el aprendizaje y mejora continua, alineándose con la naturaleza dinámica y práctica de la programación.

Por otro lado, para el caso del diseño tecnopedagógico de portafolios electrónicos de aprendizaje, Díaz Barriga A. et al. [12], recuperan la importancia de la evaluación auténtica, buscando identificar el vínculo de coherencia entre los aprendizajes de tipo conceptual con los procedimentales o competenciales que se expresan en evidencias de desempeño, en un contexto y situación determinada, reconociendo la evaluación de portafolios en la categoría de actividad crítica.

“En esta interpretación de la evaluación como actividad crítica, las normas y los criterios de evaluación no se elaboran fuera del contexto de aprendizaje ni adoptan decisiones definitivas o inalterables. Más bien de las respuestas y de los argumentos que cada uno pone en juego surgen las vías de entendimiento. Es necesaria la apertura por parte de quien enseña y de quien aprende para revisar, defender y rebatir críticamente las propias razones, los propios argumentos” [13, p. 3].

En concordancia con lo anterior, Dewey [14] argumenta que la educación debe ser concebida y practicada en una sociedad democrática, alegando que no solo debe ser un medio para transmitir conocimientos, sino que debe fomentar la participación activa y crítica de los individuos en la sociedad, promover la igualdad y la reciprocidad en las relaciones sociales, reflejando los principios democráticos en su estructura y métodos. En este sentido, importa que la educación se adapte a las necesidades y contextos de los estudiantes, haciendo que el aprendizaje sea relevante para sus vidas y comunidades.

La participación activa y crítica en el aprendizaje de la programación, y ciencias de la computación en general, es desarrollada por Papert a partir del Construcciónismo, teoría influenciada por los trabajos de Jean Piaget, la teoría de la computación y la inteligencia artificial, que se focaliza en la experiencia personal de cómo conocemos y valoramos con computadoras y a partir de la construcción de artefactos computacionales.

Bruner [15] expone dos concepciones sobre el funcionamiento de la mente: el enfoque computacional y el culturalismo. El primero considera la mente como un mecanismo que procesa información de manera organizada y regulada por reglas preexistentes, similar a un ordenador, aunque este enfoque puede no captar la naturaleza desordenada del conocimiento humano. El segundo, el culturalismo, sostiene que la mente está intrínsecamente ligada a la cultura, la cual proporciona los símbolos y herramientas necesarias para la creación y comunicación de significados. Este enfoque enfatiza la importancia del contexto cultural y la interacción social en la formación de la mente.

Es decir, para investigar los criterios de evaluación en la programación es necesario estudiar las formas de evaluar la disciplina, cómo los enseñantes significan estas prácticas en función del momento socio histórico en el que se encuentran y las características de la propuesta formativa.

Es por eso que, se considera de gran relevancia elaborar un trabajo de investigación que permita comenzar a explorar, de manera situada, esta temática en las carreras de computación del Módulo de Informática de la UNAM. Este módulo, creado el 02 de Agosto de 1993, dicta la carrera de Analista en Sistemas de Computación, carrera de pre grado que se inició con la creación de la sede, la Licenciatura en Sistemas de la Información, carrera de grado que data del 2004 y el Profesorado Universitario en Computación, carrera de grado creada en 2015. Estas tres carreras, cada una con un perfil profesional diferente, incluyen diversas materias abocadas a la enseñanza de la programación. Se destaca que, al ser la computación un campo de reciente formalización en el nivel superior, y que el Módulo de Informática es un espacio de joven conformación, no se cuentan con estudios que recopilen y analicen los criterios de evaluación utilizados en estas carreras para la enseñanza de la programación.

## 4. Conclusiones

En este artículo se buscó comunicar una primera aproximación la evaluación en la enseñanza de la programación de cara a la elaboración de un trabajo de investigación a desarrollarse que, en primera instancia, busca caracterizar los criterios de evaluación en las materias de programación de las carreras del Módulo Informática de la FCEQyN de la UNAM.

Para esto, es necesario adentrarse en la revisión de la evolución de las herramientas de cómputo y los lenguajes de programación, problematizando qué características de este proceso han influido en las propuestas de enseñanza y los enfoques evaluativos adoptados. En este sentido, una recopilación inicial nos muestra que los enfoques como el análisis del portafolio, las entrevistas basadas en artefactos construidos, el diseño de escenarios y el aprendizaje basado en proyectos pretenden habilitar procesos de evaluación de la programación desde una mirada integral del aprendizaje de los estudiantes como sujeto activo. Estos métodos intentan no solo centrarse en el resultado final, sino también en el proceso de aprendizaje, fomentando así una comprensión más profunda y aplicada de los conceptos de programación.

En este sentido, esta revisión contribuye a pensar cómo las características específicas del campo configuran las racionalidades de evaluación. Habilitando el proceso un primer acercamiento al objeto de estudio y la definición de algunos interrogantes de investigación ¿Cómo el enseñante de programación de la FCEQyN construye las instancias de evaluación y cuáles son los criterios contemplados? ¿Qué bases y fundamentos son utilizados por los docentes para definir los criterios de evaluación en las materias de programación? ¿Cómo se planifica la operacionalización de los criterios de evaluación en las instancias de evaluación?

En síntesis, esta primera aproximación permite esbozar algunas de las cuestiones clave que orientarán el trabajo de investigación futura. La exploración de los enfoques evaluativos empleados en la enseñanza de la programación no solo aporta claridad sobre cómo se configuran los criterios de evaluación, sino que también invita a cuestionar cómo estos reflejan las racionalidades y valores propios del campo. Las preguntas que emergen sobre la construcción de las instancias evaluativas y la fundamentación de los criterios en el contexto de la FCEQyN de la UNAM plantean un terreno fértil para un análisis profundo y contextualizado. Entender cómo los docentes organizan sus prácticas evaluativas, y sobre qué bases teóricas y pragmáticas lo hacen, resulta fundamental para desarrollar una mirada crítica y propositiva que permita mejorar los procesos de enseñanza y evaluación en programación.

## Referencias

1. S. A. Giannattasio, E. M. Génova, y L. J. Génova, "Recorridos de los criterios de evaluación de los alumnos del Departamento de Ingeniería Agrícola y Forestal, Facultad de Ciencias Agrarias y Forestales, Universidad Nacional de La Plata," en IV Congreso Nacional y III Congreso Internacional de Enseñanza de las Ciencias Agropecuarias, 2012.
2. R. W. Hamming, "One man's view of computer science," Journal of the ACM, vol. 16, no. 1, pp. 3-12, 1969.
3. K. Brennan y M. Resnick, "New frameworks for studying and assessing the development of computational thinking," en Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, Canadá, 2012, pp. 1-25.
4. C. Cossio-Mercado and G. P. Fernández, "PRENDER: Una propuesta didáctico-pedagógica para la enseñanza de las Ciencias de la Computación," submitted to Jornadas Argentinas de Didáctica de las Ciencias de la Computación, 2023.
5. T. Dufva, Code Literacy: Understanding the Programmed World, Master's thesis, 2013.
6. A. Lovrenčić, M. Konecki, y T. Orehovački, "1957-2007: 50 years of higher order programming languages," Journal of Information and Organizational Sciences, vol. 33, no. 1, pp. 79-150, 2009.
7. I. Zakari, "History of computer and its generations," *ResearchGate*, 06 de mayo de 2024. [[https://www.researchgate.net/publication/336700280\\_History\\_of\\_computer\\_and\\_its\\_generations](https://www.researchgate.net/publication/336700280_History_of_computer_and_its_generations)].
8. M. Á. Santos Guerra, "Sentido y finalidad de la evaluación de la Universidad," 1999.
9. A. Camilloni, *Evaluar para conocer; examinar para excluir*, 2nd ed. Buenos Aires, Argentina: Paidós, 2003.
10. M. R. Dávila, "Comparativa de abordajes de cursos introductorios de programación," Revista Latinoamericana de Ingeniería de Software, vol. 4, no. 3, pp. 143-158, 2016.
11. W. H. Kilpatrick, "Project Method: The Use of the Purposeful Act in the Educative Process," Columbia University, 1918.
12. F. Díaz Barriga, E. Romero Martínez, y A. Heredia Sánchez, "Diseño tecnopedagógico de portafolios electrónicos de aprendizaje: una experiencia con estudiantes universitarios," Revista Electrónica de Investigación Educativa, vol. 14, no. 2, pp. 103-117, 2012.
13. J. M. Álvarez Méndez, *Evaluar para aprender: los buenos usos de la evaluación*, Universidad Complutense de Madrid, España, 2008.
14. J. Dewey, *Democracia y educación*, Ed. Losada, Buenos Aires, 1967.
15. J. Bruner, *La educación puerta de la cultura*, Ed. Visor, Madrid, 1997.

# Una aproximación a las actividades en la enseñanza superior de programación de videojuegos

Gaston Caminiti<sup>1</sup>

<sup>1</sup>Universidad Nacional de Misiones y Universidad Nacional del Litoral, Argentina  
[gaston.caminiti@fceqyn.unam.edu.ar](mailto:gaston.caminiti@fceqyn.unam.edu.ar)

## Resumen

Este artículo comprende una indagación conceptual perteneciente a una investigación cualitativa a desarrollarse en la Tecnicatura en Diseño y Programación de Videojuegos de la UNLVirtual, trabajo que se centrará en reconocer cómo los docentes de programación construyen las actividades de enseñanza.

El escrito explora el modelo de enseñanza en programación de videojuegos, la importancia de las actividades y sus características, así como interrogantes que orientan el estudio del quehacer docente situado.

**Palabras clave:** programación; videojuegos; construcción; actividades

## 1. Introducción

Programar videojuegos implica codificar reglas, Mecánicas que estructuran las acciones del jugador mientras juega, generando estas interacciones Dinámicas de juego, situaciones construidas intencionadamente para evocar emoción o Estética [1]. El desarrollo de videojuegos integra saberes multidisciplinares: Ludología, Física, Matemática, Psicología, Música, Literatura, Artes Visuales, Diseño Gráfico, Modelado 3D, Programación y otras áreas de las ciencias de la computación.

En los planes de estudio de carreras de videojuegos argentinas, la mayoría de las materias son de programación, informática y diseño, siendo una minoría las materias en ciencias sociales [2] [3]. La Tecnicatura en Diseño y Programación de Videojuegos, carrera vigente desde el 2009 en la Facultad de Ingeniería y Ciencias Hídricas perteneciente a la Universidad Nacional del Litoral, coincide con dicha tendencia: el currículum prioriza contenidos vinculados con las ciencias básicas y conocimientos avanzados de programación que se complementan con conocimientos básicos de diseño de videojuegos, dibujo y animación digital, con el propósito de formar técnicos que satisfagan requerimientos de diseño y programación de videojuegos [4].

La predominancia de los espacios de programación en la oferta académica motiva la indagación de las prácticas de enseñanza del saber computacional, explorando configuraciones propias, así como problematizando su inherente transversalidad y dialogando con posturas críticas hacia la técnica industrial moderna, como ser: significar la máquina como objeto funcional-utilitarista y serio, sin rasgos lúdicos e ilusionista [5], dimensiones que el videojuego tiene por esencia.

Vinculado con esta temática y contexto, el presente escrito resulta de las primeras indagaciones conceptuales en una investigación cualitativa a desarrollarse en la Tecnicatura en Diseño y Programación de Videojuegos de la UNLVirtual, la cual se focalizará en explorar cómo los docentes de programación construyen las actividades de enseñanza.

## 2. Computación y desarrollo de videojuegos

Para Rodrigo Mendizábal, la Máquina de Pensar de Ramón Llull (1275), artefacto de juego con meta pedagógica, es prototipo y ejemplo del videojuego moderno, por su cualidad para la generación discursiva y semiótica. El autor enuncia que los videojuegos muestran todas las facetas del uso social de las tecnologías en el sentido foucaultiano[6]:

- Producen el sentido de una segunda naturaleza mediante representaciones.
- Muestran la posibilidad del poder en ejercicio.
- Articulan el mundo de lo real alrededor de un mundo sígnico lo que hace posible que los individuos puedan modelar sus propias relaciones con lo exterior.
- Descorporeiza, exponiendo la contingencia de vivir en un entorno (aunque sea informático) en el que no existen límites, ubicuo y totalizante.

Así, la informática posibilita que los videojuegos actúen como juguetes racionalizados que habilitan la conceptualización primaria del mundo [6], manifestando el carácter estructurante de lo computacional en el desarrollo y distribución de videojuegos.

El vínculo computación y videojuego encuentra paralelismo en la relación de las tramoyas y el teatro del siglo de oro: En el teatro barroco, el mundo fantástico, las metamorfosis de la escena y las apariencias maravillosas son (en gran parte) efectos producidos por las tramoyas. Las tramoyas crean lo imaginario del barroco; teniendo el ingeniero ocasionalmente una función más importante que el director de teatro en el proceso de la realización de una obra teatral [5].

En el ciclo de desarrollo de videojuegos, el director de teatro es un/a diseñador/a de juegos que idea qué se juega y cómo. Su rol implica la creación de estructuras lúdicas, su labor, con origen en el Juego real de Ur y el Senet (2600 a.C) y afín a la tesis de Huizinga del juego como motor cultural, incursiona en el estudio multirreferencial para la generación de gameplay [7].

Aprender a desarrollar videojuegos requiere un espacio de apertura a nuevos modos de construcción de saberes y software desde una perspectiva interdisciplinaria, que atravesara la práctica profesional de los/as titulados/as en programación de videojuegos, pues estos profesionales deberán manipular variables complejas cuando deban producir los objetos que el mercado les demande [8].

Las ideas expuestas en esta sección motivan los siguientes interrogantes: ¿Cómo se enseña a los programadores de videojuegos? ¿Qué tareas desarrollan? ¿Cómo se preparan para el trabajo interdisciplinario con diseñadores, artistas, modeladores, animadores y otros perfiles?

## 3. La enseñanza en programación de videojuegos

La enseñanza en programación de videojuegos se asocia con el modelo construcciónista [10]. El construcciónismo es una teoría educativa propuesta por Seymour Papert de base constructivista piagetiana [11], que posiciona al enseñante como epistemólogo constructivista y reconoce a los educandos como constructores de sus propias estructuras intelectuales [12].

La idea de aprendizaje en el construcciónismo es genética y, en consonancia con Piaget, el proceso de aprendizaje es inseparable de lo que se aprende. Es decir, lo que puede aprender un individuo y cómo lo aprende depende de los modelos con los que cuenta, siendo los objetos con los cuales pensar (computadoras entre ellos) dispositivos para la construcción de conocimiento.

Estos *objetos para pensar* son entidades definidas por Papert, recuperando a Ostwald, como artefactos cognitivos que proporcionan conexiones entre el conocimiento sensorial y el conocimiento abstracto, y entre el mundo individual y el mundo social [11].

Si bien la noción de artefacto con el cual pensar es anterior y puede remitirse a ideas desarrolladas por Jean-Jacques Rousseau en Emilio ¿Acaso no habría modo de aproximar todas las lecciones desparramadas en tantos libros, de reunirlas en un objeto común, que pudiera ser fácil ver, interesante seguir y servir de estimulante aun en esta edad? Si es posible inventar una situación en que de un modo sensible se manifiesten al espíritu de un niño las necesidades naturales del hombre, y con la misma facilidad se desarrolle sucesivamente los medios de remediar estas mismas necesidades, el primer ejercicio que se debe dar a su imaginación es la pintura viva y natural de este estado [13]

Los modos de aprendizaje que favorecen los *objetos para pensar* son, por un lado, el modo de experiencia, en el que la información es percibida y manipulada dándose por sentados tanto los objetos como el conocimiento, y, por otro lado, el modo reflexivo que impone un esfuerzo mental mayor para evaluar las distintas posibilidades de acción en el mundo [12].

Por lo tanto, el constructivismo alude una experiencia mediada por objetos, que implica cuestionar su significado y conformación. Reconociendo que la experiencia intermental da forma al desarrollo *intramental*, proceso mediado en el que artefactos producidos culturalmente (formas de habla, representaciones como ideas, creencias, signos y símbolos) dan forma y son conformados por el compromiso humano con el mundo [14].

Adicionalmente, el constructivismo enfatiza en la práctica pública, enunciando como *Entidades Públicas* a los trabajos estudiantiles mostrados, examinados, probados y admirados. Donde las representaciones visuales o auditivas de ideas y conceptos son producciones a ser experimentadas por otros y destinadas a ser compartidas, convirtiéndose en una organización pública a través de la cual el aprendizaje es poderosamente reforzado [11].

Así el script, programa, software o aplicación creada por el estudiante, en una experiencia de construcción mediada por *Objetos para pensar*, se transforman, por su atributo público, en una entidad para aprender en comunidad. Esta característica se manifiesta en desarrollo de videojuegos en la relevancia de prácticas como el playtest y las game jam, situaciones dispuestas para indagar si el videojuego desarrollado divierte a otros o cómo podría proveer más diversión a determinado público objetivo.

Otro concepto relevante en el modelo de Papert, desarrollado en forma conjunta con Marvin Minsky, son los *Micromundos*. Estos espacios, construidos con *Objetos para pensar* y considerados una *Entidad Pública*, comprenden modelos de representaciones de una realidad inmediata sobre un tema, que habilitan a los estudiantes a explorar alternativas, probar hipótesis y descubrir hechos que son verdad en relación con ese mundo. Siendo de interés distinguir que un micromundo y una simulación no son expresiones equivalentes [11].

Es decir, recuperando a Solórzano, construir un micromundo implica crear un microcosmos o lugar donde el estudiante vivencia experiencias directas y físicas, así como el sitio donde puede obtener los medios para conceptualizar y significar el mundo de determinado conocimiento. Estos ambientes se orientan por tres principios [12]:

- *Principio de poder o dominio*: Impulsar al agente en la resolución autónoma de conflictos, buscando el dominio de la situación desde su propia visión.
- *Principio de resonancia cultural*: El micromundo responde al modelo cultural propio.
- *Principio de continuidad cognoscitiva*: Respeto a los tiempos evolutivos individuales y la integración con saberes previos.

Adicionalmente, se identifica como *micromundos computacionales* a los ambientes que pueden ser definidos como conjuntos de herramientas computacionales abiertas, destinadas a que el estudiante explore, construya ideas y conceptos a través de actividades de programación [11].

## 4. Las actividades en programación de videojuegos

La centralidad en la acción del estudiante está en sintonía con el debate didáctico contemporáneo, que enfatiza la responsabilidad docente para proponer actividades secuenciadas a los estudiantes, buscando establecer climas de aprendizaje que, evocando la teoría de situaciones didácticas de Rousseau, reconocen que el sujeto aprende por lo que realiza, por la importancia de la actividad llevada a cabo, por la posibilidad de integrar nueva información en concepciones previas que posee, por la capacidad que logra al verbalizar ante otros la reconstrucción de información [9].

Los análisis de Camilloni [14] sobre teorías generales de la experiencia y de la acción, nos permiten poner en juego dos interrogantes para investigar el diseño de actividades en la enseñanza superior de programación de videojuegos:

### ¿Cómo se significan las actividades?

Cliff Bleszinski, diseñador de videojuegos mencionó que no hay nadie en esta industria que sepa lo que está haciendo, solo tenemos suposiciones instintivas [15]. Tanto el videojuego, software creado con y para otros, y el ideario construcciónista de aprender haciendo en comunidad, demandan esquemas de enseñanza y aprendizaje que habiliten la exploración y acción colectiva.

A partir de las nociones aristotélicas de kinesis y energeia, Hannah Arendt (1958) concede a la energeia la “*carenza de límites de la acción*”, porque la obra no queda encerrada entre un conjunto de individuos identificables, se es agente, pero no autor indiscutible de la acción. Esto admite, recuperando a Dewey, que la acción tiene resultados inciertos, porque es nueva, aunque se proponga servir de enlace con la tradición. Y que desarrollar en los alumnos la capacidad para la acción es responsabilidad de los profesores, mediante la creación de las condiciones en las cuales los estudiantes deban enfrentar riesgos y asumir responsabilidades [14].

### ¿Cómo se comunican las actividades?

Bernstein (2000) advierte que los modos de comunicación dominantes, regulados por el discurso pedagógico, determinan el valor formativo de las actividades. Voz (posibilidad) y mensaje (posicionamiento) estructuran la agencia, poniéndose en juego las relaciones de poder y control [14].

Si bien la programación de computadoras encuentra sus orígenes en el sistema binario de Leibniz de 1679, pudiendo simplificar la naturaleza de todo sistema digital en verdadero o falso [16], en cuanto a la formación de programadores/as la racionalidad se perfila como ingenieril [17].

Los programas no son escalables o confiables por construirse mediante algoritmos y modelos predictivos. Mientras que crear software puede ser visto como un proceso divertido y creativo, el mantenimiento es desafiante. Los programas nunca están completos, casi siempre requieren actualizaciones, correcciones de errores o agregación de nuevas características, así como soporte a nuevas tecnologías y usuarios [16]. Desvelar aquello a programar o depurar implica la interpretación de retroalimentación o “feedback” constante de los usuarios, o en este caso jugadores.

Similar al Theatrum Machinarum del Siglo de Oro, donde comunicar lo técnico adopta tintes ambivalentes entre lo práctico-realizable e inconcebible-fantástico, presentada el funcionamiento de las máquinas como forma alternativa de conocimiento y modelo conceptual interdisciplinario [5], hablar sobre programar videojuegos, construcción del emergente Décimo Arte [6], entreteje discursos técnicos, ilusorios y artificiales.

## 5. Rasgos de las actividades en programación de videojuegos

Reconocidas las actividades como centrales en la enseñanza superior de programación de videojuegos se definen tres dimensiones que las configuran:

- *Diseño:*

En las ciencias de computación, los conocimientos de la ingeniería de software tienen por meta la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento del software [18]. Pero, al examinar obras referentes en creación de videojuegos, Costikyan [19], Hunicke [1], Koster [20] y Nallar [7], se observa que los sentidos de diseño se construyen diferente, encontrando semejanzas en lo metodológico, pero diferencias en las representaciones abstractas, por ejemplo: En la programación de software de propósito general se parte del *requerimiento*, en la programación de videojuegos de la *mecánica*.

- *Motores de Videojuegos:*

El término "motor del videojuego" refiere a un conjunto de módulos, código fuente que no es el comportamiento del videojuego (mecánicas) ni el entorno juego (niveles), que administran las entradas y las salidas de usuario, el cálculo físico, el renderizado 3D, el dibujo 2D, los sonidos y recursos del proyecto [21]. Es decir, la programación de un videojuego se realiza a partir de un software ensamblado para crear videojuegos, el cual estructura la producción.

La publicación de 5 años del Observatorio de la industria argentina de Videojuegos de la Universidad Nacional de Rafaela (UNRaf), en alianza estratégica con la Asociación de Desarrolladores de Videojuegos Argentina (ADVA), señala que los motores de videojuegos más empleados son Unity y Unreal [22]. Tecnologías propietarias de acceso gratuito que no son de uso exclusivo a la creación de videojuego, empleándose en campos más amplios como la arquitectura, la inteligencia artificial, la fabricación, la planificación pública, la cinematografía y se presentan como plataformas impulsoras del desarrollo de realidad virtual (VR), realidad aumentada (AR) y el metaverso, emergiendo los motores de videojuegos como la base oculta de la digitalización cada vez mayor de la vida social, económica y política [23].

Adicionalmente, se evidencia la adopción de motores de videojuegos libres, particularmente en la enseñanza. Puede ser interesante problematizar estos casos bajo las miradas de Lins Ribeiro [24] sobre la economía de carnada, o Isoglio [25] sobre la mercantilización del software libre, o el sesgo de escalabilidad presentado por Dufva [16], ideario que favorece a quienes pueden costear dicha escalabilidad, dependiendo de sistemas unificados y estandarizados, liberados y obligados al mismo tiempo.

- *Trabajo en equipo:* En desarrollo de videojuegos se ven comprometidos múltiples roles. La actividad principal de quien programa es codificar los conceptos desarrollados por las personas diseñadoras, y existen roles como Artista Técnico, que poseen conocimientos artísticos y de programación para articular tareas. Pero si bien, programar videojuegos comprende implementar el trabajo de diseñadores y artistas, codificando lógicas e interacciones que el juego requiera [22], la responsabilidad de desarrollo es interdisciplinaria, exponiendo la interacción con lo programado a revisiones y retroalimentaciones del colectivo.

## 6. Conclusiones

Si bien la predominancia de espacios de programación en ofertas académicas de videojuegos argentinas y la importancia de la computación en el desarrollo de videojuegos es móvil suficiente para indagar las prácticas de enseñanza del área, el artículo concluye con que es la tarea docente de creación de actividades un factor determinante en la formación en programación, aprendizaje caracterizado por el aprender haciendo de significación construcción.

Estudiar prácticas de diseño de actividades permitirá conocer cómo los docentes piensan la enseñanza de la programación de videojuegos, cuya filiación, si bien es computacional, pone en juego las críticas a la técnica industrial moderna, ya que esta máquina/artefacto que codifica el estudiante no es un objeto funcional-utilitarista y serio; al contrario, está dotado de dimensión lúdica, ilusionista, artística y narrativa, porque es un videojuego, un software creado con el propósito de divertir.

Adicionalmente, la carrera en la que se desarrollará esta investigación pertenece al Sistema Institucional de Educación a Distancia UNLVirtual y se debe remarcar que las actividades mediadas por entornos virtuales presentan singularidades tecno-pedagógicas que debes ser estudiadas, por lo que el trabajo comprenderá una oportunidad inédita para indagar prácticas actuales de enseñanza de la programación en modalidad virtual.

En síntesis, en base a los conceptos expuestos en este escrito se desarrolla la pregunta inicial de investigación *¿Cómo los docentes de programación construyen las actividades de enseñanza virtual en la Tecnicatura en Diseño y Programación de Videojuegos de la UNLVirtual?* cuestionamiento ideado en sintonía con la perspectiva de reivindicación del oficio artesano de la enseñanza de Alliaud [27], valorando el estudio del quehacer docente situado.

## Bibliografía

- [1] R. Hunicke, M. LeBlanc, and R. Zubek, "MDA: A formal approach to game design and game research," in Proc. AAAI Workshop Challenges in Game AI, 2004, vol. 4, no. 1, p. 1722.
- [2] G. Esnaola Horacek, M. G. Galli, and M. D. L. P. Colla, "La formación superior en desarrollo de los videojuegos en la educación argentina," EduQ, 2017.
- [3] M. B. De Ansó, E. L. Artola, and M. E. Conde, "Experiencias pedagógicas en la Formación Superior de los Técnicos en Producción y Diseño de Videojuegos en UNPAZ," Rev. Estud. Exp. Educ., vol. 18, no. 38, pp. 195-209, 2019.
- [4] C. Giorgetti, H. Loyarte, G. Martín, M. Paredes, V. Pedrón, and S. Steitelman, "Tecnicatura en Diseño y Programación de Videojuegos implementación de una propuesta académica innovadora en la región bajo la modalidad a distancia," in Proc. Segundo Congr. Virtual Iberoam. Calidad Educ. Distancia, 2009.
- [5] S. Friedrich, "La performance teatral de las máquinas maravillosas: Configuraciones ambivalentes de la técnica y el teatro en los siglos XVI y XVII," Olivar, vol. 16, no. 23, pp. 00-00, 2015.
- [6] I. F. Rodrigo Mendizábal, Máquinas de pensar: videojuegos, representaciones y simulaciones de poder, Ecuador: Universidad Andina Simón Bolívar, Corporación Editora Nacional, Ediciones Abya Yala, 2004.
- [7] D. A. Nallar, Diseño de juegos en América Latina: teoría y práctica, Argentina: Durgan A. Nallar, 2015.
- [8] D. P. Corsi, M. G. Galli, F. I. R. Domínguez, and E. G. Torchia, "Aprender programación desarrollando videojuegos: una experiencia con Java y LibGDX," in Proc. XXXII Jornadas Argentinas de Informática, 2017, pp. 71-78.
- [9] Á. Díaz-Barriga, "Guía para la elaboración de una secuencia didáctica," UNAM, México, 2013, pp. 1-15.
- [10] M. E. Conde and E. García, "Hacé tu videojuego. Aprender a programar creando. La Construcción narrativa de un Serious Game," Construcción narrativa de un Serious Game, vol. 9, no. 44, 2016.
- [11] E. B. Saxe and A. C. Murillo, "Construcciónismo: Objetos para pensar, entidades públicas y micromundos," Actual. Investig. Educ., vol. 4, no. 1, pp. 0-0, 2004.
- [12] C. M. V. Solórzano, "Construcciónismo. Referente sociotecnopedagógico para la era digital," Innovación Educativa, vol. 9, no. 47, pp. 45-50, 2009.
- [13] J. Rousseau, Emilio o De la educación, 1762. [Online]. Available: <https://www.educ.ar/recursos/70109/emilio-o-de-la-educacion-de-jean-jacques-rousseau>. [Accessed: Day-Month-Year].
- [14] A. R. W. de Camilloni, "Situaciones, tareas y experiencias," Actual. Pedag., vol. 1, no. 59, pp. 15-32, 2012.

- [15] M. Zahartovskyi, "Optimization of the product concept for the videogame market by applying econometric modelling," M.S. thesis, [Institution Name], 2022.
- [16] T. Dufva, "Code Literacy. Understanding the programmed world," M.S. thesis, [Institution Name], 2013.
- [17] R. W. Hamming, "One man's view of computer science," J. ACM, vol. 16, no. 1, pp. 3-12, 1969.
- [18] G. Pantaleo and L. Rinaudo, Ingeniería de software, Alpha Editorial, 2015.
- [19] G. Costikyan, "I have no words & I must design: toward a critical vocabulary for games," 2002. [Online]. Available: [URL]. [Accessed: Day-Month-Year].
- [20] R. Koster, Theory of fun for game design, O'Reilly Media, 2013.
- [21] M. Lewis and J. Jacobson, "Game engines," Commun. ACM, vol. 45, no. 1, pp. 27-31, 2002.
- [22] F. Curbelo, B. Ferrero, and H. Revale, "Cinco años del Observatorio de la Industria Argentina de Videojuegos: Caracterización de la industria hacia una proyección estratégica," Ediciones UNRaf, 2022. [Online]. Available: [https://www.unraf.edu.ar/images/INVESTIGACION/OBSERVATORIO\\_VideoJuegos/PDFinformes/informe\\_observatorio\\_5aos.pdf](https://www.unraf.edu.ar/images/INVESTIGACION/OBSERVATORIO_VideoJuegos/PDFinformes/informe_observatorio_5aos.pdf). [Accessed: Day-Month-Year].
- [23] A. Jungherr and D. B. Schlarb, "The extended reach of game engine companies: How companies like epic games and Unity technologies provide platforms for extended reality applications and the metaverse," Soc. Media Soc., vol. 8, no. 2, 2022.
- [24] G. Lins Ribeiro, "El precio de la palabra: la hegemonía del capitalismo electrónico-informático y el googleísmo," Desacatos, no. 56, pp. 16-33, 2018.
- [25] A. Isoglio, "El devenir abierto del software libre: o acerca de cómo fue concertado el avance del proceso de mercantilización," in Organizaciones y movimientos periféricos nas redes digitais ibero-americanas, 2022, pp. 301-322.
- [26] J. Silva, "Un modelo pedagógico virtual centrado en las E-actividades," Rev. Educ. Distancia, no. 53, 2017.
- [27] A. Alliaud, Los artesanos de la enseñanza: acerca de la formación de maestros con oficio, 2017.

# Uso de IA en programación inicial desde la perspectiva de los estudiantes

Gladys Dapozzo, Cristina Greiner, Raquel Petris, Ana María Company, María Cecilia Espíndola

Facultad de Ciencias Exactas y Naturales y Agrimensura (UNNE), Corrientes, Argentina

[gndapozzo@exa.unne.edu.ar](mailto:gndapozzo@exa.unne.edu.ar), [cgreiner@exa.unne.edu.ar](mailto:cgreiner@exa.unne.edu.ar), [raquel.petris@comunidad.unne.edu.ar](mailto:raquel.petris@comunidad.unne.edu.ar), [anamacom@gmail.com](mailto:anamacom@gmail.com),  
[mcespindola@exa.unne.edu.ar](mailto:mcespindola@exa.unne.edu.ar)

## Resumen

La enseñanza de la programación en carreras de Informática enfrenta nuevos desafíos. Por un lado, el aumento de la matrícula, impulsado por la virtualización post-pandemia, y la creciente expectativa de los jóvenes por esta formación. Por otro lado, la aparición de la inteligencia artificial (IA) generativa, que ha impactado rápidamente en diversos campos, incluida la educación en Ciencias de la Computación. Este trabajo presenta los resultados de una encuesta a estudiantes de un curso de programación de primer año en una carrera de Informática, cuyo objetivo fue explorar cómo utilizan las herramientas de IA en su aprendizaje y su opinión sobre su eficacia. Los resultados indican que la mayoría ha utilizado herramientas de IA generativa para apoyar su aprendizaje, aunque han encontrado dificultades para obtener soluciones alineadas con las pautas de la asignatura. Esto subraya la necesidad de una instrucción formal que enseñe a formular preguntas adecuadas para alcanzar las soluciones esperadas. Como conclusión, se propone elaborar a futuro una metodología que integre las herramientas de IA en la enseñanza, buscando maximizar su uso sin comprometer el desarrollo del pensamiento computacional y la abstracción necesarios para una sólida formación en programación.

**Palabras clave:** Enseñanza de la programación, Enseñanza universitaria, IA generativa.

## 1. Introducción

### 1.1. Acerca de la Inteligencia Artificial en la educación

Pareciera que la Inteligencia Artificial (IA) irrumpió abruptamente, en estos últimos tiempos, en la vida cotidiana del hombre, pero en honor a la verdad esta es una disciplina que se está gestando desde la primera mitad del siglo pasado. Según [1] en 1943 Warren McCulloch y Walter Pitts presentaron su modelo de neuronas artificiales, considerada en la actualidad, la primera inteligencia artificial, aun cuando todavía no existía el término. Tiempo después en 1950, Alan Turing publica el conocido artículo “Computing machinery and intelligence” (“Maquinaria e inteligencia informática”) en la revista Mind, dónde plantea el enigma de ¿Pueden pensar las máquinas?, pregunta retórica con la que algunos catedráticos de programación iniciaban sus clases ante sus novedosos estudiantes. Por otra parte, basándose en el Test de Turing, científicos de la época sustentaban sus investigaciones para determinar si el comportamiento inteligente de estas máquinas era similar al de un ser humano o indistinguible de este.

El término “inteligencia artificial” [2] es acuñado por John McCarthy en 1958, se impulsó en la década de 1960 con el desarrollo del primer lenguaje de programación de IA, LISP. Los primeros sistemas de inteligencia artificial se centraron en reglas, lo que condujo al desarrollo de sistemas más complejos en las décadas de 70 y 80. En esos momentos, la IA experimentó un renacimiento gracias a avances en algoritmos, hardware y técnicas de redes permitiendo el entrenamiento-aprendizaje profundo que optimizan las redes neuronales artificiales con múltiples capas para procesar e interpretar estructuras de datos de un nivel más complejo [3].

Se podría decir que la IA se convirtió en el eje central de la innovación en la computación moderna, tanto para las personas como para las empresas. Por ejemplo, el reconocimiento óptico de caracteres (OCR), que ya tiene varias décadas en las organizaciones, combinado con las bondades de la IA para generar y extraer texto, datos de imágenes y documentos de cientos de miles de bases de datos, reconvierte el contenido no estructurado en datos estructurados útiles para las empresas, sumado a la generación de estadísticas valiosas, proyecciones de negocios, proyectos institucionales, entre

muchas opciones más. Pero esto tiene sus riesgos, las empresas deben ceder a los algoritmos de IA sus datos para entrenar al sistema, los cuales son tomados para generar el proceso y obtener los resultados, pero esos datos no serán devueltos.

Tal como se dijo anteriormente la IA irrumpió en todos los ámbitos en donde se desarrolla el hombre, por lo que también llegó a las universidades y aquí se presenta un gran dilema en los estudiantes: para qué aprender, memorizar o razonar si esto de algún modo se puede obtener desde un dispositivo móvil con algoritmos de IA. Por lo que se convierte en una preocupación específica para muchos educadores el hecho de que estas herramientas pueden resolver eficazmente tareas y problemas de exámenes en una amplia variedad de espacios curriculares [4], pero no producen conocimiento en los estudiantes.

Gracias a la IA generativa, el ámbito de la educación se encuentra en la cúspide de una revolución. Entre las ventajas de esta tecnología se puede mencionar: el aprendizaje personalizado, que permite adaptar los itinerarios educativos para que se ajusten a las necesidades individuales de los estudiantes; los sistemas de tutoría virtual pueden contribuir en áreas con escasez de docentes proporcionando retroalimentación instantánea, asegurando el progreso continuo de los estudiantes; la creación de contenidos fomenta la creatividad entre los estudiantes, pueden crear arte, componer música o incluso escribir ensayos, adaptado a su estilo y preferencias particulares; la traducción en tiempo real ha experimentado notables avances en precisión; el procesamiento del lenguaje natural ha aumentado las interacciones entre humanos y computadoras, haciéndolas más intuitivas [5].

Sin embargo, este inmenso poder trae aparejada una gran responsabilidad. Una mayor dependencia de la IA podría llevar a un estancamiento cognitivo en los estudiantes, frustrando el propósito mismo de la educación. Desde el punto de vista ético, los modelos de IA, que reflejan los datos con los que se entrena, pueden perpetuar inadvertidamente los sesgos sociales. Es crucial garantizar que estos modelos sean equitativos y no profundicen aún más las divisiones sociales. Por último, la cuestión de la autenticidad sigue siendo importante. En un mundo donde distinguir entre contenido generado por humanos y por IA se vuelve cada vez más difícil, garantizar la confianza y la transparencia es primordial. En conclusión, el futuro iluminado por la Inteligencia Artificial Generativa es a la vez prometedor y desconcertante. A medida que la IA generativa continúa reformulando los paradigmas de enseñanza, aprendizaje y comunicación, nuestra responsabilidad colectiva es garantizar que su recorrido esté anclado en la ética, la equidad y la excelencia [5].

## 1.2. Herramientas de IA en programación

Puntualmente en las aulas de las universidades donde se imparte educación informática, los investigadores han descubierto que las herramientas de IA pueden ser especialmente efectivas para la programación debido a que están entrenadas en miles de millones de líneas de código fuente abierto y, debido a que el código tiene una estructura lógica más restringida que el lenguaje natural, este proceso es más sencillo. Estas herramientas pueden generar soluciones a tareas de programación y preguntas de exámenes y explicar el contenido del código. Frente a esta realidad y avisorando que la IA llegó para quedarse y que, a medida que más se entrena, probablemente mejoren las capacidades de la IA, surgen algunos planteamientos sobre qué hacer en las aulas o cómo adaptar los cursos o las clases a esta herramienta tan asequible y eficiente [4].

ChatGPT es un chatbot basado en IA con habilidades de generación automática de código. La literatura reciente menciona su potencial para mejorar la calidad de la educación en programación al ofrecer a los estudiantes oportunidades para comprender mejor los principios de la programación. Sin embargo, existen pocos estudios empíricos que hayan explorado el impacto de ChatGPT en los procesos de programación de los estudiantes. En [6] realizaron un estudio dividendo aleatoriamente a los estudiantes en dos clases. Una clase empleó la práctica de programación facilitada por ChatGPT (CFP) y la otra clase utilizó el modo de programación autodirigida (SDP). Determinaron que los estudiantes en el modo CFP tuvieron comportamientos más frecuentes de copiar y pegar códigos de ChatGPT y que la práctica CFP aparentemente mejora el rendimiento en programación de los estudiantes universitarios, aunque los resultados indicaron que no hubo una diferencia estadísticamente significativa entre los estudiantes en el modo CFP y el modo SDP.

En otro estudio [7] se evaluó el pensamiento computacional, la autoeficacia en programación y la motivación para el aprendizaje de la programación. Los estudiantes de un grupo experimental usaron ChatGPT durante las prácticas de programación semanales, mientras los estudiantes del grupo de control no utilizaron esta herramienta. Los resultados de la investigación revelaron que las habilidades de pensamiento computacional, la autoeficacia en la programación y la motivación para el aprendizaje de los estudiantes del grupo experimental eran significativamente mayores que las de los estudiantes del grupo de control.

## 1.3. La IA en la educación en Ciencias de la Computación

En diciembre de 2023, Yasmin B. Arafat, desarrolladora de la herramienta Scratch, dijo en su discurso de apertura en ACM CompEd que nunca había habido una conversación que dominara de manera tan ubicua a la comunidad educativa como la conversación sobre la IA Generativa. Esta rapidez sorprende porque las tecnologías anteriores, como la PC e Internet, tardaron muchos años en impactar en toda la sociedad. En mucho menos tiempo la IA Generativa ha "devorado" internet y ha logrado algo que la PC e internet no pudieron realmente hacer: Generar cosas. La IA Generativa puede generar contenido con relativamente poca intervención humana, generalmente en forma de lenguaje natural.

Respecto de la enseñanza de la programación, en general, un enfoque centrado en la IA funciona bien y muestra potencial para superar algunas barreras bien conocidas y de larga data en la educación en programación, como, por ejemplo, los errores de sintaxis, en los que la IA Generativa ha demostrado superar en gran medida [8].

Entonces, ¿hacia dónde vamos? El "apocalipsis de la evaluación", en el cual docentes prohibieron el uso de herramientas de IA, está disminuyendo. "Después de todo, hacer trampa no es un fenómeno nuevo y la IA Generativa es solo otro medio para el mismo fin" [8]. Sin embargo, está claro que la política universitaria debe abordar directamente los impactos de la IA Generativa en la integridad académica, y que esto debe ser claramente reglamentado y conocido por los estudiantes.

Superado el pánico inicial, surgen oportunidades donde la IA Generativa puede impactar positivamente en la enseñanza y el aprendizaje, en particular, en la programación. Para empezar, es necesario entender más sobre cómo los estudiantes interactúan con la IA Generativa y proporcionar herramientas y estrategias para lograr un uso efectivo. Respecto de las preocupaciones, esta abarca sesgos, ética y equidad. En educación, la IA Generativa podría afectar los procesos y estructuras educativas existentes. La IA Generativa podría habilitar el surgimiento de nuevas brechas y disparidades en términos de equidad y participación, como el desarrollo de una brecha digital en IA.

Pero, por otra parte, la IA puede bajar la barrera para entender cómo programar y, a su vez, atraer a un grupo más amplio y diverso de estudiantes y profesionales al campo. Sin embargo, los efectos a largo plazo se verán en algún tiempo. ¿Qué hacer si los estudiantes asistidos por IA superan rápidamente los contenidos? ¿Evaluar con más dureza? ¿Cubrir más contenido en el mismo tiempo? ¿Repensar completamente cómo funciona la evaluación en la educación superior? Estas cuestiones tendrían efectos drásticos en el plan de estudios, así como en la creación de profundas cuestiones profesionales, económicas, sociales, éticas y otras.

En este sentido, puede verse a la IA Generativa como un catalizador para acelerar cambios en los sistemas educativos. Los que se desempeñan como docentes en el sistema educativo saben que la educación es un sistema frágil que es fácil de alterar y notoriamente lento para cambiar. Una prueba bastante reciente de esta afirmación es el impacto de la pandemia por Covid19. La emergencia produjo un aterrizaje brusco en la virtualidad, sin embargo, muchas de las prácticas de la virtualidad no se mantuvieron pasada la emergencia.

Según ChatGPT, el potencial de la IA generativa en la educación es muy amplio y podría transformar significativamente cómo se enseña y se aprende. A continuación, se trascrcribe su respuesta:

*Aquí te presento algunos de los aspectos clave:*

1. *Aprendizaje Personalizado: La IA generativa puede adaptar el contenido educativo a las necesidades individuales de cada estudiante. Puede considerar factores como el nivel de conocimiento, el ritmo de aprendizaje y los intereses personales, ofreciendo materiales y actividades que se ajusten a cada estudiante de manera única.*
2. *Creación de Contenido: La IA generativa puede crear materiales educativos, como ejercicios, cuestionarios, presentaciones y hasta explicaciones detalladas sobre temas específicos, de manera automática. Esto permite a los educadores ahorrar tiempo y concentrarse en la enseñanza y el apoyo directo a los estudiantes.*
3. *Acceso a Información y Recursos: GenAI puede proporcionar acceso a vastas cantidades de información, generando resúmenes, análisis o materiales educativos basados en las últimas investigaciones. Esto es especialmente valioso en áreas con recursos limitados, donde los estudiantes podrían no tener acceso a una educación de calidad.*
4. *Interacción y Aprendizaje Inmersivo: Las herramientas impulsadas por IA pueden crear entornos de aprendizaje interactivos e inmersivos, como simulaciones o experiencias de realidad virtual, que permiten a los estudiantes explorar conceptos complejos de manera práctica y visual.*
5. *Evaluación y Retroalimentación Automatizada: GenAI puede evaluar el desempeño de los estudiantes de manera rápida y precisa, proporcionando retroalimentación inmediata y personalizada. Esto ayuda a los estudiantes a identificar sus errores y mejorar en tiempo real.*
6. *Apoyo a Docentes: La IA generativa puede asistir a los docentes en tareas administrativas, como la creación de planes de estudio o la corrección de exámenes, liberando tiempo para que se enfoquen más en la interacción humana y en el apoyo emocional y social de los estudiantes.*
7. *Fomento del Aprendizaje Autónomo: GenAI puede facilitar el aprendizaje autodirigido, proporcionando a los estudiantes recursos adaptados a sus necesidades y permitiéndoles aprender a su propio ritmo, fuera del aula tradicional.*
8. *Educación Inclusiva: La IA generativa puede ayudar a personalizar el aprendizaje para estudiantes con necesidades especiales, creando materiales adaptados a diferentes discapacidades y estilos de aprendizaje.*

A pesar de todas estas ventajas señaladas por ChatGPT, también es crucial considerar los desafíos asociados con la implementación de la IA en la educación, como garantizar la privacidad de los datos, evitar sesgos en los algoritmos, y mantener la interacción humana en el proceso educativo. Si se integra de manera responsable, la IA generativa tiene el potencial de hacer que la educación sea más accesible, personalizada y efectiva para todos.

## 1.4. Escenario de la experiencia

La asignatura Algoritmos y Estructuras de Datos I de la carrera Licenciatura en Sistemas de Información de la Universidad Nacional del Nordeste tiene como propósito introducir a los estudiantes en la programación. Al ser una materia del primer año y primer cuatrimestre, presenta históricamente un alto nivel de complejidad debido a las características de los ingresantes, quienes, además de incorporar los temas disciplinares específicos, deben adaptarse a la vida universitaria. Desde el año 2023, esta cátedra se ha convertido en una asignatura masiva con más de 1.000 estudiantes inscriptos.

Debido a la masividad, la asignatura ha implementado algunas innovaciones pedagógicas como la metodología de programación por pares (*pair programming*), laboratorios móviles y el uso de rúbricas en Moodle, con el fin de lograr los objetivos de formación previstos. Estas estrategias resultaron satisfactorias tanto para los docentes como para los estudiantes. Sin embargo, la aparición de la IA generativa complicó este panorama.

Conscientes de la necesidad de abordar la cuestión del uso de herramientas de IA en el aprendizaje de la programación, al final del curso 2024 se realizó una encuesta a los estudiantes para recabar su opinión acerca del uso de dichas herramientas en el proceso de aprendizaje de los conceptos y técnicas de la asignatura, con el objetivo de, en base a este conocimiento, formular una metodología de enseñanza que incorpore herramientas de IA generativa como recurso para apoyar el proceso de enseñanza aprendizaje de la programación introductoria en la universidad.

## 2. Metodología

Después de la pandemia, como ocurrió en la mayoría de las carreras de Informática del país, la matrícula se incrementó notablemente llegando a 1.113 inscriptos en el ciclo lectivo 2023 y 1.083 alumnos en el ciclo lectivo 2024.

Ante la masividad la asignatura implementó un conjunto de nuevas estrategias, tales como, la programación por pares, el uso de laboratorios móviles y una nueva modalidad de evaluación, y continuó utilizando el desarrollo de actividades lúdicas al inicio del dictado, con el propósito de introducir los conceptos básicos de programación. Sumado a esto, la implementación de un método de resolución de problemas, mantenido durante todo el dictado. Los resultados de la aplicación de estas estrategias se encuentran detallados en [9] y [10].

Además, desde lo institucional se habilita la posibilidad de dictar clases virtuales (síncronas o asíncronas) hasta el 30% de la carga horaria de la asignatura. En las clases prácticas, el trabajo por pares permite la auto-contención y sostenimiento de los alumnos en el sistema universitario. El uso de dispositivos móviles en las aulas permite intensificar la práctica. En la evaluación, se fortalece e intensifica la aplicación de rúbricas implementadas en la plataforma UNNE Virtual que utiliza Moodle como soporte tecnológico, permitiendo la devolución de correcciones en forma casi personalizada. La suma de estas medidas ha permitido mitigar, en la medida de lo posible, el impacto de la masividad.

Teniendo en cuenta este contexto, se describe a continuación la organización de las actividades de la asignatura y las condiciones para regularizar o promocionar.

Se dictan clases teóricas y prácticas con una carga horaria semanal de 8 h, que comprenden:

- 2 hs de clases teóricas en modalidad virtual sincrónica a través de la plataforma Zoom en 2 horarios diferentes, para grupos de hasta 500 alumnos (capacidad máxima de la plataforma).
- 4 hs de clases prácticas presenciales, en 2 clases de 2 horas cada una. Cabe señalar que la disponibilidad del espacio físico obligó a agrupar a los alumnos en 8 comisiones de prácticos, atendiendo cada una de ellas a más de 100 alumnos.
- 1 hora de clase práctica en la modalidad virtual sincrónica, con el objetivo de poner en común las soluciones de los ejercicios que los docentes indicaban expresamente que debían subir a una tarea habilitada en el aula virtual.
- 1 hora de actividad asíncrona, destinada a la resolución del ejercicio encomendado por el docente de cada comisión.

Dada la imposibilidad de una atención más personalizada, esta organización tiene el propósito de obtener evidencia del aprendizaje de los estudiantes.

La evaluación consiste en 2 exámenes parciales prácticos y 1 examen teórico, cuya aprobación permite al alumno regularizar o promocionar, según las siguientes condiciones:

Para regularizar la materia:

- 75% de asistencia a las clases teórico-prácticas.
- Aprobación del 100% de los trabajos prácticos encomendados.
- Aprobación de 2 (dos) exámenes parciales prácticos con nota mayor o igual a 6.

Para aprobar la materia sin examen final (promoción):

Los alumnos que cumplen las condiciones para regularizar, cuyo promedio de notas de los parciales prácticos aprobados es mayor o igual a 7, pueden rendir un tercer parcial de conceptos teóricos.

- Los alumnos que aprueben el tercer parcial con nota mayor o igual a siete (7), aprueban la materia sin examen final.
- La nota final será el promedio de las notas de los tres parciales aprobados.
- Los alumnos regulares que no aprueben este examen o no opten por él, quedan automáticamente habilitados para el régimen de aprobación con examen final.

Las actividades prácticas consisten en la resolución de problemas planteados en una guía de trabajos prácticos, y se solicita a los alumnos la aplicación de un método de resolución de problemas que requiere en primer lugar la explicitación de la estrategia, la división en módulos bien definidos, y una vez expresada la solución, se finaliza representándola en un código con lenguaje C.

La particularidad de este dictado en las evaluaciones es la siguiente: El primer parcial lo hacen con su par (dada la implementación de la modalidad de trabajo por pares) y utilizando una computadora personal. En tanto que el segundo parcial, se toma en papel y en forma individual, como una forma de asegurar que quienes aprueben la materia tengan los conceptos bien afianzados, de manera que le permitan enfrentar y superar los conceptos de la siguiente asignatura de programación.

El regreso al papel fue muy cuestionado por los alumnos que estaban acostumbrados a trabajar con su computadora, y con todos los recursos en línea, entre ellos, herramientas de IA, como ChatGPT.

La preocupación de los docentes respecto del uso de IA generativa surgió a raíz de la aparición de exámenes parciales “sospechosos” en la instancia del primer parcial (realizado en su notebook con conexión a Internet). Los alumnos debían resolver un problema y subir la solución a una tarea habilitada en el aula virtual de la asignatura. Los docentes observaron que algunos códigos no respetaban las indicaciones dadas en clase o no utilizaban la técnica que se les pedía expresamente. Por ejemplo, resolvieron el problema planteado con vectores, cuando este tema aún no se había desarrollado, y lo usaban de manera incorrecta. No respetaban las pautas del método de resolución de problemas, las soluciones no presentaban una adecuada descomposición en módulos, o el tipo de instrucciones no respondía al lenguaje de programación C, seleccionado por la cátedra para la programación. Por lo que se comienza a pensar que esas resoluciones se habían generado por medio de herramientas de IA.

Esta situación puso en alerta al equipo docente, quienes acordaron un mecanismo interno consistente en convocar a los alumnos que presentaron código “sospechoso” para que expliquen y justifiquen su propuesta de solución. De esta manera se aceptaba el examen de los que supieron explicar lo que hicieron y se desestimaba el examen de los que no podían explicar su propuesta, admitiendo generalmente que lo habían hecho con ChatGPT,

Esta situación se presentó por primera vez en el año 2023. La sorpresa del equipo docente y la falta de estrategias definidas para este tema, llevó a la determinación de volver al esquema tradicional de escritura del código en papel. Si bien se puede tomar como un retroceso, la medida aseguraba el aprendizaje consciente y eficiente de los que lograran superar la instancia evaluativa.

Para el año lectivo 2024, se repitió la propuesta académica de programación de pares, el uso de las rúbricas y las prácticas con dispositivos móviles, advirtiendo y dejando explícito a los alumnos la “prohibición” de usar herramientas de IA para la resolución de las actividades evaluativas. Aún con estas recomendaciones, se encontraron situaciones similares de código que no podían explicar.

Coinciendo con [4] en que “resistirse es inútil” la cátedra se planteó conocer cómo han usado los alumnos las herramientas de IA generativa y cuál es su opinión respecto de esta tecnología y sus expectativas. Para ello se diseñó una encuesta en Moodle, con las preguntas que se describen en el capítulo de Resultados. La encuesta se publicó en el aula virtual a la semana de haber finalizado el dictado de la asignatura. Se envió a los estudiantes por la mensajería del aula virtual y por los canales alternativos de cada comisión (Slack, WhatsApp) el pedido de colaboración para que respondan la encuesta, informándoles que es anónima y dándoles un plazo de 2 semanas para completar la misma. Contestaron 139 estudiantes y los resultados se muestran a continuación.

### 3. Resultados

Analizadas las respuestas de los estudiantes, se obtuvieron los siguientes resultados:

#### 3.1. Datos demográficos:

El perfil de los estudiantes puede determinarse en función de la composición por género y edad. La Fig. 1 muestra que en su gran mayoría son varones (71%) y con edades entre 18 y 24 años (78%) (Ver Fig. 2).

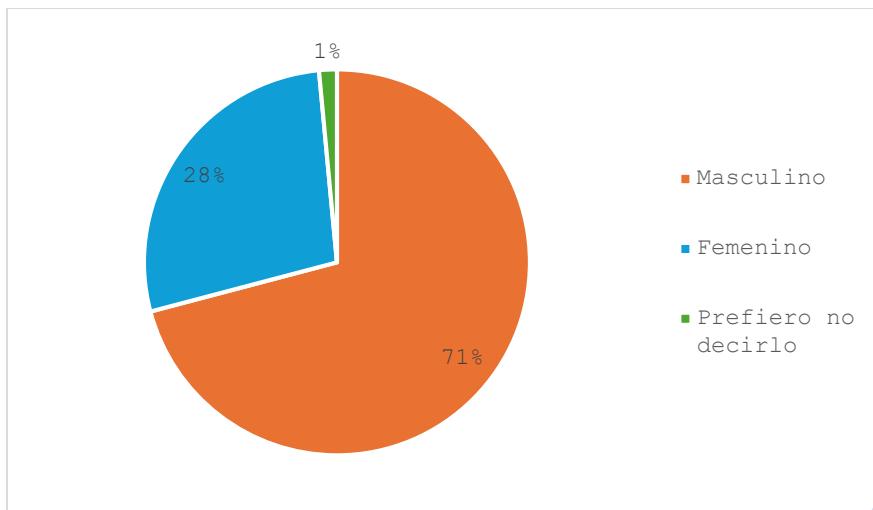


Figura 1: Composición por género

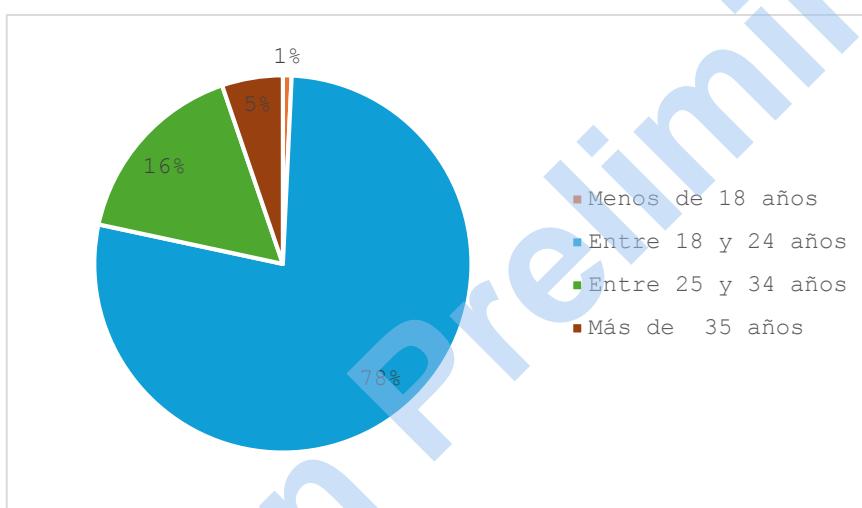


Figura 2: Composición por rango etario

Este perfil se corresponde con las características de los alumnos ingresantes a la carrera en los últimos 3 años.

### 3.2. Uso de la IA

Se realizaron algunas preguntas sobre el uso de herramientas de IA durante el desarrollo de la asignatura.

*P1. ¿Has utilizado alguna herramienta de inteligencia artificial para aprender o mejorar tus habilidades de programación?*

La Fig. 3 evidencia que un alto porcentaje (83%) de los estudiantes reconoce haber utilizado una herramienta de IA durante el cursado de la asignatura.

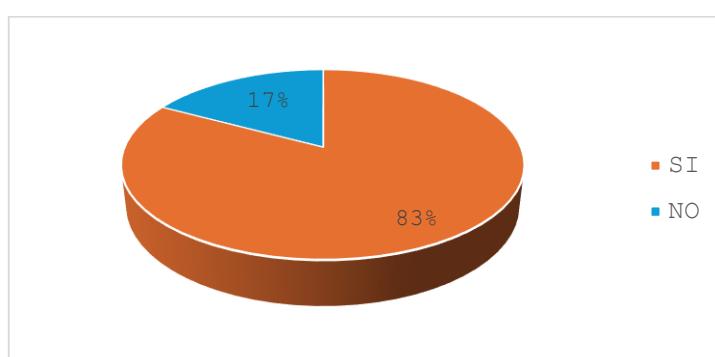


Figura 3: Uso de herramientas de IA durante el cursado de la asignatura

*P2. ¿Qué herramientas de inteligencia artificial has utilizado? (puedes seleccionar más de una opción)*

La encuesta brindó como opciones las herramientas de IA más conocidas, pero podían marcar también otras. La Fig. 4 muestra que las herramientas más utilizadas fueron ChatGPT y Copilot.

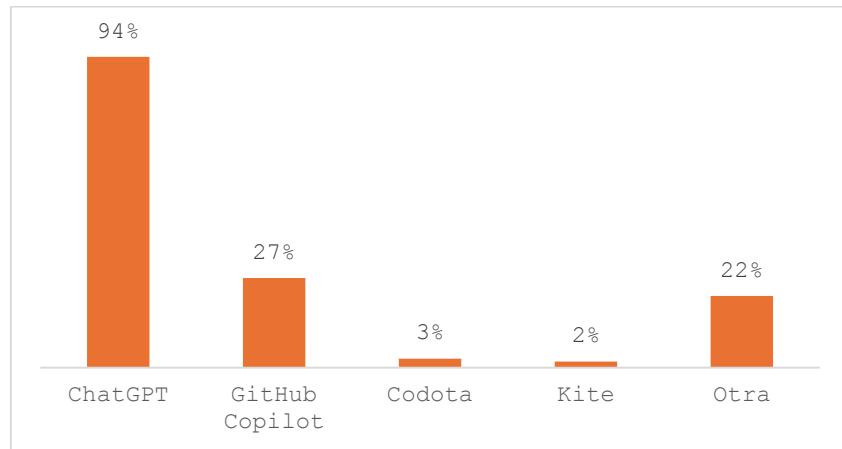


Figura 4: Herramientas de IA utilizadas

P3. ¿En qué aspectos de tu aprendizaje de programación te ha ayudado la inteligencia artificial? (puedes seleccionar más de una opción)

La actividad principal de la asignatura es la resolución de los ejercicios que componen las guías de trabajos prácticos. La ejercitación consiste en resolver los problemas planteados mediante un programa en el que se apliquen los conceptos y técnicas dadas. La tarea es pensar la solución y finalizar representándola mediante un código. En este proceso, los estudiantes indicaron que utilizaron la herramienta de IA para resolver dudas de programación (89%), depurar errores (71%), aprender nuevos conceptos (53%) y mejorar la eficiencia del código (39%), tal como se muestra en la Fig. 5. Cabe señalar que son menos los que han utilizado la herramienta para generar código (18%).

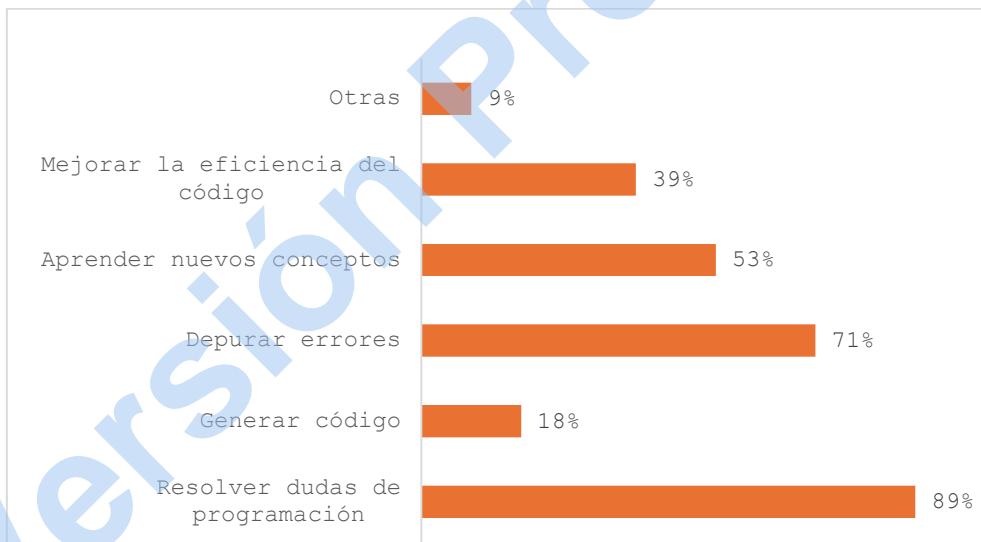


Figura 5: Aspecto del aprendizaje de programación en los que ha contribuido la IA

P4. ¿En qué instancias del proceso de enseñanza y aprendizaje te ha ayudado la inteligencia artificial? (puedes seleccionar más de una opción)

La resolución de problemas se realiza en las clases prácticas, pero también los alumnos tienen que cumplir con entregas de trabajos prácticos obligatorios que tienen que subir a una tarea habilitada en el aula virtual de la asignatura. En los exámenes parciales también resuelven problemas y finalizan representando la solución en un código, pero no tenían permitido usar herramientas de IA en esta instancia. Consultados en qué instancias del proceso de aprendizaje utilizaron IA, mayoritariamente indicaron en la práctica domiciliaria (82%), en las clases prácticas (22%) y en la resolución de los TP que debían subir al aula (20%). Muy pocos señalaron, como era de esperarse, en los exámenes parciales.

Cabe señalar que, si bien estaba “prohibido”, el equipo docente no podía verificar el uso de la IA en la instancia de examen parcial, debido a la masividad y su consecuencia en la distribución de los alumnos en el aula. Para los docentes, la recomendación desde la cátedra fue que, si el código del examen tenía características que permitieran suponer que utilizaron IA, se convocara a los alumnos a un coloquio. Si sabían explicar lo que hacía el código, aprobaban, de lo contrario, desaprobaban. Esto también fue comunicado a los alumnos, previo al examen.

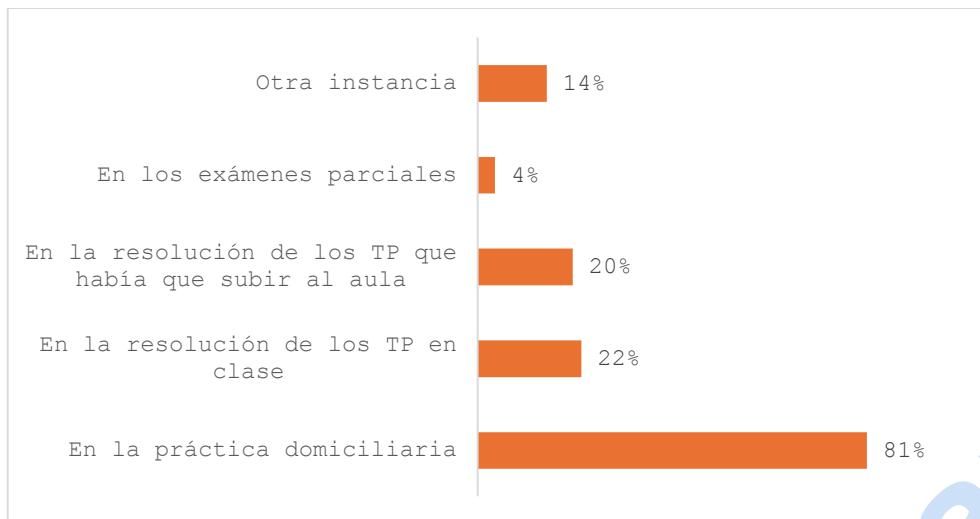


Figura 6: Instancias del proceso de aprendizaje en los que ha contribuido la IA

#### P5. ¿Según tu percepción, cómo crees que ha sido el uso de la IA en la resolución de las actividades de los TP?

En las clases prácticas los alumnos deben resolver los problemas planteados en las guías de trabajos prácticos, trabajando por pares, y compartir y debatir sus soluciones. Por tanto, están enterados, en general, de las dificultades que manifiestan sus compañeros. En este contexto, se les consultó cómo les parecía que sus compañeros usaban la IA para resolver los problemas. Un 14% considera que la mayoría de los estudiantes usaba adecuadamente la IA. En tanto que un 35% considera que pocos estudiantes usaban adecuadamente la IA (35%) y un 40% no se arriesga a emitir una opinión en este sentido. Estos valores, que se muestran en la Fig. 7, dan cuenta de lo novedoso y no consolidado uso de las herramientas de IA en programación.

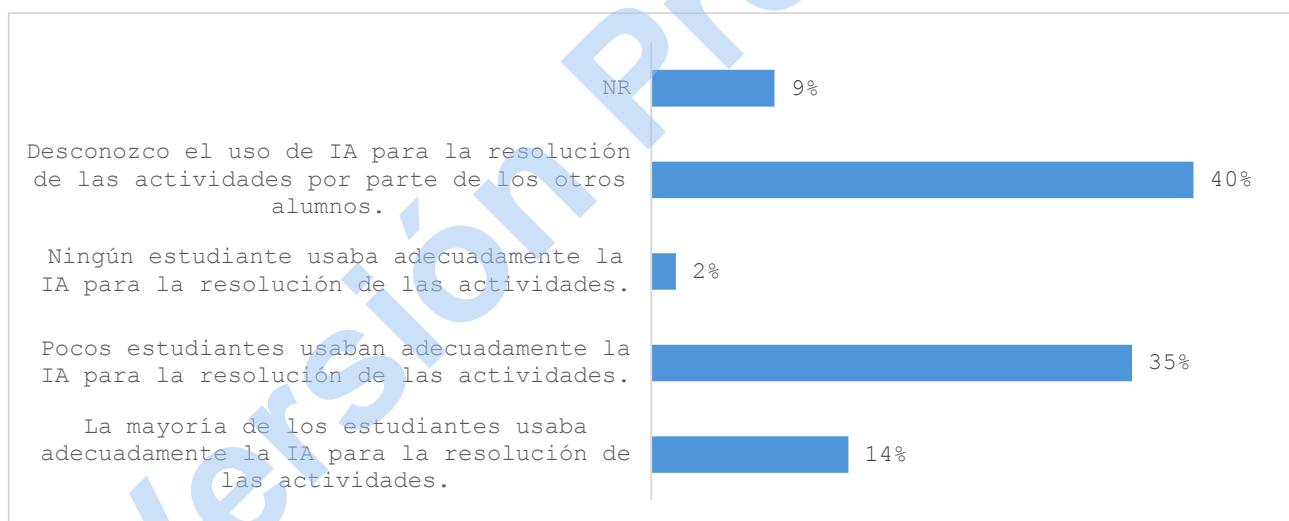


Figura 7: Percepción sobre el uso de IA en las clases prácticas

### 3.3. Efectividad de las herramientas de IA

Como se comentó antes, la asignatura tiene como propósito que los alumnos programen respetando un método de resolución de problemas y siguiendo buenas prácticas en cuanto a la modularización de las soluciones, y haciendo énfasis en la legibilidad del código. Teniendo en cuenta estas cuestiones, se les realizó una serie de preguntas vinculadas con la forma de trabajo que se espera desde la cátedra. Las preguntas fueron;

*De acuerdo con su experiencia:*

P5. ¿Las soluciones brindadas por la IA eran modulares? y esos módulos ¿sólo resolvían una tarea?

El 45% contestó que SI.

P6. ¿Consultaba otro tipo de solución para el mismo problema si no le resultaba adecuada la primera propuesta?

El 77% contestó que SI.

P7. ¿Aplicaba primero la metodología de resolución de problema y después consultaba la IA?

El 82% contestó que SI.

P8. ¿Escribía directamente el problema en la IA sin considerar la metodología de resolución de problema?

El 9% contestó que SI.

Estas respuestas permiten percibir que los estudiantes necesitan instrucción para el prompting, para adaptar las respuestas según las necesidades específicas de la tarea o pregunta.

Se destaca también que han respetado la aplicación del método de resolución de problemas dado que el 91% contestó que No en la pregunta 8.

P9. Califica la efectividad de la inteligencia artificial en tu aprendizaje de programación (1 = Nada efectivo, 5 = Muy efectivo)

Para sintetizar el grado de aprovechamiento que hicieron de la herramienta de IA, se les pidió que califiquen la efectividad de la inteligencia artificial en su aprendizaje, con un valor de 1 a 5 (Escala de Likert). En la Fig. 8 se puede apreciar que un 42% de los estudiantes calificó con un valor 3 (indiferente o nulo aporte), un 32% con 4 y un 14% con 5. Esta respuesta es muy significativa, dado que aporta la propia visión del alumno respecto a su aprendizaje.

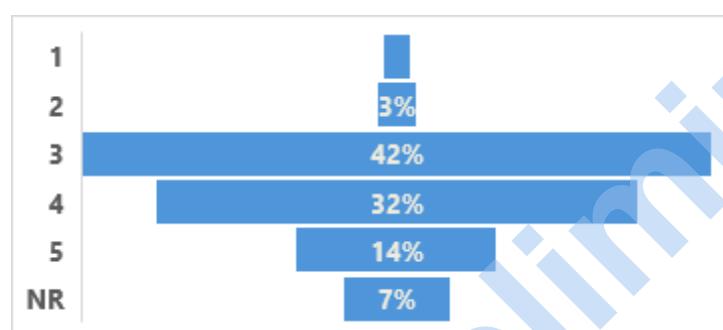


Figura 8: Efectividad de la IA en el aprendizaje de programación de los estudiantes (Escala Likert)

P10. ¿Cuáles son los principales desafíos que has enfrentado al usar inteligencia artificial para aprender programación? (puedes seleccionar más de una opción).

Los estudiantes señalaron como principales dificultades la falta de precisión de las respuestas (52%), la falta de adecuación al método de resolución de problemas (47%) y la falta de confianza en la calidad del código generado (41%).

Estos valores refuerzan la idea de que los estudiantes necesitan instrucción para la formulación de las preguntas a la IA.

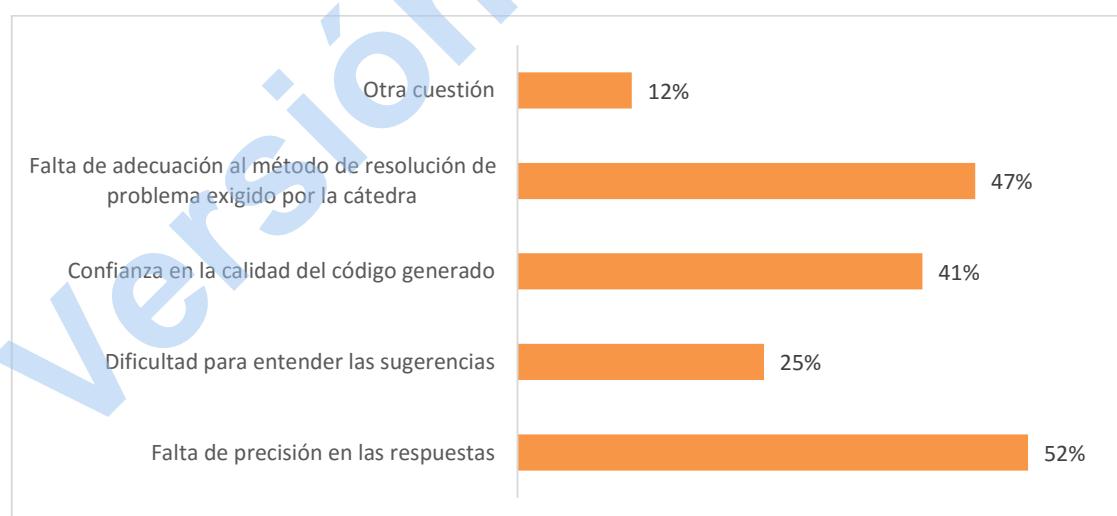


Figura 9: Principales desafíos de la IA para aprender programación

### 3.4. Importancia de la IA

Estas preguntas debían contestar todos los alumnos aun cuando no hayan usado herramientas de IA en el cursado de la asignatura.

P11 ¿Consideras que el uso de la inteligencia artificial será una habilidad esencial para futuros programadores?

En la Fig. 10 se puede apreciar que la mayoría de los estudiantes (69%) considera que la IA será una habilidad esencial para futuros programadores.

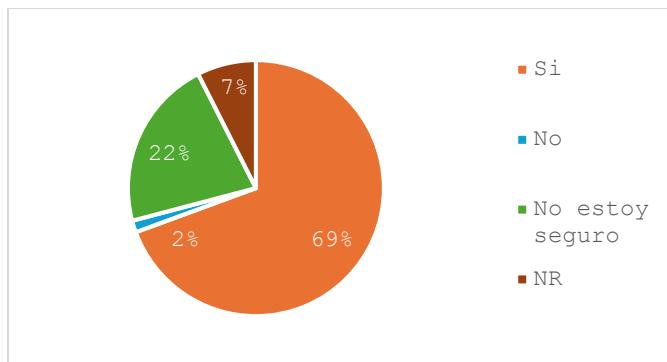


Figura 10: Importancia de la IA para los programadores

*P12 ¿Crees que la asignatura debería incorporar el uso de inteligencia artificial en la enseñanza de la programación?*

Un 60% de los estudiantes considera que debería incorporarse la IA en la enseñanza de la programación, mientras que No están seguros (23%) o dicen que No o No responden.

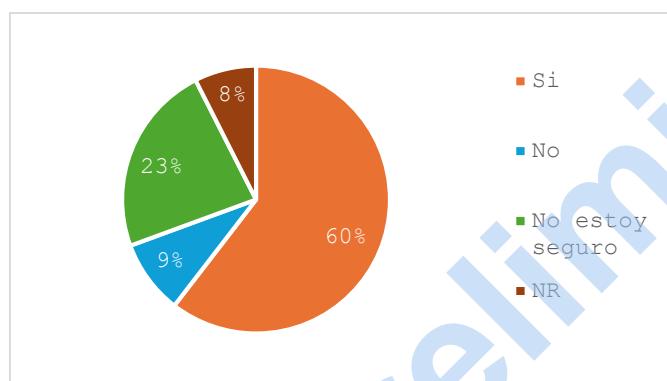


Figura 11: Incorporar IA en la enseñanza de la programación

## 4. Conclusiones

A partir de los resultados obtenidos, el equipo docente se plantea el desarrollo de una metodología de enseñanza que integre el uso de la inteligencia artificial (IA) como parte del proceso de enseñanza-aprendizaje. Se reconoce que dominar la programación sigue siendo esencial, y que las herramientas conceptuales clave, como la abstracción, la descomposición funcional y la legibilidad del código, son elementos innegociables en la formación. Sin embargo, también se valora el potencial de la IA como una herramienta de apoyo en el desarrollo de software, particularmente en tareas como la depuración de código, la corrección de errores sintácticos y la aceleración del proceso de compilación.

Los resultados reflejan que la mayoría de los estudiantes ha empleado herramientas de IA generativa para complementar su aprendizaje. No obstante, han enfrentado dificultades para obtener soluciones que se alineen con las pautas específicas de la asignatura. Esta situación pone de manifiesto la necesidad de una instrucción formal que enseñe a los estudiantes a formular preguntas adecuadas (*prompting*) para obtener soluciones más precisas y coherentes. Dado que el *prompting* puede considerarse el nuevo SQL, es crucial que los estudiantes adquieran las habilidades necesarias para generar código de calidad, cumpliendo con las buenas prácticas y los criterios establecidos, asegurando que el software sea legible, comprensible y fácil de modificar.

En conclusión, se propone elaborar una metodología que incorpore e integre de manera efectiva las herramientas de IA en la enseñanza de la programación, buscando maximizar sus beneficios sin comprometer el desarrollo del pensamiento computacional y las habilidades de abstracción, indispensables para una formación sólida en el campo de la programación.

## Referencias

- [1] P. Rodríguez, *Inteligencia artificial*. España: Ediciones Deusto, 2018.
- [2] A. García Serrano, *Inteligencia artificial. Fundamentos, prácticas y aplicaciones*. España: RC Libros, 2013.
- [3] Grupo Iberdrola, Hitos históricos en la evolución de la Inteligencia Artificial, [Online]. Available: <https://www.iberdrola.com/innovacion/evolucion-inteligencia-artificial>
- [4] S. Lau, P. J. Gou, "From "Ban It Till We Understand It" to "Resistance is Futile": How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot", in Proceedings of the 2023 ACM Conference on International Computing Education Research, New York, NY, USA, 2023.

- [5] F. J. García-Peñalvo, "Generative Artificial Intelligence: New Scenarios in Teaching, Learning, and Communication," VIII Congreso Internacional de Estudios sobre Medios de Comunicación. Universidad Complutense de Madrid, España, 6 de septiembre de 2023. Available from: <https://bit.ly/3sGUA3Y>.
- [6] D. Sun, A. Boudouaia, C. Zhu, "Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study". International Journal of Educational Technology in Higher Education, vol 2, N° 14, 2024
- [7] R. Yilmaz, F. G. K. Yilmaz, "The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation". Computers and Education: Artificial Intelligence, Volume 4, 2023.
- [8] B. Becker, B. Mac Namee, "Generative AI in Computing Education: Wrecking Ball or Holy Grail?" [Online]. Available: <https://www.ucd.ie/cs/blog/generativeaiincomputingeducationwreckingballorholysgrail/>
- [9] G. Dapozzo, C. Greiner, R. Petris, A.M. Company, M. C. Espíndola, "Nuevo contexto en las carreras de informática. ¿Con qué estrategias lo enfrentamos?" In JADiCC 2023, realizado en la Universidad Nacional del Comahue, Neuquén. Argentina. Diciembre 2023 [Online]. Available: <https://jadicc2023.program.ar/trabajos-cientificos/>
- [10] G. Dapozzo, C. Greiner, R. Petris, A.M. Company, M.C. Espíndola. "Teaching Strategies for Programming in Massive University Settings" In *CACIC 2023. Communications in Computer and Information Science*, vol 2123. Ed. Springer, 2024, pp 339–349.

# La construcción de estadísticas relativas al acceso a saberes digitales en Argentina

Cecilia Martínez<sup>1</sup>, Araceli Acosta<sup>2</sup>

<sup>1</sup> IDH CONICET y FFYH, UNC., Córdoba, Argentina

<sup>2</sup> FAMAF, UNC. Córdoba, Argentina

[cecmart@gmail.com](mailto:cecmart@gmail.com), [mail2@servidor.com](mailto:mail2@servidor.com)

## Resumen

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, la brecha de acceso y apropiación saberes - y de las condiciones necesarias para distribuir ese saber- no ha sido monitoreado y por tanto no ha sido visibilizada. Este artículo es parte de un proyecto más amplio que busca elaborar indicadores y desarrollar herramientas que permitan recuperar y visibilizar datos estadísticos que den cuenta de las brechas digitales y computacionales. Estos datos son necesarios para entender el alcance de la problemática de estas brechas y para pensar en programas y políticas focalizados. En este artículo presentamos los avances de una herramienta que permite aproximarse a visibilizar a la población escolar accede a saberes computacionales.

**Palabras clave:** Brecha digital, brecha computacional, acceso, enseñanza de computación

## 1. 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, la brecha de acceso y apropiación saberes - y de las condiciones necesarias para distribuir ese saber- no ha sido monitoreado y por tanto no ha sido visibilizada.

Respecto de la brecha de acceso a dispositivos, se ha publicado, por ejemplo, que durante la pandemia en Argentina solo un 40% del quintil inferior de ingresos pudo acceder a una computadora y a conectividad contra el 100% del quintil superior [1]. Asimismo se han identificado diferencias de continuidad con el proceso educativo entre quienes acceden y saben usar la tecnología digital y quienes no [2]. A partir de esta visibilización del problema se han generado programas y políticas de equipamiento, conectividad y capacitación en el uso de software.

De la brecha de saberes digitales y computacionales y de la capacidad instalada que tiene el sistema educativo en ofrecer estos saberes (espacios curriculares, docentes formados, materiales didácticos, etc.) se ha investigado muy poco. Recuperando a Bell [3], definimos brechas de saberes digitales computacionales como la desigualdad de acceso a conocimiento que permite comprender cómo funciona una computadora en su mínima expresión, para entender alcances y limitaciones, y cómo se puede crear tecnología a partir de lenguajes y conceptos del campo de la computación. Cómo se distribuyen estos saberes en las escuelas de Argentinas está escasamente documentado, como así también qué grupo poblacional recibe estos contenidos en escuelas con orientación en alguna rama relativa a la computación (tales como informática, programación, desarrollo de software, etc). Conocer esta información permitiría desarrollar políticas y programas educativos focalizados que reduzcan estas brechas que se generan a partir de diferentes posiciones sociales, saberes previos y géneros.

A pesar de que no se han sistematizado los datos escolares para dar cuenta de esta información, existen situaciones que sugieren que la brecha de saberes digitales y computacionales es profunda y que se traduce en problemáticas para el ejercicio de la ciudadanía en un sistema democrático. Aquí nombramos algunos ejemplos de situaciones cotidianas que indican un escaso dominio de los saberes computacionales. Por caso, durante la pandemia la ciudadanía debatió sobre el uso sanitario de plataformas que operan con algoritmos de macro datos para trazabilidad y geolocalización, capacidad de

detección y prevención temprana [4]. Según la Organización Panamericana de la Salud estos sistemas estiman casos y también recursos necesarios como camas de terapia intensiva y respiradores. Esto generó cuestionamientos sobre el uso de datos privados por el sector público lo cual resulta paradójico considerando el escaso cuestionamiento de la utilización de los mismos datos con fines comerciales por parte de las corporaciones. Del mismo modo se está promoviendo el uso de tecnología con Inteligencia Artificial sin abordar cómo funciona, de donde se toman los datos, y los riesgos relativos a los sesgos que producen. Como caso paradigmático el Gobierno de Países Bajos condenó a más de 20.000 familias por cometer ilícitos contra el estado en la recepción de subsidios. El informe de Amnistía Internacional “Máquinas Xenofobicas” expuso cómo los algoritmos utilizados para detectar fraude tenían sesgos raciales. Como resultado, los ciudadanos neerlandeses pertenecientes a minorías étnicas tenían más chances de ser acusados.

Asimismo, muchas escuelas tienen computadoras en desuso que podrían recuperarse simplemente a partir de la actualización del sistema operativo o la configuración de redes. La falta de saberes de tecnología digital y computacional sigue siendo un obstáculo para apropiarse del equipamiento existente.

En términos de seguridad informática, los ciberdelitos crecieron un 300% en estos últimos dos años. En Argentina, se registraron incrementos de home banking, mobile banking, pagos remotos y un incremento del comercio electrónico. Diversos reportes nacionales dan cuenta de un crecimiento exponencial de la ciberdelincuencia, proporcional al volumen de maniobras ligadas a la informática que tienen lugar. Un gran porcentaje de estos ataques afectaron jóvenes a través de delitos de extorsión y acoso sexual [5]. Parte de esta problemática se podría explicar porque son pocos los jóvenes que reciben en su escolaridad obligatoria saberes que les permiten comprender cómo funcionan los sistemas computacionales, sus limitaciones y potencialidades. Estudios de la Fundación Sadosky que analizan la cantidad de estudiantes que egresan de secundaria con orientación en informática calculan que la población que recibe oferta de saberes digitales y computacionales no llegaría al 3% [6]. Este porcentaje coincide globalmente con lo hallado en el Estudio Internacional de Alfabetización digital y computacional [7] que reporta que si bien el 82% de los estudiantes puede navegar por internet, manipular imágenes y manejar accesos con claves, sólo el 2% de los participantes pueden comprender cómo funciona una computadora y crear nuevos artefactos a partir de saberes computacionales. Es decir, es bajo el porcentaje de estudiantes de los niveles obligatorios que dominan saberes de computación que les permite entender, leer, usar, remezclar y participar de la construcción de artefactos digitales. Muchas investigaciones relacionan estos números con la baja matrícula en carreras tecnológicas y consecuentemente con el insuficiente desarrollo de tecnología de manera soberana.

Desde 2020 conjuntamente con la Fundación Sadosky y con subsidios de la Fundación Adela y de UNESCO, hemos relevado programas y políticas educativas en diferentes regiones del mundo que promueven la inclusión de saberes de computación en las escuelas. Estos estudios mostraron que hay pocos indicadores sobre las condiciones del sistema educativo en relación a las brechas digitales, la información está dispersa entre diferentes relevamientos estadísticos de cobertura parcial que no se integran entre sí. Ante la predominancia de discursos pedagógicos que abordan la brecha de acceso a dispositivos y la ausencia de investigaciones que analizan las brechas de saberes, se genera una falsa apariencia de que el problema de la brecha digital se reduce al acceso a dispositivos. La ausencia de datos esconde un problema educativo que incluye además de la brecha de saberes, las condiciones del sistema para enseñar estos saberes.

En este contexto, proponemos desarrollar un marco conceptual y un primer relevamiento para un Observatorio de Brechas de Saberes Digitales y Computacionales. El principal propósito de este observatorio es generar una herramienta de seguimiento de este fenómeno que permita ofrecer un sistema de indicadores articulados bajo un marco conceptual coherente y pertinente, para visibilizar las brechas de saberes digitales en las escuelas de Argentina. Información sobre los saberes que ofrece la escuela, la cantidad de docentes formados, las matrículas en los institutos de formación docente, la cantidad de estudiantes que eligen escuelas con orientación en la temática, la disparidad de género, los materiales didácticos disponibles, las normativas curriculares, etc; son datos necesarios para entender la problemática de las brechas de saberes y poder pensar en programas y políticas focalizadas.

## 2. Antecedentes

Recuperando a varios autores, 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 [8, 9, 10].

En relación al ámbito escolar, dos referentes precisan la idea de brecha digital para el ámbito escolar. DiSessa [8] va a referirse a brecha de alfabetización digital y Jeannette Wing a pensamiento computacional [10]. 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 [18] 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. Para estos autores, las principales demandas cognitivas que requiere la alfabetización computacional son la abstracción (que requiere identificar las variables o elementos esenciales de un problema para convertirlos en objetos que la computadora pueda manipular), la 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.

Respecto a la medición de estas brechas, muchas investigaciones cualitativas centradas en el análisis de casos han mostrado que los sistemas educativos ofrecen computación de manera segregada (solo para algunas escuelas y para algunos estudiantes) [11, 12]. Estas investigaciones no precisan sobre el foco de la alfabetización computacional en términos de aprendizaje de conceptos particulares. Generalmente las escuelas ofrecen computación como orientaciones optativas que el estudiantado elige en los últimos años del secundario. Este es un fenómeno frecuente en diferentes países. Para el caso de Estados Unidos, por ejemplo, Margolis y Goode documentan que son los estudiantes varones con capitales digitales previos quienes seleccionan estas orientaciones a través de materias optativas en el secundario. Esta situación contribuye a reproducir los capitales digitales de origen y a reproducir la brecha de género en el sector [13].

Análisis estadísticos internacionales muestran que la brecha digital por género y por posición socioeconómica en el campo social es particularmente prominente en países en desarrollo. En América Latina de los usuarios de internet solo el 25% son mujeres. De los graduados en carreras de tecnología solo un 18% son mujeres [14, 15, 16]. A nivel mundial, la proporción de mujeres que trabajan en carreras de computación es tres veces más baja comparado con otras ocupaciones y solo el 2% de patentamientos en tecnologías provienen de mujeres [17]. Suárez y Yoguel [19] sugieren que los niveles de productividad en América Latina tuvieron dificultades para adoptar nuevas tecnologías debido a los cambios y la curva de aprendizaje que implica un contexto caracterizado por mercados inestables. Dentro de este contexto más amplio —y reconociendo cómo la tecnología digital es central no solo para la producción económica sino también en otras áreas que mejoran la calidad de vida de una población— los países latinoamericanos están introduciendo la informática en su currículo escolar obligatorio. Si bien este tipo de mediciones son indicadores de desigualdad, no dan cuenta de la producción de estas brechas en el ámbito escolar.

## 2. 1 Antecedentes específicos de mediciones Alfabetización Computacional en el ámbito escolar

La brecha digital entre diferentes grupos socioeconómicos en edad escolar en América Latina es particularmente problemática. Trucco [20] comparó el uso de computadoras entre los estudiantes de secundaria que participaron en la prueba PISA (Program for International Student Assessment). Si bien los diferentes tipos de usos se distribuyen equitativamente entre los cuartiles económicos y socioculturales en la mayoría de los países analizados, en América Latina los usuarios frecuentes pertenecen al tercer y cuarto cuartil superior y muy pocos pertenecen al primer cuartil. Si bien el uso no garantiza alfabetización computacional, entendemos que es una condición necesaria y por tanto un indicador importante para relevar.

En este contexto, muchos países están incluyendo saberes de computación para todos los estudiantes en los niveles obligatorios del sistema. La Fundación Sadosky ha realizado un relevamiento sobre los países y particularmente las provincias Argentinas que incluyen las Ciencias de la Computación en sus sistemas educativos permitiendo visualizar en mapas si los contenidos se ofrecen de manera obligatoria u optativa. Esta información permite mapear la oferta pero no el alcance en términos de estudiantes y está disponible en este enlace <https://program.ar/observatorio/>. Nos preguntamos en este trabajo cómo se están estudiando estos procesos desde el punto de vista del acceso de los estudiantes a estos saberes.

En Estados Unidos, Margoli y Goode [21] han tomado como indicador de acceso a estos saberes la cantidad de estudiantes de nivel secundario que rinde un exámen de Computación Avanzada (Advanced Placement in Computer Science). Estos exámenes permiten ingresar a diferentes cursos en el nivel superior. Pero no tienen datos a nivel de escuela de cuántos estudiantes toman cursos de computación de manera electiva. En la mayoría de los estados, la computación no es una materia obligatoria para terminar el secundario.

Otra forma de medir la brecha de alfabetización computacional es medir aprendizajes. Para comprender cómo se han medido los aprendizajes en alfabetización computacional realizamos un breve relevamiento de tests de pensamiento, alfabetización y participación computacional que hayan sido implementados en poblaciones en edad escolar, en el período de escolaridad obligatoria. Buscando de manera sistemática en repositorios académicos, se hallaron 14 artículos que reportaban la implementación de tests; sólo siete tests de los analizados cumplían con los requisitos de abordar alfabetización computacional en el ámbito de la escolaridad obligatoria (presentados en el Anexo 1). En general encontramos que las variables, operaciones y los instrumentos que se han usado para medir y analizar la alfabetización

computacional dependen de cómo ésta ha sido definida. Desde la concepción más restrictiva de alfabetización computacional como pensamiento computacional se proponen una serie de test que ofrecen ejercicios para recuperar la misma lógica de razonamiento que se utiliza en la programación [22]. Como caso paradigmático, los tests del Programa para la Evaluación Internacional de los Estudiantes (habitualmente conocido como PISA por sus siglas en inglés: *Programme for International Student Assessment*) agregaron, a partir del 2021, una sección de pensamiento computacional aplicada exclusivamente a la matemática para resolver problemas de este campo. Más precisamente incluyen reconocimiento de patrones, descomposición, selección de herramientas para resolver problemas y escritura de algoritmos para definir una solución [23]. Otros tests siguen esta misma línea [24, 25] y los exámenes de Advanced Placement en Computación (ingreso a cursos avanzados) requieren analizar, diseñar y escribir programas y subprogramas [26].

Se observan varias limitaciones con este tipo de test. La primera radica en reducir el pensamiento computacional a ejercicios de programación, lo que requiere necesariamente que el estudiante tenga dominio de un lenguaje para resolver el problema. Para abordar esta dificultad, algunos investigadores han desarrollado tests con independencia de un lenguaje. Por ejemplo, existe una propuesta que consiste en siete ideas “poderosas” del campo de las Ciencias de la Computación que son apropiadas para niños de entre 4 y 9 años [27]. Estos incluyen otros dominios además de la programación, tales como hardware y software, modularidad, algoritmos, estructuras de control, representaciones, depuración y diseño.

La segunda limitación observada en el universo de tests analizados se relaciona con la restricción de los saberes de alfabetización computacional al pensamiento computacional. Al suceder este sesgo, se dejan de lado saberes, reflexiones y conceptos relevantes propios de las Ciencias de la Computación. Por eso, proponemos hablar de prácticas computacionales en vez de pensamiento computacional, para dar cuenta no solamente de procesos individuales, sino de reflexiones, modos de pensar y resolver problemas en un campo de conocimiento particular [27]. En este sentido, algunos test se focalizan en grandes ideas de la disciplina relacionadas con pensamiento computacional, pero no limitadas al pensamiento individual. Estas son: abstracciones, manipulación de datos e información, algoritmos, artefactos computacionales y prácticas computacionales. Este ejemplo requiere que el estudiantado responda a soluciones en diferentes escenarios y que expliquen su razonamiento en prácticas computacionales más auténticas. Esta práctica resulta superadora a otras analizadas.

Solo dos países en la región latinoamericana están evaluando los saberes computacionales de sus estudiantes. Por un lado, Costa Rica implementó un test que analiza competencias generales de resolución de problemas, comunicación y productividad de estudiantes de nivel secundario. Los resultados mostraron que además del capital cultural de las familias, el número de años que los estudiantes habían recibido clases del programa de Informática se relacionaba positivamente con el rendimiento en las competencias evaluadas.

El informe también mostró que el programa de informática mejoró la autopercepción de competencias, entre otros indicadores de aprendizaje. Durante las entrevistas con quienes coordinan el programa de Informática se pudo establecer que en épocas donde estuvo en riesgo la subsistencia del programa, estas evaluaciones y el apoyo del público fueron claves para su continuidad.

Por otro lado, en 2020, en Uruguay, Ceibal<sup>1</sup> evaluó a las y los estudiantes usando el test internacional Bebras Challenge<sup>2</sup>. A partir de una muestra de estudiantes, la evaluación encontró que la participación en el programa de Pensamiento computacional estaba correlacionado con aprendizajes en el área de programación. Asimismo, el informe “Ceibal en Números” analizó la escala y cobertura del programa. Aproximadamente 30% de las y los estudiantes de 9 a 12 años en el país han recibido lecciones de Pensamiento Computacional. Después de cada lección, se ofrecen evaluaciones en una plataforma en línea. Todos estos datos informan el programa y permiten comprender los desafíos y aprendizajes de computación para realizar los cambios necesarios y mejorar la enseñanza.

<sup>1</sup> Ceibal es el centro de innovación educativa con tecnologías digitales del Estado uruguayo. Promueve la integración de la tecnología a la educación con el fin de mejorar los aprendizajes e impulsar procesos de innovación, inclusión y crecimiento personal.

<sup>2</sup> <https://www.bebraschallenge.org/>

Ahora bien, desarrollar pruebas, administrarlas y analizarlas tiene un costo muy alto para las jurisdicciones, en dinero y en tiempo. Sin embargo, Argentina cuenta con diversas estadísticas educativas que se toman desde diferentes organismos con frecuencia anual. Nos preguntamos entonces si esta información que ya está disponible podría dar cuenta de algunos indicadores aproximados de alfabetización computacional en términos de acceso de la población estudiantil a estos saberes. Con este propósito realizamos este trabajo.

### 3. Metodología

Para definir el alcance y potencial de las estadísticas actuales de ofrecer indicadores de alfabetización computacional, se elaboró una lista de bases de datos estadísticos públicos que incluyen información relativa a las brechas de saberes computacionales con el objeto de identificar las dimensiones que pueden ser representadas con los datos existentes y aquellas sobre las que hay vacancias. En este listado se identificaron los tipos de datos, fuentes, herramienta de medición, frecuencia de publicación y posibilidades de acceso.

Siguiendo a [28] “*Es básico y fundamental para la creación de un observatorio precisar los espacios de observación y delimitar qué información se requiere con relación a sus objetivos; además de tener muy claro qué se puede y qué se quiere hacer con esa información. De otro modo, será mayor el costo que el beneficio de la captación y almacenamiento de información y escaso su valor instrumental en la generación de conocimiento útil a sus propósitos*”.

Para elaborar los indicadores se recuperan los pasos metodológicos sugeridos que forman parte del proceso de operacionalización para la medición de variables sociales [29]. En la tabla 1 detallamos las bases de datos estadísticas seleccionadas y los indicadores que pudimos construir a partir de ellas. Se relevaron otras bases tales como las pruebas ERCE (Estudio Regional Comparativo y Explicativo de la UNESCO) pero no se encontró información relevante para esta temática. Las principales bases con información pertinente a nuestro objeto fueron:

**Relevamiento Anual:** Es un operativo educativo de carácter censal, cuya unidad de relevamiento y análisis son las unidades educativas. Se indaga sobre las condiciones para enseñar y aprender y se censa la población estudiantil y docente.

**Anuarios estadísticos:** Resumen la información estadística de las principales variables del Relevamiento Anual (RA). Permite disponer de información referida a establecimientos, alumnos, cargos docentes y horas cátedra del sistema de educación argentino, agregados por provincia. En general ofrecen datos de la cobertura del sistema educativo en los diferentes niveles, modalidades y orientaciones.

**Pruebas Aprender:** Son construidas por el Ministerio de Educación Nacional y evalúan el grado de dominio que las y los estudiantes de nivel primario y secundario tienen sobre un recorte específico de contenidos (matemáticas, lenguas, ciencias sociales y ciencias naturales) durante su trayectoria escolar, los factores sociodemográficos y condiciones en que se enseña y se aprende.

**Pruebas Pisa:** Es el Programa para la Evaluación Internacional de Estudiantes (PISA es la sigla en inglés) que lleva adelante la Organización para la Cooperación y el Desarrollo Económicos (OCDE) cada tres años, desde el 2000. Argentina formó parte del estudio en todas las aplicaciones, a excepción del año 2003.

Tabla 1. Indicadores y bases

Dimensión	Indicador construido	Base	Frecuencia de Public.	Elaboración
Orientación	Porcentaje de estudiantes registrados en orientaciones relativas a la informática sobre el total de alumnos que cursan el ciclo orientado en los últimos 10 años	<a href="#">Anuarios</a>	Anual	Nacional
	Porcentaje de estudiantes registrados en orientaciones relativas a la informática en comparación con el porcentaje de estudiantes que estudian otras orientaciones. en los últimos 10 años			
Ámbito	Porcentaje de estudiantes egresados en Ciencias Aplicadas a Tecnología sobre el total de alumnos en el ciclo orientado en el ámbito estatal. en los últimos 10 años	<a href="#">Anuarios</a>	Anual	Nacional
	Porcentaje de estudiantes egresados en Ciencias Aplicadas a Tecnología sobre el total de alumnos en el el ciclo orientado en el ámbito privado en los últimos 10 años			
Zona	Porcentaje de estudiantes que estudian informática sobre el total de alumnos en el ciclo orientado en el ámbito rural y urbano. en los últimos 10 años	<a href="#">Anuarios</a>	Anual	Nacional

Género	Cantidad de mujeres egresadas en Ciencias Aplicadas en Tecnología en los últimos 10 años	<a href="#">Anuarios</a>	Anual	Nacional
	Cantidad de varones egresados en Ciencias Aplicadas en Tecnología en los últimos 10 años	<a href="#">Anuarios</a>	Anual	Nacional
	Cantidad de no binarios en los últimos 10 años	<a href="#">Anuarios</a>	Anual	Nacional
Modalidad	Porcentaje de estudiantes de escuelas técnicas que eligen informática o programación por sobre el total de estudiantes registrados en el ciclo orientado. en los últimos 10 años	<a href="#">Anuarios</a>	Anual	Nacional
Permanencia y Egreso	Porcentaje de egresados en Ciencias Aplicadas a Tecnología de nivel secundario sobre el total de egresados en los últimos 10 años	<a href="#">Anuarios</a>	Anual	Nacional
	Porcentaje de egresados con título de informática en escuelas técnicas sobre el total de egresados de técnicas en los últimos 10 años	<a href="#">Anuarios</a>	Anual	Nacional
Acceso a programación	Cantidad de estudiantes que programan por sobre el total de estudiantes del mismo nivel educativo en cada provincia	<a href="#">Aprender</a>	Anual	Nacional
	Cantidad de estudiantes que programan en comparación con otras actividades realizadas en el tiempo libre.	<a href="#">Aprender</a>	Anual	Nacional
	Cantidad de estudiantes que programan en un lugar tranquilo para estudiar	<a href="#">Aprender</a>	Anual	Nacional
Oferta de actividad	Cantidad de estudiantes y frecuencia de participación en diferentes actividades escolares	<a href="#">PISA</a>	c/ 3 años	Internac.
Acceso a actividad	Cantidad de estudiantes y frecuencia de participación de diferentes actividades no escolares	<a href="#">PISA</a>	c/ 3 años	Internac.
	Cantidad de estudiantes y frecuencia de participación en tareas matemáticas	<a href="#">PISA</a>	c/ 3 años	Internac.
Confianza	Niveles de confianza para la realización de tareas matemáticas	<a href="#">PISA</a>	c/ 3 años	Internac.
Capacidad Escolar	Cantidad de docentes de informática por estudiantes	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de laboratorios de informática por cantidad de estudiantes	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de computadoras con fines pedagógicos por cantidad de estudiantes	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Equipamiento Establecimiento - Servidor para uso escolar	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Equipamiento Establecimiento - Impresora 3D	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Equipamiento Establecimiento - Pizarras digitales	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Internet - Tipo de servicio - Gratuito	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Internet - Tipo de servicio - Pago	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Internet - Tipo de conexión - Telefónica	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Internet - Tipo de conexión - ADSL	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Internet - Tipo de conexión - Cable Módem	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Internet - Tipo de conexión - Satelital	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Internet - Tipo de conexión - Otro Tipo	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Internet - Tipo de conexión - Móvil	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Espacios con conexión - En el área de administración	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Espacios con conexión - En las aulas	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Espacios con conexión - En la biblioteca/s	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Espacios con conexión - En otro espacio del establecimiento	<a href="#">Relevam</a>	Anual	Nacional
Indicadores campo social	Cantidad de escuelas por sector y ámbito con Espacios con conexión - En el laboratorio	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Dispone de sala o laboratorio de informática - Si	<a href="#">Relevam</a>	Anual	Nacional
	Cantidad de escuelas por sector y ámbito con Laboratorio - Funciona es espacio exclusivo? - Si	<a href="#">Relevam</a>	Anual	Nacional
	Beneficiarios de alimentación gratuita - Desayuno	<a href="#">Relevam</a>	Anual	Nacional
	Tenencia de Computadora primaria	<a href="#">Relevam</a>	Anual	Nacional
	Tenencia de Computadora secundaria	<a href="#">Relevam</a>	Anual	Nacional

Este trabajo requirió de explorar cada una de las bases y consultar con expertos. En efecto, cada una de las bases incluye conjuntos de datos muy grandes y es necesario analizar cada tabla y sus variables para construir los indicadores dentro de las posibilidades de información disponible.

Con el objetivo de automatizar parte de la tarea de actualización de la información, se desarrolló un script para que extraiga la información requerida de cada base, filtre la información necesaria, complete los datos faltantes cuando sea necesario, y genere una nueva tabla incluyendo información del año. Con la información procesada, se generó una herramienta preliminar de visualización que incluye filtros por año y por provincia, para un análisis más detallado. Al momento de escribir este artículo, solo avanzamos con el procesamiento de datos de la base del Relevamiento Anual.

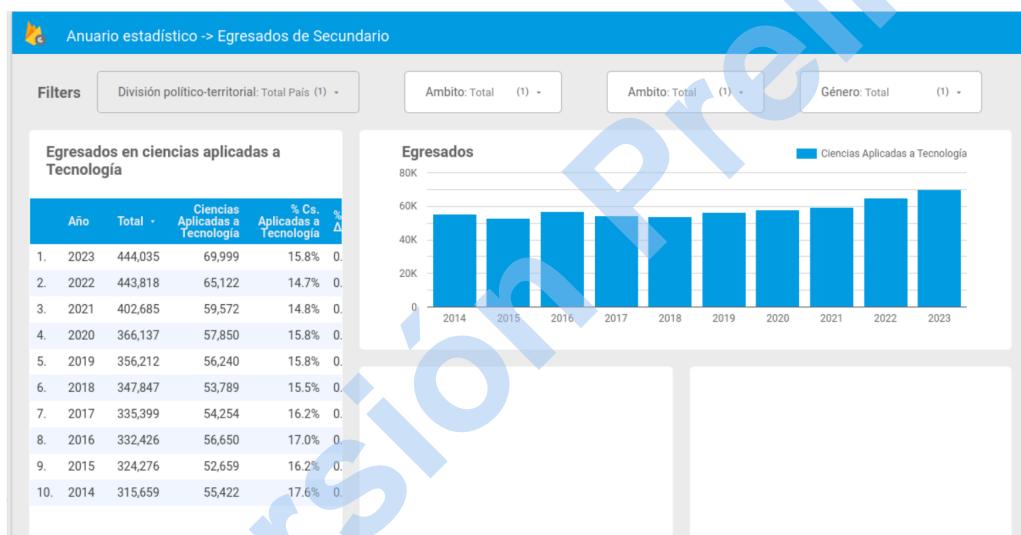
## 4. Resultados Preliminares

### a) Rápida visualización de datos relativos a las brechas de saberes computacionales

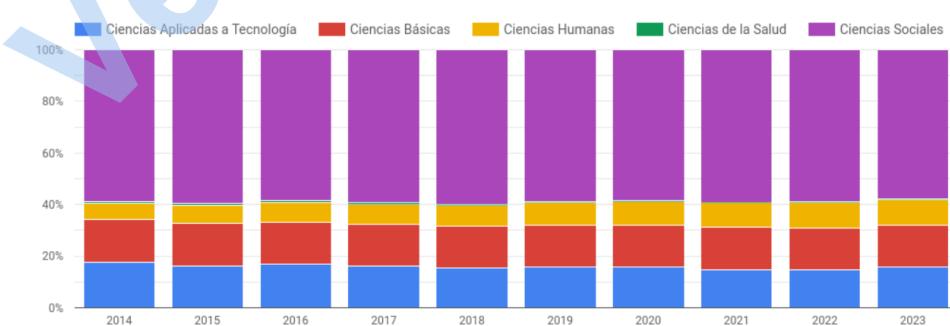
Los avances de este proyecto se pueden visualizar en [este enlace](#). El primer resultado de este proyecto es la construcción de una herramienta de visualización accesible a la comunidad educativa en general, sin requisitos de conocimiento previos en estadísticas educativas o procesamiento de bases de datos públicas, etc. Actualmente, los datos que ofrece el estado se presentan en planillas de cálculos con muchísimas hojas en cada una y un sin número de columnas que lleva un tiempo procesar. En la herramienta desarrollada, la información se organiza en tablas y gráficos con datos particulares relativos a la brecha de saberes digitales y computacionales en una serie periódica de los últimos 10 años y se desgrana por provincia, valor absoluto y porcentaje.

Asimismo, se presenta en la misma pantalla un menú desplegable que permite filtrar datos por provincia, ámbito (público o privado), sector (rural o urbano), género (fem o masculino al no encontrar datos de estudiantes no binarios). De esta manera los usuarios podrán hacer tantas preguntas como combinaciones posibles de datos. La tabla 2 presenta una captura de pantalla de una de las páginas de la herramienta donde se visualiza la tabla, gráficos y los menús desplegables.

Tabla 2: Herramienta de procesamiento de datos relativos a brechas de saberes digitales y computacionales



Porcentaje de Egresados por año y orientación



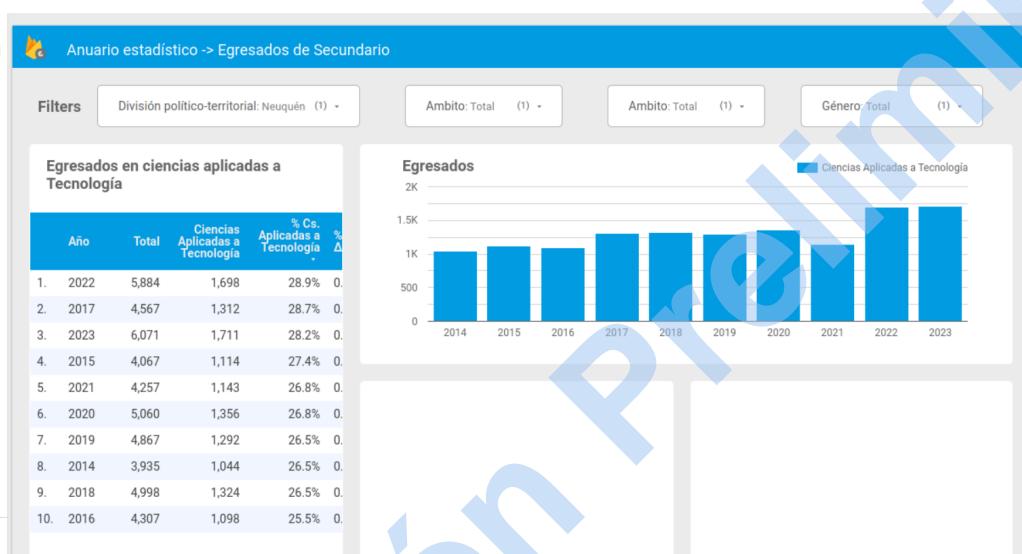
### b) Visibilización de los desafíos de acceso a saberes computacionales

El segundo resultado del proyecto es que la herramienta permite visibilizar relativamente el problema de acceso a saberes computacionales en el ámbito escolar y su evolución en los últimos 10 años. A manera de ejemplo, indagamos en la herramienta sobre la cantidad de egresados en carreras aplicadas a tecnología en los últimos 10 años en una cobertura nacional. La tabla 2 permite observar que a pesar de que la cantidad de egresados en el nivel secundario aumentó casi un 40% en los últimos 10 años, la cantidad de estudiantes egresados en Ciencias Aplicadas a Tecnología bajó en proporción aunque aumentó en números absolutos (debido a que hay más estudiantes dentro del sistema).

### c) Visibilización de las diferencias entre provincias

Se ha documentado extensamente la fragmentación de nuestro sistema educativo debido a las diferencias interprovinciales. No obstante ello, es posible identificar algunos movimientos en cuanto al acceso a saberes relativos a la tecnología digital en provincias que han realizado cambios curriculares tendientes a incluir estos saberes en la escolaridad básica. Por ejemplo, cuando vemos el caso de Neuquén, observamos que -mientras a nivel nacional el porcentaje de estudiantes egresados en este campo baja un par de puntos- a nivel provincial sube casi 4 puntos (Tabla 3). Neuquén ha sido la primera provincia en incluir saberes de Ciencias de la Computación de manera obligatoria para el ciclo básico del secundario en el año 2019.

Tabla 3: Egresados en áreas de Cs Aplicadas a Tecnologías en Neuquén.



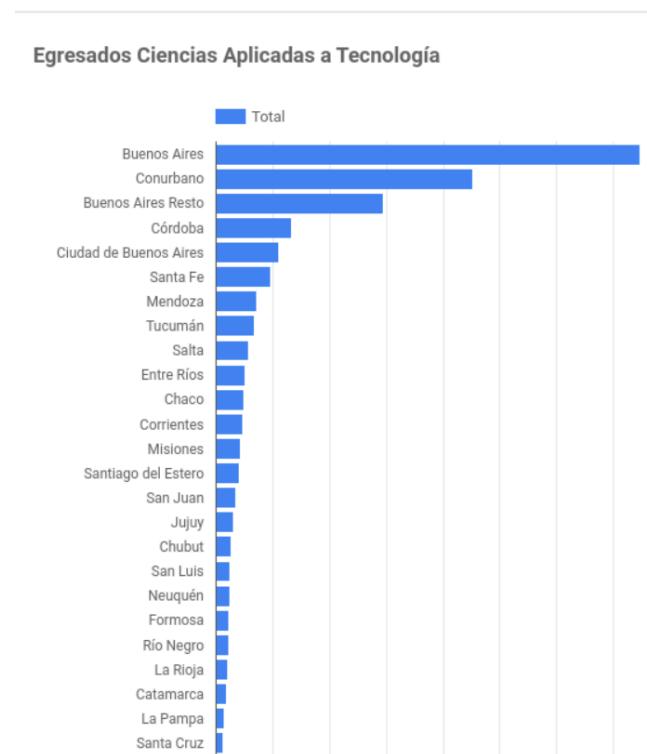
Otro modo de visualizar la información permite comparar las diferencias entre las provincias. En la tabla 4 observamos que los porcentajes de estudiantes que acceden a estas orientaciones presentan muchas variaciones. Esta información se puede filtrar por año para reconstruir los movimientos de la última década. Los datos se convierten en aproximaciones iniciales que generan nuevas preguntas de investigación relativas a las condiciones educativas de las provincias que reportan mayores porcentajes de acceso. De este modo, la información estadística es un buen recurso para promover investigaciones en estos temas.

Asimismo, cuando filtramos por ámbito (Tabla 5), observamos que en algunas provincias como Jujuy, Rio Negro o Misiones, concentran la gran mayoría de egresados en orientaciones tecnológicas en el ámbito privado. Para Jujuy, el total de egresados en tecnologías es de 24,9% pero casi 20% de ellos provienen del sector privado. En caso de Misiones es similar, del 28% total de egresados, el 19% proviene de escuelas privadas. En Rio Negro, del 25% de egresados, 19,4% son de escuelas privadas. Estos datos permiten visibilizar brechas de acceso según la posición social del estudiante en el ámbito escolar.

Respecto a la paridad de género, los datos generales no muestran allí una brecha profunda (Tabla 6) . En general se observa que cerca de la mitad de la matrícula son mujeres. No obstante, datos del nivel superior muestran otra realidad. Esta información permite hacernos preguntas sobre los momentos de la trayectoria escolar en que se profundizan las brechas de género.

La herramienta permite explorar y acercarse a datos generales para observar diferencias de acceso y hacernos preguntas en torno a ello.

Tabla 4. Egresados en Cs aplicadas a la tecnología por provincia en 2023.



División político-territorial	Total	Ciencias Aplicadas a Tecnología	% Cs Aplicadas a Tecnología
1. Total País	444,035	69,999	15.8%
2. Buenos Aires	186,914	17,291	9.3%
3. Conurbano	113,155	8,469	7.5%
4. Buenos Aires Resto	73,759	8,822	12.0%
5. Córdoba	33,459	8,151	24.4%
6. Ciudad de Buenos Aires	27,647	3,941	14.3%
7. Santa Fe	24,268	4,109	16.9%
8. Mendoza	17,905	4,091	22.8%
9. Tucumán	16,768	3,817	22.8%
10. Salta	14,473	3,739	25.8%
11. Entre Ríos	12,824	1,701	13.3%
12. Chaco	12,205	1,973	16.2%
13. Corrientes	11,576	2,138	18.5%
14. Misiones	10,856	3,136	28.9%
15. Santiago del Estero	10,184	2,718	26.7%
16. San Juan	8,543	1,380	16.2%
17. Jujuy	7,866	1,959	24.9%
18. Chubut	6,557	1,033	15.8%
19. San Luis	6,377	1,487	23.3%

Tabla 5. Egresados por provincia y por ámbito privado Tabla 6: Egresadas por provincia y por género femenino

División político-territorial	Total	Ciencias Aplicadas a Tecnología	% Cs Aplicadas a Tecnología
1. Total País	161,913	11,482	7.1%
2. Buenos Aires	70,837	2,257	3.2%
3. Conurbano	46,722	1,380	3.0%
4. Buenos Aires Resto	24,115	877	3.6%
5. Ciudad de Buenos Aires	16,995	1,918	11.3%
6. Córdoba	16,375	2,104	12.8%
7. Santa Fe	11,440	887	7.8%
8. Tucumán	5,897	589	10.0%
9. Mendoza	5,240	310	5.9%
10. Entre Ríos	4,294	106	2.5%
11. Salta	3,844	391	10.2%
12. Misiones	3,780	717	19.0%
13. Santiago del Estero	3,121	383	12.3%
14. Chaco	3,083	338	11.0%
15. Corrientes	2,515	184	7.3%
16. San Juan	2,340	125	5.3%
17. Río Negro	1,987	385	19.4%
18. Jujuy	1,780	338	19.0%
19. Neuquén	1,209	114	9.4%

División político-territorial	Total	Ciencias Aplicadas a Tecnología	% Cs Aplicadas a Tecnología
1. Total País	243,169	26,683	11.0%
2. Buenos Aires	100,222	4,900	4.9%
3. Conurbano	60,733	2,135	3.5%
4. Buenos Aires Resto	39,489	2,765	7.0%
5. Córdoba	18,502	3,453	18.7%
6. Ciudad de Buenos Aires	14,695	1,149	7.8%
7. Santa Fe	13,997	1,439	10.3%
8. Mendoza	10,059	1,648	16.4%
9. Tucumán	9,320	1,716	18.4%
10. Salta	7,996	1,650	20.6%
11. Entre Ríos	7,305	543	7.4%
12. Chaco	6,940	875	12.6%
13. Corrientes	6,440	827	12.8%
14. Misiones	6,180	1,482	24.0%
15. Santiago del Estero	5,916	1,468	24.8%
16. San Juan	4,746	491	10.3%
17. Jujuy	4,274	780	18.2%
18. Chubut	3,537	386	10.9%

d) Escasez de información específica

Como hemos venido mostrando, la información disponible en estas bases de datos no aborda específicamente el acceso y rendimiento en saberes computacionales, sino que son aproximaciones sobre cantidades de estudiantes que eligen orientaciones relacionadas con la tecnología. Actualmente estamos revisando las bases de Aprender donde se encuentran ítems relativos a prácticas de programación y robótica que nos pueden ofrecer nuevas perspectivas.

Asimismo, es necesario comprender además la capacidad del sistema para ofrecer estos saberes. No hemos conseguido datos sistematizados a nivel nacional sobre los cargos docentes y sus títulos de base que pudieran ofrecer acercamientos a la situación de la formación docente.

## 5. Conclusiones

Estamos dando los primeros pasos en la construcción de datos estadísticos que puedan aportar a los tomadores de decisiones información frecuente relativa a indicadores de acceso a saberes computacionales. Sabemos que nos queda mucho más trabajo para realizar para conseguir indicadores que puedan dar cuenta con mayor precisión de la alfabetización computacional. Estos indicadores muestran la cobertura y, en ese sentido, la capacidad relativa que tiene el sistema de ofrecer saberes relacionados con las tecnologías a los jóvenes. Pero, nada nos dicen de las mediciones de alfabetización digital.

No obstante estas primeras pruebas, nuestra experiencia en trabajo con funcionarios de los ministerios de educación de diferentes provincias nos ha enseñado la importancia de mostrar datos de acceso del estudiantado a estos saberes. Frecuentemente se desarrollan programas de alfabetización digital y a partir de ello se genera un supuesto de que el Estado ha garantizado el acceso. A partir de mostrar la información disponible es posible visibilizar que el estudiantado que participa de estas ofertas formativas es bajo en relación al resto de la población estudiantil. Esta información permite desterrar una falsa premisa de acceso a saberes computacionales.

Asimismo, esta herramienta permitirá ser sostenible a lo largo del tiempo puesto que se basa en datos públicos que toma el Estado Nacional desde 1996 y está regido por la ley de Educación Nacional. Esta situación es relevante por dos motivos. En primer lugar permite recuperar datos históricos para comparar cómo ha evolucionado la cantidad de estudiantes que reciben estos saberes a lo largo del tiempo. Esto es importante porque las provincias podrán estimar el impacto en el acceso a saberes de computación que pueden haber tenido algunos programas educativos. Por caso, en la Provincia de Córdoba además de la orientación en Informática se crearon Escuelas con orientación en Desarrollo de Software. Conocer el movimiento de la matrícula en estas escuelas permite analizar una variable de calidad de estos programas que es la cobertura.

En segundo lugar, es relevante porque permite aproximarse a la situación del acceso a saberes computacionales sin la necesidad de generar una herramienta de recolección de datos, administrar y analizarla. Esto suele ser muy costoso.

Más que conclusiones compartimos nuevos pasos a seguir. En primer lugar, poder analizar los indicadores de las otras bases de datos, integrar las bases y además construir indicadores de capacidad del sistema. En segundo lugar, generar plataformas de visualización de datos para funcionarios sin experiencia en manejo de datos estadísticos. Finalmente poder producir informes de investigación que puedan generar un estado de situación más claro respecto del acceso a saberes computacionales.

## Referencias

- [1] Programa de las Naciones Unidas para el Desarrollo, 2020. Argentina. Inclusión digital y desarrollo Humano. Documento de trabajo N2. Disponible en [Link](#)
- [2] Terigi, F. (2020). Escolarización y pandemia: alteraciones, continuidades, desigualdades. *REVCOM*. M.
- [3] Bell, T., Tymann, P., & Yehudai, A. (2011). The Big Ideas of K-12 Computer Science Education. *Bulletin of EATCS*, 124.
- [4] Harari, Y. N. (2020). The world after coronavirus. *Financial Times*, 20(03), 2020
- [5] UFECI, Unidad Fiscal Especializada en Ciberdelincuencia en Argentina (2020) Informe Anual correspondiente al 2020.
- [6] Scasso, Marino, Colombini y Bortolotto (2019) Evaluación de la Iniciativa Program.ar. Fundación Sadosky.
- [7] Fraillon, J., Ainley, J., Schulz, W., Friedman, T., & Duckworth, D. (2020). Preparing for life in a digital world. *IEA International computer and information literacy study 2018 international report* (p. 297). Springer Nature.
- [8] DiSessa, A. A. (2000). *Changing minds: Computers, learning, and literacy*. Mit Press.
- [9] Jacob, S. R., & Warschauer, M. (2018). Computational thinking and literacy. *Journal of Computer Science Integration*, 1(1).
- [10] 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).
- [11] 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.
- [12] Margolis, J., Estrella, R., Goode, J., Holme, J. J., & Nao, K. (2017). *Stuck in the shallow end: Education, race, and computing*. MIT press.
- [13] Echeveste, M. E., & Martínez, M. C. (2022) El rol de los Capitales Digitales en Escuelas Técnicas de Programación y las luchas estudiantiles por su lugar en el campo. *Revista de Sociología de la Educación RASE*, 14(3):307–324\
- [14] Tuffley, D. (2014). The gender digital divide in developing countries. *Future Internet*, 6(4), 673-687. \
- [15] Gray, T. J., Gainous, J., & Wagner, K. M. (2017). Gender and the digital divide in Latin America. *Social Science Quarterly*, 98(1), 326-340.
- [16] Acilar, A., & Sæbø, Ø. (2021). Towards understanding the gender digital divide: A systematic literature review. *Global Knowledge, Memory and Communication*.
- [17] Mariscal, J., Mayne, G., Aneja, U., & Sorgner, A. (2019). Bridging the gender digital gap. *Economics*, 13(1).
- [18] Jacob, S. R., & Warschauer, M. (2018). Computational thinking and literacy. *Journal of Computer Science Integration*, 1(1).
- [19] Suarez, D., & Yoguel, G. (2020). Latin American development and the role of technology: an introduction. *Economics of Innovation and New Technology*, 29(7), 661-669.
- [20] Trucco, D (2013) The digital divide in the Latin American context. In Ragnedda, M., & Muschert, G. W. (2013). *The digital divide*. Florence, KY: Routledge.
- [21] Margolis, J., Ryoo, J. J., Sandoval, C. D., Lee, C., Goode, J., & Chapman, G. (2012). Beyond access: Broadening participation in high school computer science. *ACM Inroads*, 3(4):72–78
- [22] Relkin, E. V. (2022). *The Development of Computational Thinking Skills in Young Children* (Doctoral dissertation, Tufts University).
- [23] Schleicher, A. (2019). PISA 2018: Insights and interpretations. *oecd Publishing*.
- [24] Gök, A., & Karamete, A. (2023). A validity and reliability study of the Turkish computational thinking scale. *Journal of Educational Technology and Online Learning*, 6(2), 421-437.
- [25] Snow, E., Rutstein, D., Bienkowski, M., & Xu, Y. (2017, August). Principled assessment of student learning in high school computer science. In *Proceedings of the 2017 ACM conference on international computing education research* (pp. 209-216).
- [26] Yadav, A., Burkhart, D., Moix, D., Snow, E., Bandaru, P., & Clayborn, L. (2015). Sowing the seeds: A landscape study on assessment in secondary computer science education. *Comp. Sci. Teachers Assn.*, NY,

NY.

[27] Bers, M. U., Govind, M., & Relkin, E. (2022). Coding as another language: Computational thinking, robotics and literacy in first and second grade. In *Computational thinking in prek-5: empirical evidence for integration and future directions* (pp. 30-38).

[28] Angulo, A. A. (2022). Los observatorios y la inteligencia colectiva como generadores de conocimiento. *Libro resultado de investigación*, 128.

[29] Guichard, E., Henríquez, G., & Barriga, O. (2008). Medición y tratamiento de variables. In *I Encuentro Latinoamericano de Metodología de las Ciencias Sociales (La Plata, 10 al 12 de diciembre de 2008)*.

## Anexo 1

### Antecedentes de tests de alfabetización computacional

Antecedentes
Pruebas PISA- Módulo Pensamiento Computacional
<a href="#">Normative Analysis of the TechCheck Computational Thinking Assessment. Relkin, E., Johnson, S. K., &amp; Bers, M. U. (2023).</a>
<a href="#">A validity and reliability study of the Turkish computational thinking scale (2023) Gök, A., &amp; Karamete, A. (2023)</a>
<a href="#">International Computer and Information Literacy Study. Technical Report of ICILS Fraillon, J., Ainley, J., Schulz, W., Friedman, T., &amp; Duckworth, D. (2020).</a>
<a href="#">Principled Assessment of Student Learning in High School Computer Science Snow, E., Rutstein, D., Bienkowski, M., &amp; Xu, Y. (2017, August).</a>
<a href="#">Sowing the Seeds: A Landscape Study on Assessment in Secondary Computer Science Education</a>
<a href="#">Assessment in Computer Science courses: A Literature Review Kallia, M. (2017).</a>
<a href="#">Designing an Assessment for Introductory Programming Concepts in Middle School Computer Science Yadav, A., Burkhardt, D., Moix, D., Snow, E., Bandaru, P., &amp; Clayborn, L. (2015).</a>
<a href="#">New Horizons in the Assessment of Computer Science at School and Beyond: Leveraging on the ViVA Platform Giordano, D., Maiorana, F., Csizmadia, A. P., Marsden, S., Riedesel, C., Mishra, S., &amp; Vinikienė, L. (2015).</a>
<a href="#">Coding as Another Language: Computational Thinking, Robotics and Literacy in First and Second Grade Bers, M. U., Govind, M., &amp; Relkin, E. (2022)</a>
<a href="#">Relationships between computational thinking and the quality of computer programs. Boom, K. D., Bower, M., Siemon, J., &amp; Arguel, A. (2022).</a>
<a href="#">A scoping review of computational thinking assessments in higher education Lu, C., Macdonald, R., Odell, B., Kokhan, V., Demmans Epp, C., &amp; Cutumisu, M. (2022).</a>
<a href="#">Extending Computational Thinking into Information and Communication Technology Literacy Measurement: Gender and Grade Issues Kim, H. S., Kim, S., Na, W., &amp; Lee, W. J. (2021).</a>
<a href="#">Computational Thinking Test: design &amp; general psychometry Román-González, M., Pérez-González, J. C., &amp; Jiménez-Fernández, C. (2015, October).</a>

# Reflexiones sobre la naturaleza de la Computación

Gonzalo Pablo Fernández<sup>1</sup>, Pablo E. “Fidel” Martínez López<sup>123</sup>,  
Alejandro Artopoulos<sup>4</sup> y Alejandra Lliteras<sup>5</sup>

<sup>1</sup> Departamento de Ciencia y Tecnología - Universidad Nacional de Quilmes

<sup>2</sup> Universidad Nacional de Hurlingham (UNAHUR), Centro de Investigación y Desarrollo en Informática Aplicada (CIDIA). Hurlingham, Argentina.

<sup>3</sup> Fundación Sadosky

<sup>4</sup> Universidad de San Andrés

<sup>5</sup> LIFIA, UNLP y CIC

gonzalo.pablo.fernandez@unq.edu.ar, fidel@unq.edu.ar,  
alepoulos@udesa.edu.ar, alliteras@udesa.edu.ar

## Resumen

En este trabajo buscamos clarificar ciertas nociones vinculadas a la naturaleza de Computación con el objetivo de entender cuáles son los elementos que definen la idea de cómputo y las características fundamentales de los mismos, así como las implicaciones de esto al enseñar Computación. Comenzamos con un breve resumen histórico sobre el surgimiento de la Computación, y cómo esta idea es previa a la idea de computadoras. Luego proponemos definiciones de computación y cómputo y las analizamos, para seguir después con la noción central de modelos de cómputo y de autómatas de propósitos generales. Concluimos resumiendo los conceptos trabajados y justificando la importancia de tener una visión clara de estos temas para poder enseñar Computación.

A través de estas reflexiones buscamos ayudar a docentes de Computación e Informática a clarificar varias cuestiones que suelen acarrear dificultades en la enseñanza y de esta manera mejorar su práctica docente, y también permitir a profesionales de la disciplina y a un público general alcanzar una comprensión más amplia sobre la programación y las computadoras.

**Palabras clave:** Computación, Cómputo, Modelos de cómputo

## 1. Introducción

Es común que las personas al hablar de Computación piensen inmediatamente en computadoras. No suena descabellado que alguien crea que las “Ciencias de la Computación” sean la disciplina que se encarga del estudio de aquellos equipos electrónicos a los que se conoce actualmente como computadoras. Quienes trabajamos en la disciplina y la estudiamos sabemos que esto no es así, que la computadora es sólo una herramienta y por eso solemos responder con una vieja frase erróneamente atribuida a Edsger Dijkstra: “Las Ciencias de la Computación no tratan sobre computadoras más de lo que la Astronomía trata sobre telescopios”<sup>1</sup>. Una frase similar que sí fue dicha por Dijkstra es

“El tema se hizo conocido prematuramente ... como ‘Ciencia de las Computadoras’ [*Computer Science*], lo que, en realidad, es como referirse a la cirugía como ‘Ciencia de los Bisturíes’ [*Knife Science*]; y quedó firmemente implantado en la mente de la gente que la Ciencia de las Computadoras trata sobre máquinas y sus equipos periféricos.”<sup>2</sup> [1, párr.8].

La analogía parece ser bastante clara; el objeto de estudio de la Astronomía no son los telescopios sino los astros (por eso se llama Astronomía y no Telescopiología o Ciencias de los Telescopios) y el de la Medicina no son los bisturíes o las aspirinas sino la salud humana. Los telescopios y los bisturíes son simplemente herramientas que se usan para facilitar la tarea de los astrónomos o cirujanos, respectivamente. Ni siquiera son indispensables ya que se pueden estudiar los astros sin un telescopio y se puede hacer investigación médica sin bisturíes.

Lo que estamos diciendo sobre la Astronomía o la Cirugía parecería ser obvio, incluso para una persona que no sea experta en esas disciplinas. Sin embargo, en el caso de la Computación, esto parecería no ser tan inmediato de vislumbrar. Es difícil imaginar a alguien que haga investigación en Ciencias de la Computación sin usar una computadora. La frase de Dijkstra enfatiza el problema de que en inglés, a las Ciencias de la Computación se les suele llamar “*Computer Science*” cuya traducción literal al castellano es “*Ciencias de las Computadoras*”<sup>3</sup>. En castellano la palabra no es “computadoras” sino “computación” y aún así, la confusión persiste, quizás porque el término “computación” tiene la misma raíz etimológica que la palabra “computadora”.

Ahora, si las computadoras son (como los telescopios o los bisturíes) solamente herramientas que utilizan quienes investigan en Ciencias de la Computación, entonces ¿cuál es el objeto de estudio de la disciplina? Si la Astronomía toma

<sup>1</sup> Consultar [https://en.wikipedia.org/wiki/Computer\\_science#Etymology\\_and\\_scope](https://en.wikipedia.org/wiki/Computer_science#Etymology_and_scope) y también Wikiquote [https://en.wikiquote.org/wiki/Computer\\_science#Disputed](https://en.wikiquote.org/wiki/Computer_science#Disputed)

<sup>2</sup> “the topic became ... prematurely known as ‘computer science’ – which, actually, is like referring to surgery as ‘knife science’ – and it was firmly implanted in people’s minds that computing science is about machines and their peripheral equipment.”

<sup>3</sup> En inglés también se utiliza el término “*Computing Science*”, pero su difusión es muchísimo menor.

su nombre a partir de su objeto de estudio (los astros), ¿de dónde toman el nombre las Ciencias de la Computación? Así como la Astronomía pudo desarrollarse sin telescopios, ¿puede la Computación desarrollarse sin computadoras? Más aún, así como la Astronomía existía desde antes de que se inventaran los telescopios, ¿existían las Ciencias de la Computación antes de la invención de las computadoras? ¿La Computación se creó para estudiar las computadoras o las computadoras se inventaron para llevar a cabo las ideas que pensaban quienes estudiaban Computación?

Estas preguntas motivan la redacción de este artículo, en el que nos proponemos reflexionar sobre la naturaleza de la Computación, para entender cuáles son los elementos que definen la idea de cómputo y las características fundamentales de los mismos, así como las implicaciones de esto al enseñar Computación. Comenzamos con un breve resumen histórico sobre el surgimiento de la Computación, y cómo esta idea es previa a la idea de computadoras. Luego proponemos definiciones de computación y cómputo y las analizamos, para seguir después con la noción central de modelos de cómputo y de autómatas de propósitos generales. Concluimos resumiendo los conceptos trabajados y justificando la importancia de tener una visión clara de estos temas para poder enseñar Computación.

## 2. Sobre el origen de la Computación

Si intentamos rastrear el origen de la Computación como disciplina probablemente lleguemos a nombres como Alan Turing (siglo XX), Charles Babbage (siglo XIX) o Gottfried Leibniz (siglo XVII). Es cierto que por aquellas épocas no existía la Computación como ciencia y no había gente que se dedicara a estudiar Computación pero si seguimos retrocediendo en el tiempo podemos encontrar ideas similares a las que motivaron la formalización de la Computación como la conocemos hoy en día varios siglos antes (incluso milenios). La evidencia más antigua de ideas asociadas a la Computación se remontan a la Babilonia del 1800 A.C. Por aquella época se utilizaban sistemas de cálculo de las reservas de grano y de los impuestos a cobrar en las primeras ciudades-estado – algo así como los procedimientos de la primera AFIP<sup>4</sup> de la historia de la humanidad. En palabras de Denning y Tedre,

“Procedimientos del estilo de algoritmos fueron encontrados en las tablas de arcilla de la antigua Babilonia [2], y el término “algoritmo” proviene del matemático persa de año 800 D.C., Muhammad ibn Mūsā al-Khwārizmī cuyos procedimientos precedieron la noción moderna de algoritmo [3].”<sup>5</sup>. [4, pág. 366]

Esto nos muestra que ciertas nociones acerca del uso de “procedimientos generales para resolver problemas matemáticos” (o, como dicen Denning y Tedre, de crear “algoritmos”) están presentes desde el principio de la historia de la humanidad. Tales procedimientos surgen como respuesta a la búsqueda de formas mecánicas de realizar cálculos matemáticos sin que sea necesaria una comprensión profunda de matemáticas, idealmente de forma completamente automática.

¿A qué nos referimos exactamente con “procedimientos generales”, “formas mecánicas” o “realizar cálculos matemáticos sin una comprensión profunda de matemáticas”? Nos referimos a poder describir la forma en la que se puede solucionar un cierto problema matemático de manera mecánica, en el sentido de que cualquiera pueda realizarlo sin tener que pensar ideas y sin la necesidad de entender cuál es la justificación de por qué lo que está haciendo efectivamente resuelve el problema. Por ejemplo, la descomposición de un número en factores primos o el cálculo del máximo común divisor entre dos números son problemas matemáticos para los cuales se conocen procedimientos mecánicos que los resuelven. En general cuando se enseñan las técnicas para resolverlos se enseña también por qué tales técnicas funcionan pero sería posible enseñarlas sin explicar su justificación y una persona debería ser capaz de realizarlas incluso sin saber qué es un número primo o un divisor.

Sería válido preguntarse en este punto qué tienen que ver estos problemas matemáticos con la Computación y por qué decimos que están asociados al origen de la Computación. La respuesta la podemos encontrar en los eventos que iniciaron el desarrollo moderno de la Computación como disciplina en el siglo XX (refiriéndonos ahora sí a Turing y compañía). La motivación en ese entonces era la misma: la búsqueda de la automatización de cálculos matemáticos (automatización no en el sentido de que lo ejecute una máquina sino en el de que pueda realizarlos cualquier persona sin tener que pensar ni entender exactamente qué está haciendo o por qué funciona). De hecho, las personas involucradas eran todas de las disciplinas de la lógica y las matemáticas (obviamente no existía gente que se dedicara a la Computación porque todavía no existía la Computación como tal).

El punto de partida fue una publicación del matemático David Hilbert a fines del siglo XIX, en la que presentó una serie de 23 problemas matemáticos no resueltos [5]. Uno de esos problemas era el de hallar un “proceso mecánico para resolver ecuaciones diofánticas”. Nuevamente, la motivación era resolver problemas matemáticos de forma mecánica. Con el objetivo de responder a la inquietud de Hilbert, otros lógicos y matemáticos de principios del siglo XX, como Haskell B. Curry, Alonzo Church y Alan Turing, entre otros, estudiaron maneras de formalizar la automatización de soluciones a problemas matemáticos. En otras palabras, se dedicaban a describir, con un lenguaje preciso y formal, los procesos de los que venímos hablando para resolver problemas matemáticos. Estos lenguajes no eran otra cosa que **lenguajes de programación**, ya que permitían, al igual que los lenguajes de programación actuales, describir comportamientos para resolver problemas. La única diferencia es que esos lenguajes de programación no se ejecutaban en computadoras porque, claro, las computadoras todavía no existían.

<sup>4</sup>En Argentina la AFIP, Administración Federal de Ingresos Públicos, es la agencia estatal de recaudación de impuestos, agencia similar a la IRS (Internal Revenue Service) de EEUU.

<sup>5</sup>“Algorithm-like procedures have been found on ancient Babylonian clay tablets (Knuth, 1972), and the term “algorithm” comes from the 800 CE Persian mathematician Muhammad ibn Mūsā al-Khwārizmī whose procedures preceded the modern notion of the algorithm (Knuth, 1981)”

Todos estos estudios y desarrollos dieron origen a lo que en su momento se conoció como Teoría de la Computación (que hoy en día se suele conocer simplemente como Computabilidad). Esta disciplina, precursora de las actuales Ciencias de la Computación, tenía como objetivo precisamente estudiar qué cálculos matemáticos se podían automatizar mediante procedimientos mecánicos y cuáles no. La forma de confirmar que un cálculo matemático podía ser automatizado (lo que hoy diríamos que es “computable”) era mostrar una descripción formal de un mecanismo que lo resolviera (en otras palabras, escribir un programa en un lenguaje de programación de tal forma que el resultado de su ejecución fuese el resultado esperable del cálculo en cuestión).

Esto ya nos permite empezar a responder, al menos en parte, algunas cuestiones referentes a las preguntas planteadas al inicio de este artículo. Aunque no lo hemos definido formalmente aún, ya podemos identificar el objeto de estudio de la Computación. Nos referimos a los procedimientos para solucionar problemas matemáticos. Lo que la disciplina estudia es cuáles de esos problemas pueden resolverse de forma automática, y cómo expresar esas soluciones. También nos permite entender el surgimiento de las computadoras como *la consecuencia* de la Computación, y no, como podría pensarse, que la Computación surgió luego de las computadoras, para estudiarlas. De hecho, a Turing se lo suele asociar al origen de las computadoras no porque tuviera como objetivo inventarlas sino porque su trabajo inspiró a diversos grupos a proponer las primeras computadoras electrónicas<sup>6</sup>. En otras palabras, mientras los matemáticos inventaban mecanismos formales para describir maneras de resolver ciertos cálculos, otras personas buscaban construir máquinas físicas que efectivamente realizaran estos cálculos de forma mecánica, siguiendo esas indicaciones. Puesto que la Computación busca determinar si un problema matemático se puede resolver de forma mecánica y cómo expresarlo, entonces es posible ver a una computadora nada más que como la máquina que realiza ese proceso mecánico para resolver el problema. Es claro que esas primeras computadoras eran muy diferentes a las que existen hoy en día. Aunque en ese momento se las llamó “computadoras”, hoy podríamos decir que eran “máquinas que realizan cálculos”. Para evitar confusiones, usaremos el término “máquina” para referirnos a cualquier dispositivo que realice cálculos. Esto incluye a las computadoras *actuales*, a las computadoras de entonces y a otros aparatos electrónicos que también son capaces de realizar cálculos como calculadoras, electrodomésticos, robots, etc.

Luego del surgimiento de estas primeras máquinas capaces de realizar cómputos automáticamente, se potenció la investigación en Computación como una disciplina técnica, alejándose de sus raíces matemáticas. A su vez, la investigación en Computación permitió avances tecnológicos en el diseño de computadoras. Ambas disciplinas, el desarrollo de computadoras y el de la Computación se retroalimentaron mutuamente. Gracias a la colaboración entre ambas comenzó un proceso que nos llevó a las computadoras de la actualidad, a otros avances tecnológicos como internet, la realidad virtual o la inteligencia artificial, y a un enriquecimiento de la Teoría de la Computación, hasta llegar a lo que hoy llamamos Ciencias de la Computación.

### 3. Sobre computación y cómputos

Una de las preguntas disparadoras de este artículo hablaba justamente del nombre de la disciplina y su relación con su objeto de estudio. Ya dijimos que las computadoras no son el objeto de estudio de la Computación ni el concepto del cual toma su nombre la disciplina. La palabra “Computación” en realidad se deriva de la palabra “cómputo” [8], que es el término con el cual nos vamos a referir a estos problemas matemáticos a partir de ahora. Un punto destacable es que no se trata de problemas matemáticos “puros” con números, que solo tienen validez en la práctica de las matemáticas escolares o la investigación matemática universitaria. Se trata de problemas matemáticos con datos de aplicación a todas las actividades (y ciencias) cuantificables. Por eso desde hace 30 años en la medida que se fueron descubriendo nuevas fuentes de datos y mejores tecnologías computacionales emergieron nuevas subdisciplinas como la Bioinformática, la Sociología Computacional o la Lingüística Computacional, esta última fundamental para el desarrollo de Grandes Modelos de Lenguaje de la IA generativa [9].

La noción de **cómputo** va siempre asociada al procesamiento de la información. *Computar es procesar información de alguna manera, o sea transformar cierta información en otra diferente*. Un **cómputo** no es más que el cálculo de un dato a partir de otros datos. Descomponer un número en factores primos es un cómputo que transforma un número en otros números (aquellos que son todos primos y tales que al calcular el producto entre todos ellos se obtiene el número original). Calcular el máximo común divisor es un cómputo que transforma un par de números en otro número (aquel que es el más grande que divide a los primeros).

Un cómputo no necesariamente trabaja únicamente con números. Cuando hablamos de *datos* nos referimos a cualquier porción de información que puede ser representada, como números, palabras, imágenes, personas, etc. Un ejemplo básico de procesamiento de información es transformar dos cantidades en una cantidad que represente el total de ambas; a ese procesamiento particular usualmente le damos el nombre de *suma*, y se puede expresar de muchas maneras diferentes, siendo la forma más tradicional la utilización de números en notación decimal. Otros ejemplos de cómputos más avanzados serían:

<sup>6</sup>Si bien en este trabajo nos focalizamos en el desarrollo teórico, no es posible obviar los fundamentos sociotécnicos de la Computación, es decir la simetría entre teoría y desarrollo de instrumentos de experimentación. Hay que señalar que el desarrollo como disciplina está mutuamente apalancado con el desarrollo tecnológico de la computadora y el software. Es decir requiere de un intercambio entre ideas, experimentar sus aplicaciones y ponerlas a prueba en contextos sociomateriales concretos. En el proceso del desarrollo teórico de la idea de una computadora, Turing experimentó y puso en funciones la máquina Bombe durante la Segunda Guerra Mundial. La máquina Bombe solo servía para desencriptar mensajes de la máquina encriptadora alemana ENIGMA. El poder experimentado por los aliados a través de Bombe fue un proxy del poder computacional futuro. [6, 7]

- obtener el promedio de notas de cada estudiante, donde los datos a transformar son las notas de cada estudiante y el dato transformado es el promedio de ellas;
- realizar una consulta en un buscador de internet, siendo en este caso los datos a transformar las palabras que se escriben en el buscador (una cadena de texto) junto a las páginas accesibles en internet, y el dato transformado, la lista de páginas seleccionadas como el resultado de la búsqueda;
- buscar en un texto posibles errores de ortografía, de puntuación, de concordancia o repeticiones de conectores (por ejemplo: pero, y, aunque, también) para evitarlas y mejorar la redacción, siendo el dato a transformar el texto completo y el dato transformado la lista de errores encontrados;
- decidir el mejor camino posible en la ciudad o hacia un destino de vacaciones en un mapa en línea con GPS analizando los datos de tráfico y las velocidades máximas permitidas en calles, avenidas y autopistas, siendo toda esa información los datos a procesar y el camino elegido el dato resultante; y
- analizar el comportamiento alrededor de un avión en una simulación de un “túnel de viento”, donde la información a transformar incluye los datos conocidos sobre la estructura del avión y la mecánica del viento a su alrededor, y la información transformada incluye los datos sobre cuál sería el comportamiento del aire alrededor de ese avión.

Todos estos son ejemplos de posibles cómputos. Notemos que al hablar de cómputo no nos estamos refiriendo necesariamente a un cómputo realizado por una máquina. Los cómputos son sólo la transformación de información, independientemente de quién la realice. En la sección anterior hablamos sobre máquinas que realizan cálculos para referirnos, además de a las computadoras, a cualquier aparato que sea capaz de realizar cálculos (es decir, cómputos). Más allá de las computadoras en particular, vivimos rodeados de máquinas así. Los teléfonos inteligentes son computadoras de bolsillo. Muchos electrodomésticos como heladeras o televisores ahora realizan operaciones mucho más complejas que sólo aquello para lo que fueron inventados originalmente. Aparatos tan simples como los relojes o incluso hasta las lámparas ahora pueden conectarse a internet. Todas estas máquinas están realizando cómputos (procesando y transformando información) constantemente.

Aunque lo más común es que el procesamiento de información lo hagan esas máquinas, la mayoría de los cómputos pueden ser realizados también por personas. Está claro que la suma y el promedio son cómputos que una persona podría realizar. Lo que quizás no sea tan claro es que *todos* los ejemplos mencionados hasta ahora son cómputos realizables por personas. Es más, *cualquier* cómputo que pueda ser realizado por una máquina puede ser realizado también por una persona.

Esta afirmación puede sonar poco intuitiva al principio (en especial considerando los últimos avances en inteligencia artificial) pero al analizarla detenidamente, es razonable al menos en teoría. Dijimos que las computadoras surgieron como máquinas que realizaban cómputos de forma automática. También dijimos que estos cómputos no eran cómputos cualquiera sino sólo aquellos que podían realizarse de forma mecánica. Más aún, la forma mecánica de realizarlo debía ser definida por una persona. Entonces, todo aquello que las máquinas de antaño eran capaces de hacer (los cómputos que eran capaces de realizar) era definido por una persona y por lo tanto, esa persona también podía hacerlo (o al menos, sabía cómo hacerlo). Sería válido pensar que, después de tantos años, las computadoras avanzaron tanto que los cómputos que pueden realizar ahora son mucho más avanzados que aquellos que podían realizar las primeras máquinas. Sin embargo, lo cierto es que las máquinas de hoy en día son igual de poderosas (en términos de los cómputos que son capaces de realizar) que las de hace casi 100 años. Son mucho (muchísimo) más rápidas pero no más poderosas. Cualquier cómputo que pueda ser realizado por una máquina hoy en día, podría haber sido realizado por una máquina del siglo pasado (salvo por la cantidad de tiempo que le tomaría) y, por lo tanto, también puede ser realizado por una persona (nuevamente, suponiendo que dispone de suficiente tiempo para hacerlo<sup>7</sup>).

Además de la cuestión del tiempo, hay otra razón por la que estos cómputos son llevados a cabo por máquinas y no por personas: las personas pueden cometer errores mientras que las máquinas no (al menos mientras no haya fallas en el hardware). De hecho, cuando se habla de un error en una máquina suele ser un error cometido por una persona al programarla y no a un defecto de la máquina en sí, ya que después de todo, las máquinas solamente realizan operaciones según las indicaciones que las personas les dan.

Volvemos a remarcar de todas formas que la elección de hacer los cómputos mediante máquinas está dada por la practicidad y NO porque una persona no pueda hacerlo, y este es un resultado que nos interesa destacar: las máquinas no son más inteligentes que los humanos. Ni siquiera son inteligentes en lo absoluto. Todo lo contrario: lo único que saben hacer es seguir a rajatabla reglas predefinidas ya escritas. Y como ya se mencionó, esas reglas tuvieron que haber sido escritas por un humano, así que si una máquina es capaz de realizar un cómputo es porque antes una persona especificó la forma de hacerlo (recordemos que las máquinas sólo realizan cálculos que pueden ser automatizados pero son las personas las que definen los procedimientos que automatizan tales cálculos). En última instancia, la “inteligencia” pasa únicamente por descubrir o inventar cuál es la estrategia que permite describir de forma mecánica el procedimiento para resolver un problema en particular. Una vez que ese procedimiento está definido, no es necesario inteligencia para reproducirlo. Por

<sup>7</sup>El tiempo que puede requerir a una persona en hacer algunos de los cómputos mencionados antes, o los asociados al entrenamiento o uso de los sistemas basados en Inteligencia Artificial, excedería completamente su tiempo de vida. O sea, si bien podrían ser hechos por una persona (o quizás más bien podríamos decir por un equipo de personas, que pudieran irse reemplazando a lo largo del tiempo), no sería razonable esperar que así fuera, pues podría tomar millones de años un cómputo que una máquina moderna puede hacer en milisegundos...

eso también decimos que puede resolverlo incluso una persona que no tiene el conocimiento matemático para entender lo que está haciendo. En conclusión, no puede pasar que una máquina pueda hacer algo que una persona no sepa cómo hacer.

Curiosamente, sí existen problemas matemáticos que pueden (potencialmente<sup>8</sup>) ser resueltos por personas y no por máquinas: todos aquellos que no pueden ser automatizados. Volveremos sobre esto más adelante. Por ahora, sigamos profundizando en la definición de cómputo.

## 4. Modelos de cómputo como la base de la Computación

Dijimos que un cómputo es un procesamiento de información. Sin embargo, no cualquier procesamiento de información es un cómputo, pues para poder hablar de cómputo se requiere cierto grado de precisión y ausencia de ambigüedades. Por ejemplo, actividades cotidianas como seguir una receta de cocina o lavarse los dientes NO constituyen ejemplos de cómputos, por carecer de precisión en los diferentes elementos involucrados y sus operaciones realizables.

Es importante poder distinguir entre acciones automatizables y acciones que requieren intervención humana. Los cómputos que son completamente automatizables requieren una especificación no ambigua del mecanismo a través del cual se desarrollan las acciones. Cocinar es una actividad humana muy compleja que todavía no puede ser abordada por una máquina, ni siquiera aquellas que utilizan inteligencia artificial (al menos no en su totalidad<sup>9</sup>). En una receta de cocina, una instrucción puede ser “*Poner 2 huevos en un recipiente y mezclar hasta que quede una mezcla homogénea*”. Una persona entiende perfectamente que tiene que romper los huevos y no ponerlos con cáscara y sabe más o menos hasta qué punto tiene que mezclar. Esto es porque tiene la capacidad de razonar para poder comprender la definición completa de la tarea a realizar, ignorando los detalles que no estén del todo especificados o completándolos. Para una máquina sería imposible hacer lo mismo ya que en ningún lugar se aclara que “*Poner 2 huevos en un recipiente*” implica en realidad primero romperlos y solamente poner el contenido en el recipiente descartando la cáscara. Tampoco está especificado qué significa que la mezcla quede homogénea. Para que una máquina pueda interpretar correctamente este tipo de instrucciones hay que formalizar varios aspectos incluyendo el comportamiento esperado por la instrucción de poner huevos en el recipiente y algún mecanismo que permita categorizar el estado de la mezcla para determinar si ya es homogénea o todavía no. Es decir, las posibles operaciones y elementos que participarán del cómputo deben ser establecidos con precisión *a priori* del cómputo en sí, ya que la máquina no tiene la capacidad de hacerlo por su cuenta.

Existen diferentes formas de expresar el procesamiento que define un cómputo, y también de cuáles son los aspectos de ese proceso que se quieren analizar o detallar. Todas estas características respecto al procesamiento de información de un cómputo vienen dadas por un **modelo de cómputo** [10, 11]. Un **modelo de cómputo** es un mecanismo formal, teórico, preciso y no ambiguo, que define tres elementos: primero, de qué manera expresar la información, segundo, las operaciones básicas posibles para manipular dicha información, y por último, ciertas reglas de procesamiento mediante combinaciones de operaciones básicas y representaciones de la información. Cuando dijimos que los matemáticos de mitad del siglo XX desarrollaban lenguajes de programación en realidad hicimos un abuso del lenguaje. Lo que desarrollaban realmente eran modelos de cómputo. Es decir, formas de describir un procesamiento de información. Los lenguajes de programación son sólo especificaciones sintácticas para escribir programas de forma concreta y, cuya ejecución está dada por el modelo de cómputo subyacente. Si el cómputo es la transformación de información en sí, el modelo de cómputo es el sistema que describe los elementos y procesos que establecen cómo realizar esa transformación.

Para definir un modelo de cómputo se deben determinar los elementos básicos con los cuales se puede operar, las operaciones que se pueden realizar entre tales elementos y las reglas que indican qué efecto tiene cada operación. Un cómputo es, entonces, la descripción del proceso que realiza un determinado modelo de cómputo para convertir cierta información en otra, expresada en función de las operaciones y elementos del modelo. Por eso carece de sentido hablar de cómputo sin definir primero un modelo de cómputo que le dé entidad.

Para el ejemplo de la suma, un modelo de cómputo posible elige representar<sup>10</sup> los números en notación decimal, y define como operaciones básicas la suma de dígitos decimales y el tratamiento de diferentes dígitos de las secuencias de dígitos que representan ambos números. En dicho modelo es posible expresar la conocida operación de suma (así como también la de multiplicación y otras operaciones aritméticas). Pero podrían considerarse otros modelos, donde los números se representan con cantidades de porotos, y la operación de suma fuese simplemente juntar dos grupos de porotos.

Un modelo de cómputo para representar la transformación de una búsqueda en internet en una lista de posibles resultados requiere mucha más complejidad, representando la información de muchos más elementos y operaciones (tales como las palabras de un idioma y la estructura de ese idioma, los datos sobre los miles de millones de sitios de internet disponibles, estructuras de información adecuadas para que la búsqueda pueda ser hecha dentro de un tiempo razonable, y muchísimas otras características que se pueden incorporar a esa transformación, tales como preferencias basadas en la información del usuario que consulta, o su ubicación, y muchas más, además de todas las operaciones necesarias para

<sup>8</sup>Aquí nos referimos a que existe la posibilidad. Está claro que no cualquier persona, solo por el hecho de ser una persona, automáticamente va a ser capaz de hacer una demostración matemática. Pero si piensa lo suficiente y estudia lo suficiente, es posible que eventualmente lo consiga. Eso no va a pasar nunca con una computadora, sin importar de cuánto tiempo disponga (no al menos con el estado actual de la Computación).

<sup>9</sup>En la actualidad, lo más cercano que podemos encontrar a un “robot que cocina” no hace más que mezclar y calentar la comida automáticamente por tiempos predeterminados, pero requiere que una persona introduzca los ingredientes en el recipiente y seleccione la opción que define los tiempos y modo: <https://tienda.moulinex.com.ar/robot-de-cocina-click-chef/>

<sup>10</sup>Usamos el término “representar” en el sentido de “modelar”, o sea, describir la idea en término de elementos concretos. Por ejemplo, en el caso de números, la idea es la cantidad expresada y los elementos concretos son los dígitos decimales; decimos que la secuencia de dígitos “representa” al número, lo “modela”.

manipular todos esos elementos). Esto muestra que no todos los modelos de cómputo tienen las mismas características, así como tampoco todos los cómputos las tienen.

Notemos que al hablar de un modelo de cómputo no estamos hablando de computadoras. Ni siquiera de lo que venimos llamando “máquinas que realizan cómputos”. Tanto las computadoras como el resto de las máquinas son dispositivos computacionales físicos mientras que los modelos de cómputo son artefactos computacionales teóricos, abstracciones que describen el funcionamiento de una máquina teórica. Es decir, un modelo de cómputo no es algo concreto que efectiviza el procesamiento sino que alcanza con tener la descripción de cómo debe realizarse (podría ser una hoja de papel en la que estén escritos los elementos, las operaciones y qué debe hacer cada una de éstas). Si el modelo es una descripción abstracta, entonces una computadora se puede considerar como una posible implementación física de una de estas máquinas teóricas, o, variando la perspectiva, las máquinas teóricas se pueden considerar como la expresión de ciertas características de algunas máquinas físicas.

Históricamente, el término “computadora” no siempre se refirió a lo que hoy en día se conoce como computadora (o sea, computadoras personales o de escritorio). En el contexto de la Computación, el término “computadora” solía usarse, como su nombre lo indica, para hablar de cualquier entidad que realiza cómputos, lo que podríamos llamar una *entidad computacional*. Esta definición abarca, además de las computadoras personales y el resto de las ya mencionadas máquinas que realizan cómputos, a los ábacos y hasta equipos de personas<sup>11</sup>. Las entidades computacionales entonces pueden ser tanto máquinas como personas y hasta otros elementos. Para referirnos a lo que venimos llamando “máquinas que realizan cómputos” utilizaremos el término *dispositivos computacionales* ya que no estamos hablando de cualquier entidad con la capacidad de realizar cómputos sino específicamente aquellas que lo hacen de forma automática (dejando por fuera a los ábacos y las personas). Cada uno de estos dispositivos se distingue por el modelo de cómputo que implementa y cada modelo de cómputo tiene un propósito particular. Es claro que hay algo más detrás de este tema que debemos profundizar.

## 5. Sobre las máquinas y sus propósitos

Como dijimos en la sección anterior, un modelo de cómputo no es un dispositivo computacional sino una descripción del funcionamiento de uno de ellos. Un dispositivo computacional es la implementación física de esa descripción. Durante el último siglo se han dado numerosos ejemplos de la rica interacción entre los modelos de cómputo y sus implementaciones físicas: muchos ejemplos de modelos teóricos inspiraron construcciones físicas, y también la construcción de máquinas físicas para implementar cómputos inspiraron nuevos modelos teóricos y permitieron refinar los mismos. Denning y Tedre, por ejemplo, le dan mucha más importancia a las computadoras que a los modelos de cómputo que las representan [12]. Pero dado que nos interesa desarrollar habilidades humanas y procesos mentales, creemos que el foco debe estar puesto en los modelos, siendo las computadoras una de las manifestaciones posibles (y necesarias, aunque no únicas) de ellos.

Recordemos que originalmente se les llamaba “computadora” a cualquier entidad física (humana, mecánica o digital) que permitiera realizar alguna transformación de información. Es decir, se les llamaba “computadoras” tanto a las personas que realizaban los cómputos como a determinadas herramientas especialmente diseñadas para realizar cómputos específicos (como el ábaco). Más adelante surgieron máquinas (los dispositivos computacionales) que permitían realizar estos cálculos de forma automática (es decir, sin la supervisión de una persona). Pero estas primeras máquinas tampoco necesariamente eran como las máquinas a las que hoy en día llamamos “computadoras”. Cada una de ellas estaba diseñada para realizar un cómputo particular. Muchas de las llamadas “primeras computadoras” de la historia (como la Pascalina de Pascal en 1642, que sólo podía realizar sumas, o la calculadora mecánica de Leibniz en 1671 que podía hacer multiplicaciones y divisiones) eran así.

En 1833, Charles Babbage diseña la “máquina analítica” (aunque nunca llega a construirla). Esta “computadora” tenía la particularidad de que no realizaba un único cómputo sino que podía configurarse de distintas formas para que realizará distintos cómputos. Podría decirse que, antes de ella, los dispositivos computacionales eran “de propósito específico” ya que tenían un único propósito y sólo podían ser utilizadas para eso, mientras que la máquina analítica de Babbage era un dispositivo computacional “de propósito general”.

Aparece entonces una importante distinción entre clases de dispositivos computacionales.

Por un lado están los de *propósito específico*, que realizan una tarea particular y sólo pueden hacer eso. Así como las primeras computadoras, que realizaban operaciones particulares, muchos de los electrodomésticos que usamos cotidianamente quedan expresados por esta definición (por ejemplo, la calculadora para hacer cuentas, o el lavarropas automático que permite seleccionar distintos programas de lavado), ya que son dispositivos físicos que automatizan una tarea determinada.

Por otro lado están los dispositivos computacionales de *propósito general*, que se caracterizan por incluir algún mecanismo que permite configurar cuál va a ser su propósito. En ambos casos nos referimos a dispositivos que permiten automatizar un proceso. Sin embargo, los de propósito general tienen la particularidad de que permiten realizar *cualquier cómputo*<sup>12</sup>. En otras palabras, un dispositivo computacional de propósito general *es un tipo de dispositivo particular que permite simular a cualquier otro dispositivo computacional* (incluyendo a otros dispositivos de propósito general).

Esta simulación de un dispositivo por otro (o de un modelo de cómputo por otro) debe interpretarse en forma simbólica. Es claro que una computadora no puede lavar ropa porque no posee las piezas de hardware necesarias para hacerlo. Cuando

<sup>11</sup>De hecho, solía existir la profesión de “computadora” que era aquella persona que se dedicaba a realizar cómputos como se ve en la película Talentos Ocultos (*Hidden Figures*).

<sup>12</sup>En realidad nos referimos a cualquier cómputo que sea automatizable ya que, como veremos más adelante, existen cómputos que se sabe que ningún dispositivo computacional puede realizar.

decimos que un dispositivo puede simular a otro queremos decir que, en términos de procesamientos de información, puede realizar las mismas transformaciones. En el caso del lavarropas, las transformaciones de información no son transformar ropa sucia en ropa limpia sino interpretar las opciones de configuración seleccionadas a través de botones o perillas como datos que determinan, por ejemplo la cantidad de tiempo que debe durar el lavado, así como también activar y desactivar las piezas de hardware, como el motor o la válvula. Todo ese procesamiento puede ser llevado a cabo por una computadora. En el caso de la calculadora, como aparato es un dispositivo físico que implementa un modelo de cómputo para la aritmética y permite, entre otras cosas, realizar el procesamiento de información que ya mencionamos antes conocido como suma. Una computadora puede simular a una calculadora y así si tenemos una computadora, no es necesario tener también una calculadora, pues la primera puede simular a la segunda.

Es importante resaltar el hecho de que cuando decíamos que las computadoras antiguas eran tan poderosas como las actuales nos referíamos justamente a esto: más allá de los elementos de hardware con los que cuenten, a nivel de capacidad de cómputo, son todas igual de poderosas, como profundizaremos en la sección 6. Así, desde una perspectiva computacional un dispositivo se define únicamente por los cálculos que puede realizar. Sólo importan los datos de entrada y el dato de salida. Por eso se dice que las computadoras (y en general cualquier dispositivo computacional) lo único que hacen es realizar cálculos. Es cierto que también acceden a internet y muestran imágenes en la pantalla, entre otras cosas (así como el lavarropas lava la ropa o los robots activan sus motores y articulaciones) pero eso es porque los cálculos realizados tienen efectos en el mundo físico. Todas estas interacciones con el mundo externo son efectos colaterales del cálculo y no el cálculo en sí. La Computación (o más específicamente, la Computabilidad<sup>13</sup>) sólo estudia a los cálculos y no sus efectos en el mundo físico.

Volviendo a la idea de dispositivos físicos que automatizan tareas, podemos pensar como ejemplo elemental en una caja musical (que si bien no es un dispositivo computacional, ya que no procesa información, sirve como ejemplo para pensar en las implicaciones de que un dispositivo simule a otro). En lugar de tomar un instrumento y reproducir una a una las notas que queremos, le damos cuerda y luego la melodía se reproduce automáticamente. Sin embargo, una caja de música siempre reproduce la misma melodía. Podríamos decir que es un reproductor de música de propósito específico (en tanto está construida con el objetivo de tocar específicamente esa melodía y no otra). En cambio, un tocadiscos lo podemos considerar un reproductor de música de propósito general, ya que permite reproducir cualquier melodía basándose en el disco que se le coloque. Es decir, un tocadiscos puede simular cualquier caja de música si se le coloca el disco adecuado. De manera similar, las máquinas de juegos de fichas en los arcades de la década de 1980 son dispositivos de videojuegos de propósito específico. Cada máquina implementa un juego pero no se puede jugar a Donkey Kong en una máquina de Pacman. Un ejemplo de dispositivo de videojuegos de propósito general podría ser una consola de videojuegos, en tanto puede simular cualquiera de las máquinas de fichas originales si se le inserta el cartucho correcto.

Estos ejemplos sirven para ilustrar la idea de simulación y de que los dispositivos “de propósito general” pueden simular a los de propósito específico utilizando algún componente adicional que describa de alguna forma aquel otro modelo que se quiere imitar (el disco en el caso de la caja de música o el cartucho en el caso de las consolas de videojuegos).

Llevando esto al extremo, la computadora es un dispositivo computacional de propósito general que puede simular a los demás dispositivos computacionales, incluso a los de propósito general. Una computadora moderna implementa un modelo de cálculo que permite simular un tocadiscos (y por lo tanto, cualquier caja musical), una consola de videojuegos (y por lo tanto, cualquier máquina de arcade), y cualquier otro dispositivo que implemente un modelo de cálculo (suponiendo que tiene ciertos componentes adecuados, como un parlante, por ejemplo). Sin embargo, mencionamos que para que un dispositivo de propósito general pueda simular correctamente a otro hace falta un componente adicional que describa las propiedades de aquello que se quiere simular. Así como los tocadiscos necesitan un disco y las consolas de videojuegos un cartucho, los dispositivos computacionales requieren un **programa**<sup>14</sup> para permitirles simular a otros autómatas.

El programa que ejecuta una calculadora en nuestro sistema operativo favorito puede ser interpretado entonces como la descripción del modelo de cálculo de una calculadora expresado en términos del modelo de cálculo que implementa nuestra computadora personal. Esto se ve mucho más fácilmente en los teléfonos inteligentes, que permiten instalar aplicaciones que pueden realizar muchísimas tareas diferentes en un mismo dispositivo. Curiosamente, los celulares comenzaron siendo dispositivos de propósito específico, ya que tenían como única funcionalidad la de realizar llamadas. Con el tiempo se les fueron agregando funcionalidades, como juegos y reproductores de música pero seguían siendo de propósito específico porque solamente podían realizar las tareas que venían incorporadas. No era posible entonces tomar uno de estos dispositivos y usarlo para realizar una tarea arbitraria. Con el lanzamiento de los llamados “teléfonos inteligentes” esto cambió. Los celulares pasaron a ser tan “poderosos” (en términos de lo que les es posible computar) como las computadoras (y cualquier otro dispositivo computacional de propósito general).

Una última consideración a tener en cuenta al hablar de “computadoras” es que la mayoría de la gente no utiliza el término “computadora” para referirse al dispositivo computacional de propósito general sino que ese término refiere a un sistema compuesto por varios componentes de hardware (incluyendo pantalla, teclado, parlantes, conexión a internet, etc.) entre los cuales hay uno (el procesador) que es el verdadero dispositivo computacional de propósito general. Antes mencionamos que cada dispositivo computacional tiene un propósito. El propósito de la computadora (refiriéndonos ahora sí a aquellos dispositivos electrónicos que incluyen a las computadoras personales o las notebooks) es simular otros dispositivos computacionales (ya sean de propósito específico o de propósito general), así como el propósito de los modelos de cálculo

<sup>13</sup>Estamos conscientes de que las Ciencias de la Computación al día de hoy se han desarrollado considerablemente desde sus inicios y ahora existen varias ramas de la Computación, como la robótica, para las cuales sí es importante considerar el impacto de los cálculos en el mundo real. Cuando hablamos aquí sobre la Computación nos estamos refiriendo a sus raíces.

<sup>14</sup>Para más detalles sobre la noción de programa, ver la sección 8.

que ellas implementan es simular otros modelos de cómputo. Notemos la importancia de este resultado. Si un dispositivo particular puede realizar *cualquier* cómputo, entonces ese único dispositivo podría reemplazar a todos los demás. De hecho, es en parte lo que está sucediendo con los teléfonos inteligentes. Muchas aplicaciones reemplazan dispositivos de propósito específico. Es como tener una “navaja suiza” que reemplazara a *cualquier* herramienta, (o, como en el caso de la analogía con la caja de música y el tocadiscos, tener un tocadiscos que puede reproducir *cualquier* melodía en lugar de tener infinitas cajas de música).

## 6. Un modelo para expresarlos a todos: modelos de cómputo universales y sus límites

La misma consideración realizada sobre los dispositivos físicos puede realizarse sobre los modelos de cómputo. Así, podemos hablar de modelos de cómputo específicos (por ejemplo, el de las operaciones aritméticas o el que implementa el lavarropas) y de modelos de cómputo generales, que permiten simular cualquier otro modelo de cómputo. A estos modelos de cómputo generales se los denomina **modelos de cómputo universal** [13, 14], pues alcanza con uno de ellos para representar a cualquiera de los otros. Luego, podemos decir que las computadoras modernas (así como el resto de los dispositivos computacionales de propósito general) implementan modelos de cómputo universales.

El modelo de cómputo universal más conocido (pues fue el primero para el que se demostró tal propiedad) es la Máquina de Turing<sup>15</sup> [13, 14] (que a pesar de usar el término “máquina” no hace referencia a un dispositivo físico sino únicamente al modelo de cómo funcionaría). Una Máquina de Turing cuenta con una cinta de longitud infinita sobre la cual se pueden escribir símbolos (lo que podría verse como una hoja cuadriculada con un único renglón pero que siempre hay más cuadros tanto para la derecha como para la izquierda de forma que nunca llegaríamos al borde de la hoja) y un cabezal que se puede mover a lo largo de la cinta (como una máquina de escribir bidireccional). El cabezal puede observar el símbolo escrito en la posición de la cinta sobre la que se encuentra y puede reemplazarlo por otro. El procesamiento está dado por una secuencia de operaciones donde cada operación consiste en leer el símbolo de la posición de la cinta sobre la cual se encuentra el cabezal, escribir un símbolo (que podría ser el mismo) en dicha posición y moverse una posición hacia la derecha o hacia la izquierda, ambas cosas definidas a partir de un estado interno actual, que también puede modificarse como efecto de esta operación a través de lo que hoy se denomina “máquina de estados finita”. Los datos de entrada son los que están escritos sobre la cinta (usando alguna representación de los mismos a través de secuencias de símbolos) antes de iniciar el procesamiento y el dato de salida es el que queda escrito en la cinta (usando la misma representación) al finalizar el procesamiento.

Un ejemplo bien simple de cómo usar una máquina de Turing para computar es representar el cómputo de la función que toma un número natural y devuelve el número siguiente. En matemática, esa función se suele describir como  $f(x) = x + 1$ . Los números naturales se pueden representar usando una tira de símbolos cuya longitud coincide con el número que representan. Por ejemplo, el número 3 se representaría con una tira de 3 símbolos y el número 5 con una tira de 5 símbolos (es indistinto qué símbolo utilizar, podría ser una equis, un “palote” o cualquier otro). Así, a partir de cierta posición se podría colocar un símbolo en cada una de las siguientes celdas, tantas como el número a representar. El procesamiento de la máquina de Turing para computar esta función consiste simplemente en escribir un símbolo más a la derecha del último símbolo escrito.

Otro modelo de cómputo muy popular es el Cálculo Lambda [15, 16]. En este modelo los elementos son abstracciones de funciones (usualmente llamadas *abstracciones lambda*). Cada función tiene un *parámetro*, que nombra al dato que se transformará, y un *cuerpo*, que describe el resultado de evaluar esa función en el parámetro – o sea, la transformación a realizar. Por ejemplo, el elemento que representa a la función identidad ( $f(x) = x$ ) tiene como cuerpo a su parámetro, ya que al ser evaluada el resultado debe ser el mismo que el recibido como argumento – esto se nota  $\lambda x . x$ . La única operación disponible es la aplicación de una función a un argumento. Como sólo existen funciones, cualquier elemento del dominio del problema debe representarse únicamente usando funciones (hasta los números). El procesamiento se resuelve haciendo reducciones de una expresión inicial (la que expresa el cómputo deseado en forma de una función aplicada a sus argumentos) hasta llegar a una expresión que no puede reducirse más. La reducción consiste en tomar una parte de la expresión que incluya una función aplicada a un argumento y reemplazarla por el resultado de evaluar dicha función a dicho argumento.

Un modelo similar, aunque mucho menos popular, es el de la Lógica Combinatoria [17, 18]. Como en el Cálculo Lambda, los únicos elementos que existen son funciones y la única operación que existe es la aplicación. La diferencia radica en el mecanismo para describir a las funciones. Mientras que en el Cálculo Lambda las funciones se describen a partir de su parámetro y su cuerpo, en la Lógica Combinatoria se describen a través de combinaciones sucesivas de otras funciones, a partir de 2 funciones primitivas, llamadas S y K, para las que se explica su comportamiento al ser aplicadas.

En el modelo de cómputo de Funciones Recursivas Generales [19] también se describen funciones a través de combinaciones de otras funciones, pero en este caso a partir de un conjunto de funciones primitivas como la proyección, la composición y la minimización no acotada.

<sup>15</sup>Turing llamó a sus máquinas “Máquinas de Computar” y a la que era universal la llamó “Máquina Universal” (*Universal machine*) o “Máquina de Cómputo Universal” (*Universal Computing Machine*). Más adelante se las conoció simplemente como “Máquina de Turing”, ya que ese fue el nombre con el que las nombraba Church.

Es notable como todos estos modelos surgen más o menos en la misma época y con características muy similares a pesar de que fueron desarrollados de forma independiente. También es notable que a pesar de ser tan diferentes, cualquiera de ellos puede representar a los otros, codificando adecuadamente los elementos del dominio representado con el de base y modelando su ejecución usando la ejecución del dominio de base.

Otro modelo de cómputo interesante es el Juego de la Vida de Conway [20] que es igual de poderoso que todos los anteriores, en términos de la capacidad de cómputo [21]. Este modelo pretende describir una simulación de una colonia de bacterias. Se parte de un estado inicial que consiste en una colonia particular representada como una grilla donde en cada celda de la misma puede haber a lo sumo una bacteria. El cómputo consiste en modificar la colonia de acuerdo a ciertas reglas que indican cómo pasar de un estado al siguiente. Estas reglas aplican sobre cada bacteria de la colonia y pueden hacer que una bacteria determinada muera o se reproduzca en función de la cantidad de otras bacterias cercanas. El cómputo finaliza cuando la colonia se estabiliza (es decir, cuando un estado es idéntico al anterior y, por lo tanto, todos los estados siguientes serán idénticos también).

Es interesante reflexionar durante un momento sobre las implicaciones profundas en la existencia de modelos de cómputo universal.

La consecuencia más grande de la noción de modelo universal es que todo cómputo automatizable es expresable mediante uno de estos modelos. La tesis de Church-Turing [22] afirma que todos los modelos de cómputo universales pueden realizar exactamente los mismos cómputos<sup>16</sup>. Aunque no ha sido demostrada formalmente tampoco existen evidencias de que no sea cierta, y hay un amplio consenso en la comunidad científica de que efectivamente lo es. Actualmente no se conoce ningún cómputo que sea automatizable y no sea expresable por un modelo de cómputo universal, y de ser cierta la tesis de Church-Turing, los cómputos expresables por los modelos de cómputo universal son exactamente todos los cómputos automatizables.

Otra consecuencia importante es que alcanza con tener uno de estos modelos para poder representarlos a todos, ya que cualquiera de ellos es capaz de simular a cualquiera de los otros. Vimos que los dispositivos computacionales implementan algún modelo de cómputo, y como los dispositivos computacionales de propósitos generales implementan modelos de cómputo universal, cualquier dispositivo de cómputo de propósitos generales puede simular a cualquier modelo de cómputo y a cualquier otro dispositivo, como ya vimos. Es por ello que las computadoras de propósitos generales han impactado tan profundamente en la cultura y la sociedad moderna.

## 7. Lenguajes de programación como modelos de cómputo

Hasta ahora venimos hablando de “entidades computacionales” haciendo referencia exclusivamente a entidades físicas (ya sean personas o máquinas). Sin embargo, todo el análisis realizado aplica también a elementos virtuales. Las máquinas virtuales son un claro ejemplo: en un dispositivo de propósito general (un procesador) se puede simular otro dispositivo de propósito general (otro procesador). Y dado que el dispositivo simulado implementa también un modelo de cómputo universal (pues el dispositivo a simular es de propósito general), puede simular a otros dispositivos de propósito general (así como también a los de propósito específico). Pero en este caso es un “dispositivo virtual”, que, para distinguirlo de los físicos, denominaremos *artefacto computacional*. Para generalizar, podemos hablar directamente de “autómatas” como entidades (tanto físicas como virtuales) que realizan cómputos de forma automática. Así, los dispositivos computacionales (tanto las de propósito específico como las de propósito general) son autómatas pero también son autómatas aquellos artefactos computacionales que cumplen la misma función.

El caso de las máquinas virtuales es un ejemplo extremo, pero pueden encontrarse muchos otros ejemplos de simulaciones entre autómatas virtuales. De hecho, los lenguajes de programación pueden ser considerados también como autómatas<sup>17</sup> y, en la mayoría de los casos, como autómatas de propósito general. Cada lenguaje de programación expresa las soluciones de algún modelo de cómputo. Al igual que sucede con los dispositivos computacionales de propósito general, y a pesar de las diferencias que estos lenguajes puedan tener entre sí, es bien sabido que todos los lenguajes de programación (de propósito general) tienen la misma capacidad de expresar soluciones a problemas. Esto significa que cualquier lenguaje puede simular de alguna forma a cualquier otro (o en otras palabras, los modelos de cómputos asociados a los lenguajes de propósitos generales son modelos universales).

Además, para muchos de estos lenguajes se pueden construir entornos de trabajo que varían en su objetivo: pueden servir para asistir a un profesional en la construcción de programas complejos, o a un estudiante para aprender los fundamentos de la programación. Estos entornos presentan una variación similar a la explicada entre autómatas y computadoras en cuanto a su especificidad.

<sup>16</sup>La tesis de Church-Turing fue propuesta con ese nombre por Kleene en 1952 pero en realidad unifica 3 ideas en un único postulado. Cada una de ellas intentaba definir el concepto de “computable” a partir de un modelo de cómputo diferente; el Cálculo Lambda de Church en 1932 [15, 16], el modelo de Funciones Recursivas Generales de Gödel en 1934 [19] y la Máquina de Universal de Turing en 1936 [13, 14]. Los 3 modelos son equivalentes [23]. Para más información al respecto, ver la sección 6.

<sup>17</sup>Los lenguajes de programación en realidad no son más que especificaciones sintácticas para escribir programas de forma concreta. No pueden realizar cómputos como el resto de las entidades con las que venimos trabajando. Sin embargo cada uno de ellos tiene asociado un mecanismo de ejecución. En ese sentido, diremos que los lenguajes de programación son autómatas cuando en realidad nos estamos refiriendo a aquello que ejecuta el código escrito en ese lenguaje, que es lo que realmente efectiviza el cómputo (lo que generalmente se le llama un intérprete del lenguaje de programación). Así, cuando hagamos referencia a un lenguaje (o a un programa escrito en dicho lenguaje) como una entidad computacional, estaremos haciendo referencia en realidad a su ejecución por medio de un dispositivo; por ejemplo, al decir ‘Python’ en realidad estamos diciendo ‘la ejecución del intérprete de Python’, pero por conveniencia se utilizará la forma abreviada. En caso de querer referirnos al código del programa en sí o a las reglas del lenguaje (en el caso de Python), usaremos la denominación completa (el lenguaje de programación Python, o un programa escrito en Python).

Tomemos como ejemplo los entornos educativos PilasBloques [24] o Code.org [25]. Estos entornos, diseñados para dar los primeros pasos en programación, presentan una serie de actividades guiadas para que el estudiante aprenda a programar. Cada actividad implementa un autómata que sigue un modelo de cómputo particular (que suele expresarse a través de un personaje a controlar junto con las instrucciones que puede ejecutar). En cada uno de estos modelos de cómputo las operaciones son las instrucciones llamadas “primitivas” y la información que se procesa y transforma es el estado del escenario. Los modelos de cómputo en PilasBloques o Code.org están acotados. No se puede utilizar el autómata de una actividad para representar al de otra actividad. Pasa algo diferente con el entorno Gobstones [26, 27, 28], también orientado a la enseñanza de la programación. La diferencia es que el modelo que Gobstones implementa cumple con la característica antes mencionada que suelen tener todos los lenguajes de programación de propósito general y es que el modelo de cómputo subyacente es universal (tiene la capacidad de simular cualquier otro modelo). En concreto, podemos utilizar el modelo de cómputo de Gobstones para simular cada uno de los autómatas de PilasBloques pero no podemos simular el modelo de bolitas de Gobstones usando los modelos de PilasBloques. Esto significa que los modelos de cómputo que implementan los autómatas de PilasBloques no son universales. Sin embargo, tampoco están restringidos a un único comportamiento, en tanto se los puede programar para cumplir diferentes objetivos. Estos autómatas pueden tener propósitos variados, según el programa provisto, pero no entran en la definición de “autómata de propósito general” en tanto no tienen la capacidad de simular completamente a cualquier otro autómata (su modelo de cómputo asociado no es universal). Podemos clasificar entonces a los autómatas de propósito específico en “programables” y “no programables” según si su comportamiento depende de un programa o es fijo. En el caso de los autómatas de propósito general, son todos programables.

Una cuestión que quedó pendiente es determinar cuáles son los problemas que los autómatas de propósito general pueden resolver (es decir, cuáles son los cómputos que un modelo de cómputo universal puede realizar). Dijimos que todos los autómatas de propósito general tienen la misma capacidad de cómputo, ya sea que estemos hablando de dispositivos físicos como computadoras o de artefactos virtuales como el intérprete de Gobstones. También que todos estos autómatas pueden simularse entre sí. Sin embargo, reiteramos que todos aquellos cómputos realizables por estos autómatas son sólo aquellos que pueden ser automatizados. Algunos ejemplos de cómputos que se sabe que no son realizables por autómatas de propósito general (llamados “problemas no computables”) son determinar si una fórmula cualquiera de la lógica de primer orden es válida o no y conocer la cantidad de tiempo que va a tardar en ejecutarse un programa cualquiera (en un lenguaje cualquiera)<sup>18</sup>. Al conjunto de problemas que pueden ser resueltos por autómatas de propósito general se lo llama “problemas computables”. A los lenguajes de programación que implementan modelos de cómputo universales se los llama “lenguajes Turing-completos”. La gran mayoría de los lenguajes de programación son Turing-completos.

Teniendo esto en mente, podemos concluir que lo interesante no es aprender a expresar soluciones en un lenguaje particular, sea éste el que sea, sino a entender los principios que subyacen a la construcción de programas en cualquier lenguaje. Esto lleva a una paradoja, como fue presentado por Martínez López et.al. [29], la cual afirma que, al aprender a programar, el lenguaje no es importante porque no debería ser el foco de la enseñanza aprender un lenguaje particular pero a la vez es importante porque es necesario contar con algún lenguaje para que una solución pueda ser expresada.

## 8. Aspectos fundamentales de los Modelos de Cómputo

Existen ciertas características comunes a todos los modelos de cómputo. Elaboramos tres de los aspectos de los modelos de cómputo que consideramos fundamentales para comprender la naturaleza de las Ciencias de la Computación.

El primer concepto fundamental de los modelos de cómputo, especialmente para aquellos que se busca que sean universales, es la idea de **representación de información**. La representación consiste en utilizar ciertos elementos para representar otros elementos, y va más allá de la computación. Por ejemplo, cuando un niño pequeño toma un palo y juega a ser un pirata con una espada, el palo representa a la espada; si bien en realidad el palo no es una espada, a todos los efectos del juego, sí lo es: podemos decir que la espada es una **representación abstracta** cuya **implementación física** es el palo. De la misma forma, un modelo de cómputo debe tener la capacidad de representar cualquier tipo de información utilizando únicamente sus elementos básicos. Por ejemplo, en el modelo de cómputo de Gobstones, los elementos básicos son el tablero y las bolitas de colores. Pocas veces nos interesa resolver un problema en el universo de discurso en el que sólo existen bolitas de colores. Pero los elementos del dominio de Gobstones (las bolitas de colores), se pueden usar para representar elementos de otros dominios. Cuando introdujimos la noción de cómputo remarcamos que las transformaciones de información se hacían sobre cualquier tipo de información. Además, ya dijimos que todos los autómatas de propósito general pueden realizar los mismos cómputos así que es esperable que exista en Gobstones alguna forma de realizar, por ejemplo, la transformación asociada a la búsqueda en internet. En este ejemplo los elementos del dominio son las páginas de internet y las palabras, así que para describir tal cómputo en Gobstones se debe proponer alguna representación de esos elementos a partir de bolitas. Así como con las bolitas en Gobstones, las representaciones puede adquirir las formas más diversas. Sin embargo, la más usual (pero no la única) es utilizar números y grupos de números para representar diferentes elementos, ya sean abstracciones de elementos físicos (como cuerpos sólidos en una simulación, naves en un videojuego, etc.), o bien otros elementos abstractos (colores, listas, etc.). Otra forma de representación abstracta consiste en utilizar símbolos para transmitir ciertas ideas, como se hace en álgebra. Los modelos de cómputo universales permiten así representar cualquier elemento mediante algunos de sus elementos básicos, extendiendo su dominio de expresión a cualquier información que se deseé.

<sup>18</sup>El primer problema no computable fue descubierto por Turing y consistía en transformar la descripción de una máquina de Turing dada y sus datos de entrada en la determinación sobre si la ejecución de dicha máquina terminaría o no. Este problema se conoce habitualmente como el problema de la detención de la máquina de Turing (el *halting problem*) [13, 14].

Al pensar en un problema particular para ser resuelto con un cómputo es más adecuado pensar en términos de los elementos representados en el dominio del problema a resolver y no de su representación subyacente a partir de los elementos del dominio del modelo de cómputo. Por ejemplo, es usual representar imágenes mediante un “mapa de bits”, que no es otra cosa que una matriz de números que indican los valores de los colores de cada punto en la imagen. Ahora bien, al visualizar la imagen, no sirve que se nos presenten los números, sino que para poder apreciar la imagen debemos poder ver los colores. De la misma forma, al procesar la información, pensamos en “transformar a escala de grises” o en “aplicar un filtro sepia” o en “mejorar los contrastes”, en lugar de pensar en los procesamientos numéricos involucrados para lograr tales modificaciones. Esto está vinculado a la comprensión que las personas deben tener de los cómputos y los elementos representados.

*Comprender que la información se representa de ciertas formas en cada modelo concreto y cómo las operaciones del dominio se expresan en términos de esa representación subyacente es una parte importante a la hora de pensar en un programa.* Por ejemplo, la representación básica más difundida es la representación que utilizan las computadoras digitales, que permiten representar a los números mediante una secuencia de dígitos binarios. Esta representación, si bien es importante, es solamente una de las posibles formas de representación, y no debería tener privilegios a la hora de enseñarse, ya que entender cualquier forma específica y no más que esa forma es insuficiente para tener una idea completa de la Computación. De hecho, en los modelos de cómputo de los lenguajes de alto nivel los elementos básicos son números decimales y estructuras de datos y no dígitos binarios. Actualmente se utilizan los términos “computacional” y “digital” como si fuesen intercambiables, pero en realidad la denominación de “digital” solamente hace referencia, como vimos, a una forma de representar la información (y de construir dispositivos que tengan esa capacidad) y no está vinculada directamente a la idea de “computacional” como la discutimos en este artículo. Por eso creemos que es muy importante que quiénes enseñan computación comprendan las nociones que presentamos. Es necesario comprender la idea de representación y las posibilidades que tal noción ofrece dentro de la disciplina. Los ejemplos sirven para visualizar esta idea y para aprovecharla, pero para dominarla se requieren diferentes niveles según el grado de manejo que se desee.

Otro concepto importante referido a los modelos de cómputo es la forma en la que se establece cuál es el cómputo que queremos expresar en cierto modelo de cómputo. Así, un **programa** es *una descripción de cómo se deberá desarrollar un cómputo en cierto modelo de cómputo*. En los modelos de cómputo universales los programas se escriben en un lenguaje específicamente diseñado para expresar cómputos genéricos: un **lenguaje de programación**. En modelos más específicos, como los usados en biología computacional u otras ciencias, la manera de establecer cuál es el cómputo deseado, o sea, la noción de programa, puede variar, y consistir por ejemplo en una serie de ecuaciones matemáticas, o cierto conjunto de datos para usar como base para modificar el modelo de cómputo, etc. Pero dada la flexibilidad y generalidad provista por los modelos de cómputo universales, es enriquecedor conocer los fundamentos de los lenguajes de programación y de cómo expresar soluciones con ellos. La **Programación** es la rama de las Ciencias de la Computación que se encarga de expresar soluciones a problemas computacionales en algún lenguaje de programación. Sin embargo, resulta enriquecedor conocer también otros modelos de cómputo y las formas de expresar soluciones en ellos, ya sea un modelo para biología computacional, o uno para ciencias sociales computacionales, u otros, siempre teniendo en cuenta que no se trata de conocer un caso particular por sí mismo, sino en tanto ejemplo de una idea más general.

El tercer concepto que consideramos como pilar de la Computación es la noción de **eficiencia**. Dado un modelo de cómputo, la ejecución del procesamiento prescripto por un cómputo utilizará ciertos elementos e insumirá cierta cantidad de operaciones. Al implementar este modelo abstracto en una máquina física, la cantidad de elementos utilizados y la cantidad de operaciones a realizar implicarán un consumo de recursos, tales como el espacio de almacenamiento o el tiempo necesario, y para que la implementación de esta solución sea eficaz y útil debe poder ser realizable en un tiempo razonable y con una cantidad razonable de elementos. A esto se lo conoce con el nombre de *eficiencia: la utilización razonable de los recursos*, tanto los utilizados para representar la información, como la cantidad de operaciones necesarias para llevar adelante la transformación de tal información. El diseño de soluciones eficientes, es decir, de escribir programas que expresen el cómputo deseado, requiere también considerar si la ejecución de ese programa puede ser realizada de manera eficiente: no es lo mismo utilizar dígitos decimales para representar números que utilizar porotos, ya que la cantidad de dígitos decimales necesarios para representar un número es mucho menor que la cantidad de porotos (por ejemplo, para representar el número un millón hacen falta un millón de porotos pero solamente 10 dígitos decimales). La eficiencia puede ser temporal – cuánto tiempo llevan las operaciones necesarias para realizar el cómputo –, o espacial – cuántos símbolos se utilizan para representar la información. Por ejemplo en el caso de la representación de números dada antes consideramos la eficiencia espacial. Sin embargo es usual que a más símbolos se requieran más operaciones, por lo que ambas están relacionadas. Al trabajar con cómputos y modelos de cómputos, la eficiencia es un concepto guía, pues dos cómputos que resuelven el mismo problema en forma ideal no son necesariamente equivalentes en cuanto a su expresión concreta, si, por ejemplo, uno tarda segundos y el otro milenarios.

Por otro lado, la implementación física de estos modelos también requiere considerar usos eficientes de los elementos físicos: no es lo mismo utilizar lámparas (válvulas) como se hacía en las primeras computadoras electrónicas, antes de que existieran los chips de silicio, que hacerlo mediante un circuito integrado implementado con materiales superconductores; tampoco es lo mismo utilizar un carrete de cinta magnética continua para almacenar (representaciones de) información, que utilizar una memoria flash como las de un pendrive, que son mucho más eficientes tanto en tamaño como en velocidad. La eficiencia ha sido siempre una fuerza rectora en la evolución de la computación, tanto a nivel teórico dentro de los diferentes modelos de cómputo, como en su implementación física, en las máquinas que implementan estos modelos. Las computadoras digitales son el modelo de dispositivo predominante actualmente porque tanto la representación que utilizan

como la implementación física de las mismas son eficientes. De surgir otra representación u otro dispositivo que fuese más eficiente, podría reemplazarlo, como puede suceder con la computación cuántica.

Al considerar modelos de cómputo universales, la noción de eficiencia se vuelve determinante, porque al simular un autómata particular en términos de los elementos del modelo de cómputo universal utilizado, la eficiencia pocas veces es la misma que en el autómata representado. Esto lleva a que determinados modelos de cómputo sean más adecuados para expresar cierta tarea por sobre otros. También hace que encontrar implementaciones eficientes para representar un cómputo particular en términos de un modelo universal sea particularmente importante. Pero esto es un aspecto avanzado, que corresponde más a usos profesionales de la computación.

## 9. ¿Y qué pasa con la Inteligencia Artificial?

Gran parte de lo dicho hasta ahora pareciera contradecirse con el comportamiento de los sistemas modernos de Inteligencia Artificial. Una explicación detallada de por qué esto no es así requeriría un nuevo artículo (puede consultarse el que publicó recientemente la Fundación Sadosky [30]), pero a grandes rasgos, lo diferente no reside en los cómputos que realizan, sino en la forma de describirlos. En la programación tradicional el programa describe las reglas de transformación en función de la entrada, por lo que resulta factible comprender cuáles son los pasos que el modelo de cómputo sigue para efectivizar el cómputo y producir la salida (a pesar de que sean muchos pasos, y la complejidad pueda ser grande). En cambio, en un sistema basado en Inteligencia Artificial moderno, especialmente los que se definen con aprendizaje automático, el programa no expresa las reglas que relacionan el dato de entrada con el de salida directamente, sino que utilizan un cómputo basado en ciertos parámetros internos que se ajustan automáticamente para imitar las respuestas de un montón de datos para los que sí se conoce cuál debería ser la salida correspondiente. Al proceso de ajustar los parámetros internos se lo conoce como “fase de entrenamiento” del sistema, luego del cual esos parámetros expresan la relación de la entrada con la salida de una forma que no es simple de comprender como sí podría hacerse con un programa tradicional.

Dado que expresar directamente la relación de los datos de entrada con el dato de salida como se hace en la programación tradicional requiere que una persona o equipo de personas (los programadores) comprendan la relación exacta y la expresen en forma de programa, pero al expresar la relación de los datos de entrada con el dato de salida en forma implícita mediante parámetros que se calculan automáticamente no requiere esa comprensión, es factible abordar cómputos mucho más complejos que dan la apariencia de que el sistema es inteligente. Para mayor detalle sobre esto, consultar el artículo de la Fundación Sadosky [30].

Completando esta reflexión, vale la pena volver a destacar que un sistema basado en Inteligencia Artificial también expresa un cómputo utilizando un modelo de cómputo, y, aunque no lo hace en la forma tradicional, ese cómputo no escapa a las consideraciones que analizamos. Solamente muestra que aún hay mucho camino por recorrer para comprender las maneras de expresar cómputos a través de entidades computacionales.

## 10. Conclusiones

Reflexionamos sobre la naturaleza de la Computación como disciplina y sobre las Ciencias de la Computación en particular. Consideramos sus orígenes tanto del lado teórico como técnico. Es importante destacar que ambas prácticas se desarrollaron a la par y se enriquecieron mutuamente. Por el lado teórico, la Teoría de la Computación surge a partir del interés de matemáticos por determinar qué problemas se pueden resolver de forma mecánica. Para ello desarrollaron lenguajes de programación que permitieran describir soluciones a tales problemas. Las computadoras son motivadas por la construcción de máquinas que realizaran cómputos de forma automática.

Identificamos el objeto de estudio de la Computación: los cómputos, es decir los procesos de transformación de información. Clasificamos estos procesos según si pueden ser automatizables o no, y nos concentraremos únicamente en aquellos que pueden ser expresados de forma precisa. Definimos entonces el modelo de cómputo como elemento fundamental de la Computación, el cual nos permite explicar y expresar formalmente los cómputos.

Analizamos y discutimos distintas acepciones de la palabra “computadora”. En sus orígenes, se usaba para describir cualquier entidad que pudiera hacer cómputos. Hoy en día se utiliza principalmente para referirse a computadoras personales, notebooks y otros dispositivos similares. Dentro de la disciplina, solemos usarla para hablar de lo que aquí llamamos “autómatas de propósito general”, aunque en muchos casos nos referimos simplemente a “autómata programable” (por ejemplo, cuando decimos que programar es decirle a “la computadora” lo que tiene que hacer). En ciertas ocasiones englobamos también a los autómatas de propósito específico (como los electrodomésticos) en la definición, siempre que sean programables. Más allá de qué definición usemos, está claro que es importante conocer estos distintos niveles cuando estemos enseñando sobre Computación y/o Programación para evitar confusiones.

También categorizamos a los autómatas y a sus correspondientes modelos de cómputo. Para resumir, entendemos por “autómata” cualquier dispositivo físico o virtual capaz de realizar cómputos de forma automática. Las personas son “computadoras” (en tanto pueden realizar cómputos) pero no son “autómatas” (en tanto no lo hacen de forma automática). Cada autómata tiene asociado un modelo de cómputo que explica cómo el autómata procesa información. Los dispositivos computacionales (como las computadoras, los robots, etc.) son autómatas físicos. Los artefactos computacionales (como

los sistemas operativos, los lenguajes de programación y otros programas) son autómatas virtuales<sup>19</sup>. Tanto los autómatas físicos como los virtuales pueden distinguirse según su propósito. Algunos son “programables” (si su comportamiento puede ser alterado a partir de una especificación, el programa) y otros no. Si son lo suficientemente flexibles como para que su comportamiento pueda pasar a ser el de cualquier otro autómata (usando el programa correcto) entonces decimos que son “de propósito general”. Si son programables pero están limitados (o si directamente no son programables en lo absoluto) entonces decimos que son de propósito específico.

Los lenguajes de programación son (o más bien, pueden ser vistos como) autómatas virtuales programables<sup>20</sup>. Los de mayor difusión y uso son de propósito general. Cuando se dice que todos los lenguajes tienen el mismo poder de cómputo se suele entender que “pueden hacer lo mismo” aunque en realidad la frase se refiere a que pueden realizar los mismos cálculos. Así como sucedía con las computadoras (que una computadora no puede lavar ropa pero puede realizar las mismas transformaciones de información que un lavarropas), con los lenguajes sucede lo mismo. No es cierto que todos los lenguajes pueden “hacer lo mismo” (los lenguajes que no tienen memoria por ejemplo no pueden “guardar” una variable en memoria) pero en términos de transformaciones de información, cualquier cálculo que pueda ser expresado en un lenguaje particular puede ser expresado en cualquier otro lenguaje (que también sea de propósito general).

Una vez más, siendo estos temas aspectos tan centrales a la Computación, creemos que es indispensable que si alguien quiere enseñarla, aunque sea sólo a escribir código, tenga presentes estas ideas y pueda abordar la enseñanza desde un punto de vista más global, considerando a los lenguajes como modelos de cálculo y a los programas como formas de expresar transformaciones de información, en lugar de acotar todas las definiciones a un modelo de cálculo particular o una implementación de computadora específica.

## Agradecimientos

Agradecemos a Mara Borchardt, Magdalena Garzón y Gabriel Baum por sus valiosas sugerencias al revisar el trabajo.

## Referencias

- [1] E. W. Dijkstra. «On a cultural gap». En: *The Mathematical Intelligencer*. Vol. 8. 1. Springer-Verlag New York, 1986, págs. 48-52.
- [2] D. E. Knuth. «Ancient Babylonian algorithms». En: *Communications of the ACM* 15.7 (1972), págs. 671-677.
- [3] D. E. Knuth. «Algorithms in modern mathematics and computer science». En: *Algorithms in Modern Mathematics and Computer Science*. Springer, Berlin / Heidelberg, Germany, 1981, págs. 82-99.
- [4] P. J. Denning y M. Tedre. «Computational Thinking: A Disciplinary Perspective». En: *Informatics in Education* 20.1 (2021), págs. 361-390. DOI: 10.15388/infedu.2021.21.
- [5] D. Hilbert. «Mathematical problems». En: *Bull. Amer. Math. Soc.* 8 (1902), págs. 437-479.
- [6] D. Saha, P. Brooker, M. Mair y S. Reeves. «Thinking like a machine: Alan Turing, computation and the praxeological foundations of AI». En: *Science & Technology Studies* 37.2 (2024), págs. 66-88. DOI: 10.23987/sts.122892.
- [7] J. Herstad y A. Mørch. «Mutual Configuration: Exploring the Dynamic Interplay of Human-Computer Interaction as a Socio-Technical System». En: *Proceedings of the 8th International Workshop on Cultures of Participations in the Digital Age (CoPDA 2024): Differentiating and Deepening the Concept of “End User” in the Digital Age, June 2024, Arenzano, Italy* (2024).
- [8] M. Tedre. *The Development of Computer Science: A Sociocultural Perspective*. University of Joensuu, 2006.
- [9] J. Van Dijck. «Datafication, dataism and dataveillance: Big Data between scientific paradigm and ideology». En: *Surveillance & society* 12.2 (2014), págs. 197-208.
- [10] J. E. Savage. *Models of Computation: Exploring the Power of Computing*. Brown University, 1998.
- [11] A. Aho. «Computation and Computational Thinking». En: *Ubiquity* 2011.January (2011). URL: <http://ubiquity.acm.org/article.cfm?id=1922682>.
- [12] P. J. Denning y M. Tedre. *Computational Thinking*. The MIT Press Essential Knowledge Series, 2019. ISBN: 9780262536561.
- [13] A. M. Turing. «On Computable Numbers, with an Application to the Entscheidungsproblem». En: *Proceedings of the London Mathematical Society* 2-42.1 (1936), págs. 230-265. DOI: 10.1112/plms/s2-42.1.230.
- [14] A. M. Turing. «On Computable Numbers, with an Application to the Entscheidungsproblem: A correction». En: *Proceedings of the London Mathematical Society* 2-43.6 (1938), págs. 544-546. DOI: 10.1112/plms/s2-43.6.544.

<sup>19</sup>Recordemos que cuando hablamos decimos que un programa o, más específicamente, un lenguaje de programación es un autómata no nos referimos al programa o al lenguaje en sí, sino a su ejecución en el autómata sobre el que se ejecutan.

<sup>20</sup>Recordemos que al hablar de un lenguaje de programación como autómata nos referimos en realidad al programa que ejecuta programas en ese lenguaje.

- [15] A. Church. «A Set of Postulates for the Foundation of Logic (Part I)». En: *Annals of Mathematics* 33.2 (1932), págs. 346-366.
- [16] A. Church. «A Set of Postulates for the Foundation of Logic (Part II)». En: *Annals of Mathematics* 34.4 (1933), págs. 839-864.
- [17] H. B. Curry. «Grundlagen der Kombinatorischen Logik». En: *American Journal of Mathematics (in German)* 52.3 (1930), págs. 509-536. DOI: 10.2307/2370619.
- [18] H. B. Curry y R. Feys. *Combinatory Logic. Vol. I.* Amsterdam: North Holland, 1958. ISBN: 0-7204-2208-6.
- [19] K. Gödel, S. C. Kleene y J. B. Rosser. *On Undecidable Propositions of Formal Mathematical Systems.* <https://api.semanticscholar.org/CorpusID:117048598>. 1934.
- [20] M. Gardner. «Mathematical Games - The fantastic combinations of John Conway's new solitaire game 'life'». En: *Scientific American* 223.4 (1970), págs. 120-123. DOI: 10.1038/scientificamerican1070-120.
- [21] P. Chapman. *Life Universal Computer.* 2002.
- [22] S. C. Kleene. *Introduction to metamathematics.* D. Van Nostrand Company Inc., New York, 1952.
- [23] J. B. Rosser. «An Informal Exposition of Proofs of Gödel's Theorem and Church's Theorem». En: *The Journal of Symbolic Logic* 4.2 (1939), págs. 53-60. DOI: 10.2307/2269059.
- [24] A. Sanzo, F. Schapachnik, P. Factorovich y F. Sawady O'Connor. «Pilas Bloques: A scenario-based children learning platform». En: *2017 Twelfth Latin American Conference on Learning Technologies* (2017), págs. 1-6.
- [25] F. Kalelioğlu. «A new way of teaching programming skills to K-12 students: Code.org». En: *Computers in Human Behavior* 52 (2015), págs. 200-210.
- [26] P. E. Martínez López. *Las bases conceptuales de la programación. Una nueva forma de aprender a programar.* Ebook (1era ed.). La Plata, el autor, 2013. ISBN: 978-987-33-4081-9.
- [27] P. E. Martínez López, D. Ciolek, G. Arévalo y D. Pari. «The Gobstones method for teaching computer programming». En: *XXV Simposio de Educación Superior en Computación (SIESC'17), dentro de la XLIII Conferencia Latinoamericana de Informática (CLEI'17)* (2017), págs. 1-9.
- [28] P. E. Martínez López, F. Aloia, D. A. Ciolek, F. Martínez, D. Pari y P. Tobia. *Ciencias de la computación para el aula: 1er. ciclo de secundaria (libro para docentes).* Ciudad Autónoma de Buenos Aires, Argentina: Fundación Sadosky, 2019. URL: [http://program.ar/descargas/cc\\_para\\_el\\_aula-1er\\_ciclo\\_secundaria.pdf](http://program.ar/descargas/cc_para_el_aula-1er_ciclo_secundaria.pdf).
- [29] P. E. Martínez López, E. A. Bonelli y F. A. Sawady O'Connor. «El nombre verdadero de la programación: Una concepción de enseñanza de la programación para la sociedad de la información». En: *Anales del 10mo Simposio de la Sociedad de la Información (SSI'12), dentro de las 41ras Jornadas Argentinas de Informática (JAIIO '12)* (2012), págs. 1-23. ISSN: 1850-2830.
- [30] M. J. Gómez, P. E. Martínez López, J. Dabbah y M. Borchardt. *Diez preguntas frecuentes y urgentes sobre Inteligencia Artificial.* Fundación Sadosky, 2024. URL: <https://program.ar/wp-content/uploads/2024/08/Diez-preguntas-frecuentes-y-urgentes-sobre-Inteligencia-Artificial.pdf>.

## Glosario

- **Artefacto computacional:** Cualquier *entidad computacional virtual* capaz de realizar los cómputos de manera automática (es decir, un *autómata computacional* virtual).
- **Autómata:** Entidad (física o virtual) que funciona **de manera automática**.
- **Autómata computacional:** *Autómata* que **realiza algún cálculo** – i.e. la intersección entre *autómata* y *entidad computacional*.
  - **– de propósito específico no programable:** *Autómata computacional* cuyo funcionamiento **no puede ser alterado** para modificar el cálculo que realiza.
  - **– de propósito específico programable:** *Autómata computacional* cuyo funcionamiento **puede ser alterado** (es decir, es **programable**) para modificar el cálculo que realiza **pero no lo suficiente** como para realizar **cualquier cálculo**.
  - **– de propósito general:** *Autómata computacional* cuyo funcionamiento **puede ser alterado** (es decir, es **programable**) para que realice **cualquier cálculo**. Tiene un modelo de cálculo **universal** asociado.
- **Dispositivo computacional:** Cualquier *entidad computacional física* capaz de realizar los cómputos de manera automática (es decir, un *autómata computacional* físico).
- **Entidad computacional:** Entidad (física o virtual) capaz de **realizar algún tipo de cálculo**. Tiene un modelo de cálculo asociado.

# Evaluación de Habilidades de Pensamiento Computacional: avances, tendencias y resultados

Marcela Daniele<sup>1</sup>, Teresa Quintero<sup>2</sup>, Francisco Bavera<sup>1</sup>

<sup>1</sup>Departamento de Computación, Facultad de Cs. Ex., Fco.-Qcas. y Naturales, UNRC, Río Cuarto, Argentina

<sup>2</sup>Departamento de Física, Facultad de Cs. Ex., Fco.-Qcas. y Naturales, UNRC, Río Cuarto, Argentina

[marcela@dc.exa.unrc.edu.ar](mailto:marcela@dc.exa.unrc.edu.ar) , [tquintero@exa.unrc.edu.ar](mailto:tquintero@exa.unrc.edu.ar)

## Resumen

Se presenta una revisión de artículos de evaluación de Habilidades de Pensamiento Computacional (HPC) que exponen diferentes métodos e instrumentos. Además, se presentan algunas investigaciones sobre HPC desarrolladas hasta el momento por el grupo de investigación, en el marco de una serie de formaciones dictadas para docentes de los distintos niveles del sistema educativo argentino. Se incluye una breve discusión sobre las definiciones de pensamiento computacional para luego exhibir una revisión de métodos y técnicas de evaluación de HPC. Se detectaron una gran cantidad de propuestas, métodos, instrumentos y criterios de evaluación de HPC que dificulta encontrar formas establecidas y realizar un análisis más acabado del campo de investigación abordado. Se observa la necesidad de avanzar en estudios y análisis comparativos de los métodos de evaluación de HPC, a fin de comparar características fundamentales que contribuyan a la definición de estándares para la medición y evaluación de estas habilidades claves para resolver problemas de manera estructurada y eficiente.

**Palabras clave:** Pensamiento computacional, Revisión, Métodos de evaluación, Formación docente, Habilidades Cognitivas

## 1. Introducción

Aunque se remonta a tiempos antiguos, el pensamiento computacional se entiende como un serie de habilidades recientes que todos requieren, que se puede usar para resolver los problemas complejos y que está incluida en los estándares internacionales y en los programas de capacitación.

Desde las décadas de los sesenta y setenta, el matemático Seymour Papert [1] hablaba sobre lo que él mismo denominaba *pensamiento procedimental*. Abogó por que los niños pudieran aprender a pensar de manera diferente, fomentando el pensamiento procesual a través de la programación. En 1986 durante una conferencia Papert expresó: “*Nadie sabe cómo será el futuro de las escuelas, pero sí sabemos lo que no será. No habrá un montón de niños sentados en sus mesas con lápiz y papel escribiendo todo el día*”. No obstante, fue recién en el año 2006, cuando Wing [2] comienza a hablar de “*Pensamiento Computacional*” (PC) y propone una definición que sentó las bases a amplias y prometedoras discusiones en torno a la definición del término, a la incorporación del PC en los diseños curriculares desde la educación primaria, a la indiscutible necesidad de la formación de los educadores, a las mediciones y evaluaciones requeridas para validar los resultados y mejorar las propuestas.

En una amplia revisión de artículos [2][8][10][12] [13] [14][15][16] [17], puede verse que los autores coinciden de que aún existen importantes controversias para lograr una definición consensuada de PC. No obstante, existen coincidencias en los términos que forman parte de dicha definición como son: resolución de problemas, descomposición, abstracción, generalización, patrones y algoritmos o soluciones algorítmicas. De la misma manera, existen propuestas que avanzan en las mediciones de las habilidades de pensamiento computacional construidas tanto por niños, niñas, adolescentes como por educadores que están recibiendo distintas formaciones en la temática.

Esta revisión de artículos surge, en un punto, como parte del trabajo que este equipo de investigación viene aportando y en otro, como forma de compartir con los colegas una síntesis de lo encontrado y reflexionado en el camino de desandar y andar la noción de PC y las habilidades de PC. Para ello partimos de situar al equipo de investigación en su ámbito y sus desarrollos, luego se presenta un recorrido por diversas definiciones de PC, que destacan diferentes procesos mentales involucrados en su desarrollo y consolidación, para luego presentar la revisión de una serie de métodos de evaluación de las Habilidades de PC tanto en estudiantes, docentes cómo en actividades propuestas.

## 1.1. PC en la UNRC

En el año 2016, en el ámbito de la Facultad de Ciencias Exactas, Físico-Químicas y Naturales de la Universidad Nacional de Río Cuarto, se conformó un equipo de trabajo multidisciplinar integrado por informáticos, físicos, matemáticos y pedagogos. Algunos miembros con sólida formación en didáctica de las ciencias. Como equipo se dedicaron al diseño de una *Especialización Docente de Nivel Superior en Didáctica de las Ciencias de la Computación – EDNSDCC-* (Res. Min Ed. Prov. de Córdoba N° 210/2017) dirigida a docentes de educación primaria, con una carga horaria total de 400 horas distribuidas en ocho módulos y con una duración de dos años. Entre los años 2018 y 2019, el equipo implementó la Especialización Docente de Nivel Superior en Didáctica de las Ciencias de la Computación, como una iniciativa conjunta de la Fundación Sadosky, el Instituto Superior de Formación Docente Ramón Menéndez Pidal y la Facultad de Ciencias Exactas Físico-Químicas y Naturales de la Universidad Nacional de Río Cuarto. Esta instancia de formación destinada a docentes de Nivel Primario tuvo como objetivo introducir a los docentes en el Pensamiento Computacional, las Ciencias de la Computación y la programación con el fin de brindarles las herramientas y experiencias necesarias para que puedan replicarlo en sus aulas.

Esta fue la principal motivación del equipo que marcó un comienzo en la investigación de metodologías y propuestas orientadas a la construcción del PC en la formación continua de docentes de escuela primaria. La formación docente continua en torno a capacitaciones que abordan la didáctica de las ciencias de la computación, permite caracterizar la construcción del PC en los docentes en formación y evaluar el impacto sobre la transformación de sus propias prácticas. Persiguiendo el cumplimiento de estos objetivos planteados, surgen diversos interrogantes que dan marco al proceso de investigación, ¿Cuáles han sido los avances en la construcción del PC en el sistema educativo mundial y local? ¿Cuáles son las herramientas más utilizadas para su desarrollo? ¿Cuáles propuestas de formación de formadores existen? ¿Construyen el pensamiento computacional los maestros de primaria en el contexto actual? ¿Cómo lo hacen? ¿Se produjo una transformación en sus prácticas áulicas a partir de la formación recibida? ¿Impacta en el aprendizaje de sus estudiantes? ¿Se pueden retroalimentar las propuestas de formación docente con los resultados obtenidos? ¿cómo medir, cómo evaluar los métodos existentes?

Las investigaciones realizadas por este equipo fueron gradualmente arrojando apreciables resultados en torno al desarrollo del PC en la formación continua de docentes de educación primaria en didáctica de las ciencias de la computación y la evaluación del impacto de éstas capacitaciones en sus aulas de escuela primaria. Los primeros resultados surgieron a partir del análisis de productos de integración realizados por los docentes de educación primaria, al finalizar el primer año de cursado de la EDNSDCC, utilizando una metodología cualitativa, basada en la teoría fundamentada, cuyo análisis permitió encontrar que en todas las producciones y las propuestas para sus prácticas áulicas, se vieron reflejadas habilidades cognitivas relacionadas al mundo digital, mostrando diferentes niveles de desarrollo y profundidad [18]. También se realizaron estudios dando cuenta del rol de cierta clase de problemas en la construcción de procesos de generalización, tomando un problema “ejemplar” trabajado en el segundo módulo (introducción a la resolución de problemas) dictado en la EDNSDCC, detallando un análisis didáctico-matemático, para visibilizar las características de los problemas que favorecen el desarrollo de capacidades vinculadas al PC. Al mismo tiempo, se examinan aquellos que permitan mostrar, a partir de la reflexión sobre los mismos, la fortaleza de la sistematización de la información de casos particulares, las diferentes maneras de registrar y, a partir de la necesidad impuesta por los sucesivos problemas, la potencialidad de unos registros sobre otros [19]. Posteriormente, se utilizaron Problemas Bebras como herramienta de recolección de datos para medir habilidades de PC adquiridas por este grupo de docentes, lo que permitió identificar que un alto porcentaje, pudo resolver problemas de complejidad media implicando habilidades, como, abstracción, reconocimiento de patrones, modelos y simulación, algoritmos y descomposición [20] [21][22].

Dando continuidad a la investigación iniciada, le surge a este equipo de trabajo la necesidad de revisar y analizar un importante número y variedad de trabajos recientes que aportan métodos, técnicas, herramientas y resultados que abordan la discusión en torno a la definición de PC, aportes, avances en la inclusión en la currícula escolar, y fundamentalmente, trabajos que arrojan aportes sobre medición y evaluación de las habilidades del PC a partir de propuestas formadoras en ciencias de la computación. Este trabajo aporta una revisión y comparativa de métodos, herramientas y resultados de medición y evaluación de la construcción de habilidades de PC.

## 2. Pensamiento Computacional

### 2.1. Definiciones de PC

El pensamiento computacional se ha convertido en una herramienta esencial en la educación moderna. Las primeras aproximaciones de este concepto están vinculadas a la adquisición de habilidades cognitivas que se desarrollan a través de la programación y el diseño de algoritmos. La integración del pensamiento computacional en el currículo educativo ha ganado popularidad a nivel mundial, ya que posibilita preparar a los estudiantes para enfrentar los desafíos del siglo XXI, de un escenario mundial globalizado, en el que la tecnología y sus avances invaden la vida del hombre de hoy.

El entendimiento y uso de técnicas computacionales, a menudo desarrolladas a través de la programación de códigos funcionales para el procesamiento de datos y la resolución de problemas lógicos, son esenciales para adquirir competencias básicas. Estas competencias están relacionadas con las estructuras y conexiones mentales que facilitan el pensamiento de orden superior. Para muchos Papert es el padre del Pensamiento computacional, él fue co-fundador del Laboratorio de Inteligencia Artificial del Instituto Tecnológico de Massachusetts. Es conocido por sus trabajos con lenguajes de programación visual, y como creador del lenguaje de programación de alto nivel LOGO, diseñó un lenguaje funcional y estructurado, que es fácil de aprender, ya que su objetivo principal era obtener resultados educativos.

La definición formal de Pensamiento Computacional, o Computational Thinking, fue introducida por primera vez en la comunidad científica por Wing [2] en 2006. Wing definió este término como una forma de pensar que no se limita exclusivamente a programadores o científicos en computación, sino que abarca un conjunto de habilidades útiles para todas las personas. Esta definición también incluye una amplia gama de herramientas mentales que reflejan y destacan el vasto potencial individual. Para Wing [2] se trata de un grupo de herramientas mentales con las que toda persona debería contar y aplicar para resolver situaciones. Esta primera definición es reconocida por defender la incorporación del PC en la educación de todo ser humano, lo que tiene un impacto para el profesorado en todas las áreas del conocimiento, como así también en los distintos niveles educativos.

Según Cuny, Snyder y Wing [10] el PC es una actividad mental para formular problemas de forma que admitan una solución computacional; es el proceso de pensamiento involucrado en la formulación de un problema y la expresión de sus soluciones de tal manera que una computadora, humano o máquina, puede llevar a cabo eficazmente. En esta definición se encuentra un mayor énfasis en la actividad mental.

Por otra parte, Wing [5] en el 2010 expresa que el PC son los procesos de pensamiento involucrados en la formulación de problemas y sus soluciones, de modo tal que las soluciones se representen de forma que pueda ser desarrollada efectivamente por un agente de procesamiento de información.

Para el ISTE y CSTA [8] el pensamiento computacional es un enfoque para resolver un determinado problema que con la integración de tecnologías digitales con ideas humanas. Considera la creatividad, el razonamiento o pensamiento crítico, pero refuerza esas habilidades al tiempo que realiza formas de organizar el problema de manera que la computadora pueda ayudar.

En 2012, The Royal Society [11], pone el énfasis en el proceso de pensamiento, considerando al PC como El proceso de identificar aspectos de la informática en el entorno que nos rodea y aplicar herramientas y técnicas informáticas para comprender y analizar sistemas y procesos, tanto naturales como artificiales.

Se encuentra que el PC puede ser una forma de pensar para resolver problemas de manera efectiva y eficiente un problema [12], un proceso mental [13], estrategias para resolver problemas [14], o habilidades de pensamiento [15], [16]. Los avances tecnológicos y los análisis sobre las competencias de alfabetización digital fomentan la aparición de nuevos elementos relacionados con el pensamiento computacional. Aunque se han desarrollado proyectos para su inclusión en el currículo de diversos niveles educativos, no existe una definición formal del término. Por ello, estas iniciativas varían en contenido y naturaleza [17], por lo que se hace necesario conocer las diferentes definiciones de PC y las distintas metodologías de evaluación de las habilidades de PC (HPC) en relación a esas definiciones.

### 2.2. Habilidades de PC

Si bien no existe un consenso estricto sobre la definición de Pensamiento Computacional, la mayoría de los autores coinciden en una serie de habilidades comunes [2-9]. Estas habilidades incluyen abstracción, pensamiento algorítmico, reconocimiento de patrones, descomposición, generalización, análisis lógico y evaluación.

La *Abstracción* es definida como el proceso de reducir los detalles innecesarios de un artefacto para hacerlo más comprensible. La habilidad en la abstracción reside en la elección del detalle a ocultar de manera que el problema se

vuelva más fácil, sin perder lo importante. Una parte fundamental de la misma es la elección de una buena representación de un sistema.

El *Pensamiento Algorítmico* como una forma de llegar a una solución a través de una definición clara de pasos que abordan el problema.

El *Reconocimiento de Patrones* permite identificar y clasificar datos en categorías basadas en sus características principales, a partir de reglas, principios y patrones observados en datos. Es parte del análisis de datos que ayuda a discernir características comunes en los datos que pueden clasificarse y organizarse para una fácil interpretación [9].

La *Descomposición*, una manera de pensar acerca de los artefactos en términos de sus partes y componentes. Cada pieza debe entenderse, solucionarse, desarrollarse y evaluarse por separado, facilitando la resolución de problemas y garantizando una mejor solución a problemas complejos y grandes.

La *Generalización* asociada a la identificación de patrones, similitudes y conexiones, y la explotación de estas características para transferir el proceso de solución de problemas a una gran diversidad de contextos. Es una forma de resolver rápidamente los nuevos problemas sobre la base de las soluciones en los problemas anteriores, y la construcción en la experiencia previa. Haciendo preguntas tales como "¿Esto es similar a un problema que ya he solucionado?" y "¿Cómo es diferente?" Algoritmos que resuelven algunos problemas específicos se pueden adaptar para resolver toda una clase de problemas similares [9].

El *Análisis Lógico* consiste en aplicar e interpretar la lógica (booleana) de manera correcta. La *Evaluación* consiste en sistematizar (a través de distintos criterios) y hacer un juicio de valor. Se incluyen dentro de esta habilidad la depuración, validación y verificación de las soluciones a problemas. Involucra el análisis lógico para predecir y verificar los resultados.

### 3. Evaluación de las HPC

#### 3.1. Metodología

Esta investigación es de tipo descriptiva documental, ya que las fuentes utilizadas para la obtención de la información estuvieron constituidas por textos de investigaciones previas. Para la búsqueda de la bibliografía se utilizaron los buscadores digitales Google Scholar, Science Direct y la red social científica Researchgate, desde el año 2015 hasta julio de 2024. Se revisaron una serie de artículos seleccionados en función de las palabras claves incluyendo fuentes impresas y electrónicas, artículos en revistas y congresos, libros y páginas web. De estas fuentes se extrajeron los métodos de evaluación de las HPC utilizados que se sistematizan en el siguiente apartado.

#### 3.2. Revisión de Métodos de evaluación de las Habilidades de PC

Se presentan los resultados obtenidos de la revisión y el análisis documental sobre los métodos de evaluación de habilidades de pensamiento computacional (HPC), se seleccionaron para compartir seis casos, de diferentes procedencia, para dar un panorama de lo encontrado. Los resultados de estos artículos seleccionados se organizan en la tabla 1.

Tabla 1: Artículos sobre métodos de evaluación de HPC seleccionados

Artículo	Objetivos	Métodos	Población	Principales resultados
[23] Kong, SC. (2019). Components and Methods of Evaluating Computational Thinking for Fostering Creative Problem-Solvers in Senior Primary School Education. In: Kong, SC., Abelson, H. (eds) Computational Thinking Education. Springer, Singapore.	Identificar estudios relacionados con la programación y la evaluación del PC para contribuir a la propuesta de componentes y métodos de evaluación apropiados para la educación primaria superior	Estudio documental a partir de la información de estudios anteriores (desde 2010 a mayo de 2019) para identificar los componentes y métodos para evaluar el desarrollo del PC de los estudiantes.	Estudiantes de educación primaria superior.	Resume los componentes de los conceptos de PC que fueron evaluados por estudios anteriores y los resultados. Los estudios evaluados usaron métodos tanto cuantitativos como cualitativos para medir. La revisión indicó que además de los conceptos y prácticas para medir el desarrollo de PC de los estudiantes, son relevantes las perspectivas de los estudiantes sobre el

				aprendizaje de la programación.
[24] Román-Gonzalez, et al. Test de Pensamiento Computacional: diseño y psicometría general. Octubre 14-16, 2015, Madrid, ESPAÑA III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2015)	Diseño y aplicación de un <i>Test de Pensamiento Computacional (TPC)</i> que pretende medir el nivel de aptitud-desarrollo del PC en el sujeto.	Medidas pre-test del nivel inicial	400 estudiantes españoles de entre 12 y 13 años.	
[25] A. D. F. Pérez and G. M. Valladares, Development and assessment of computational thinking: A methodological proposal and a support tool, 2018 IEEE Global Engineering Education Conference (EDUCON), Santa Cruz de Tenerife, Spain, 2018, pp. 787-795, doi: 10.1109/EDUCON.2018.8363311	Definición de una metodología para el desarrollo del PC y la posterior medición del desarrollo obtenido en los estudiantes.	Un pre-test y un post-test con un grupo experimental y un grupo de control. Se propusieron actividades específicas seleccionadas de las pruebas Bebras para desarrollar el PC y pruebas que permiten analizar en qué medida se ha desarrollado de HPC.	Estudiantes preuniversitarios y docentes que necesariamente no cuentan con conocimientos avanzados de Informática	Generación de una plataforma que presenta todos los conceptos necesarios para desarrollar y <b>medir el pensamiento computacional</b> y que ha sido desarrollada y pensada exclusivamente para ello, por lo que ofrece un servicio, un producto y una dedicación muy específica para este fin. destaca es que esta plataforma está preparada para que cualquier persona, docente o no, pueda llegar a impartir un curso sin necesidad de tener unos conocimientos determinados en la materia, tan solo querer enseñar y fomentar este tipo de habilidades.
[26] C. Otero Avila, L. Foss, A. Bordini, M. Simone Debacco and S. A. da Costa Cavalheiro, "Evaluation Rubric for Computational Thinking Concepts," 2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT), Maceio, Brazil, 2019, pp. 279-281, doi: 10.1109/ICALT.2019.00089.	Creación y aplicación de una rúbrica para medir la presencia cualitativa de los conceptos de abstracción, descomposición, generalización y pensamiento algorítmico, en actividades que se proponen para el desarrollo del PC	Se diseñó una rúbrica estructurada en 5 niveles de manera que sea posible identificar y medir un valor cualitativo para su aplicación, considerando la taxonomía de Bloom. Estudio de caso: se aplicó la rúbrica a tres actividades propuestas previamente en la literatura.	Profesores de Informática en la evaluación de actividades del proyecto CS Unplugged.	Encontraron que todos los criterios (conceptos del PC) fueron identificados en las actividades analizadas. En mayor o menor medida, los evaluadores identificaron la presencia de todos los conceptos en las actividades evaluadas. Ninguna de las actividades alcanzó los niveles más altos (4 y 5) de la rúbrica. Este hecho no descalifica a las actividades evaluadas, pues lo más importante es que sean adecuadas a las metas y al público objetivo. Además el resultado de la clasificación de una actividad sometida a la rúbrica puede estimular a los docentes autores de este tipo de material didáctico a incluir algunas competencias consideradas en los niveles superiores de la rúbrica.
[27] Sun, L., You, X. & Zhou, D. Evaluation and development of STEAM teachers' computational thinking skills: Analysis of multiple influential factors. Educ Inf Technol 28, 14493–14527 (2023).	1- Desarrollar una escala especializada para la evaluación de las habilidades PC de los profesores. 2- Medir los niveles de PC a los	Dos subestudios: 1) desarrollaron una medición de la capacidad de los docentes para integrar el PC en el currículo (Escala de Pensamiento Computacional para	Profesores STEAM Se recogieron 739 cuestionarios. Los docentes encuestados pertenecían a	Desarrolló una escala especializada para medir las habilidades de PC de los docentes (TCTS). Demostrado que la escala TCTS tiene una alta confiabilidad y validez y puede usarse como una herramienta

<p><a href="https://doi.org/10.1007/s10639-023-11777-7">https://doi.org/10.1007/s10639-023-11777-7</a></p>	<p>profesores STEAM de K-12 y analizar el impacto de los factores influyentes de los atributos personales, los atributos ocupacionales y el apoyo ambiental en las habilidades PC de los profesores.</p>	<p>Docentes, Computational Thinking Scale for Teachers -TCTS). 2) utilizaron la TCTS para medir el nivel de PC de los docentes STEAM, Adoptaron la prueba t de muestra independiente y el ANOVA unidireccional para explorar las diferencias en las habilidades de PC de los docentes STEAM en atributos personales, ocupacionales y contexto. Análisis de correlación y regresión lineal para verificar la relación entre varios factores influyentes y las habilidades de PC de los docentes STEAM.</p>	<p>246 escuelas primarias y secundarias de 30 localidades de China. Se recogieron un total de 1000 cuestionarios en línea de profesores de STEAM de K-12 de China, de los que se obtuvieron 925 cuestionarios válidos, luego de eliminar duplicados.</p>	<p>profesional y científica para medir las habilidades de PC de los docentes. Las habilidades de PC de los profesores de STEAM chinos están en un nivel medio-alto, y varios factores en los atributos personales, ocupacionales y de apoyo ambiental de los profesores STEAM desempeñaron un papel predictivo en sus habilidades de PC.</p>
<p>Villalustre Martínez, L. (2024). Análisis del nivel de pensamiento computacional de los futuros maestros: una propuesta diagnóstica para el diseño de acciones formativas. Pixel-Bit. Revista de Medios y Comunicación., 69, 169-194.  <a href="https://doi.org/10.12795/pixelbit.101205">https://doi.org/10.12795/pixelbit.101205</a>.</p>	<p>Medir el pensamiento computacional de los sujetos universitarios.</p>	<p>Se empleó el Test de Pensamiento Computacional (TPC), validado por Román-González (2015), constituido por 28 preguntas con cuatro opciones de respuestas, de las cuales sólo una de ellas es correcta. para analizar la existencia de diferencias en las cuatro dimensiones que componen el pensamiento computacional en función del género y la experiencia previa de los estudiantes en lenguajes de programación se realizaron pruebas t para muestras independientes.</p>	<p>164 estudiantes universitarios de los grados de maestro/a en educación infantil y primaria</p>	<p>Los resultados revelan que los hombres obtuvieron mejores resultados y que la experiencia previa en programación influyó en el nivel de desarrollo del pensamiento computacional. Además, se identificaron tres perfiles de estudiantes mediante un análisis de clúster. Las mujeres con experiencia previa en programación robótica y el uso de lenguajes de programación mostraron los mejores resultados en el cuestionario. Estos hallazgos resaltan la relevancia de realizar evaluaciones diagnósticas para conocer el nivel de competencia de los estudiantes en este ámbito, ya que puede ayudar a identificar áreas de mejora y adaptar las acciones formativas de acuerdo a las necesidades de cada grupo de estudiantes.</p>

Fuente: elaboración propia.

En los artículos anteriores, que sirven de muestra, pueden observarse el uso de distintas metodologías, que van desde las rúbricas desarrolladas para evaluar HPC, actividades Bebras prediseñadas, o test diseñados y evaluados para el caso en estudio. También puede observarse en esta selección que compartimos metodologías usadas en poblaciones variadas que van desde estudiantes de primaria a superior y docentes de todos los niveles y en distintos países.

Puede observarse además, los resultados de las investigaciones aspectos de género en relación al desarrollo de HPC, como por ejemplo, los obtenidos por Villalustre donde los hombres obtuvieron mejores resultados en los test de HPC y que la experiencia previa en programación influyó en el nivel de desarrollo del pensamiento computacional. Encontraron que además, las mujeres con experiencia previa en programación mostraron los mejores resultados en el cuestionario. Aspectos que son importantes al considerar cuando se trabaja con grupos de estudiantes que siempre son heterogéneos y donde cobra importancia la realización de actividades diagnósticas.

También se encuentra en el artículo de los investigadores de nacionalidad China, el desarrolló un test con una escala especializada para medir las habilidades de PC de docentes, que fue validado y aplicado para medir las HPC de un gran número de docentes de STEAM.

Otro punto importante que mencionan algunos trabajos es la necesidad de conocer las perspectivas de los estudiantes (ya sean niños, jóvenes, estudiantes de básica o docentes) sobre el aprendizaje de la programación, para adecuar y prever actitudes en los programas de formación.

### 3. Conclusiones

Los avances en la evaluación de habilidades de pensamiento computacional han sido significativos en los últimos años, reflejando una mayor comprensión y desarrollo de métodos para medir estas competencias esenciales. Estas habilidades son consideradas esenciales en la transformación de la educación moderna, ya que preparan a los estudiantes para enfrentar un mundo cada vez más impulsado por la tecnología y la innovación.

Este trabajo aporta una revisión y comparativa de métodos, herramientas, resultados de medición y evaluación de la construcción de habilidades de Pensamiento Computacional.

En el marco de las investigaciones realizadas por este equipo, relacionadas a la construcción de HPC de docentes, fundamentalmente de primaria, que han tomado una serie de formaciones propuestas en didáctica de las ciencias de la computación, surge la motivación para avanzar en una revisión sistemática. Los resultados obtenidos con estos docentes, se estudian en base a sus producciones, la observación de la modificación de sus prácticas docentes y el estudio del impacto en la construcción de HPC, estas son algunas premisas en las que avanzan los intereses investigativos del equipo.

Este trabajo presentó una revisión acotada de artículos significativos y recientes que abordan y proponen diferentes métodos e instrumentos para la evaluación de Habilidades de Pensamiento Computacional (HPC). Sobre seis artículos seleccionados se analizaron los objetivos, la población, los métodos propuestos e instrumentos utilizados, y los principales resultados obtenidos con la evaluación.

En esta revisión se detecta que la literatura en la temática exhibe una proliferación de propuestas, métodos, instrumentos y criterios de evaluación que dificulta encontrar formas establecidas y realizar un análisis más acabado del campo de investigación aquí abordado. Se observa la necesidad de avanzar en estudios y análisis comparativos de los métodos de evaluación de HPC, usados y propuestos, a fin de extraer y comparar características fundamentales que contribuyan a la definición de estándares para la medición y evaluación de estas habilidades claves para resolver problemas de manera estructurada y eficiente en diversos campos.

Por último, se detecta la necesidad de profundizar y refinar los estudios en el campo de la construcción y evaluación de HPC, en pos de avanzar hacia la obtención de métodos estandarizados que posibiliten la evaluación de su impacto en educadores y estudiantes de todos los niveles educativos, con metodologías aplicables en el ámbito nacional y latinoamericano.

### Referencias

- [1] Papert, S. *Mindstorms: Children, computers, and powerful ideas*. Basic Books Inc. 1980. <http://www.arvindguptatoys.com/arvindgupta/mindstorms.pdf>
- [2] Wing, J. Computational thinking. CACM Viewpoint. 2006. <https://doi.org/10.1145/1118178.1118215>
- [3] Catlin, D., y Woollard, J. Educational robots and computational thinking. En and others (Ed.), 4th TRtWR & RIE 2014. International workshop: Teaching robotics & teaching with robotics. 2014.
- [4] Wing, J. Computational thinking and thinking about computing. *Philosophical Transactions Of The Royal Society*, 2008, 366, 3717–3725. <https://doi.org/10.1098/rsta.2008.0118>
- [5] Wing, J. (2011). Computational thinking: What and Why. <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>
- [6] Brennan, K., y Resnick, M. Entrevistas basadas en artefactos para estudiar el desarrollo del Pensamiento Computacional en el diseño de medios interactivos. Vancouver, BC, Canada: American Educational Research Association. 2012.
- [7] Bers, M. U. Codings as a Playground: Programming and Computational Thinking in the Early Childhood Classroom. New York: Routledge. 2018. <https://doi.org/10.4324/9781315398945>
- [8] ISTE, y CSTA. (2011). Pensamiento computacional: caja de herramientas para líderes. Descargado de [http://eduteka.icesi.edu.co/pdfdir/PensamientoComputacional\\_Definicion.pdf](http://eduteka.icesi.edu.co/pdfdir/PensamientoComputacional_Definicion.pdf)

- [9] Csizmadia, A., Dorling, M., Ng, T. y Selby, C. Computational thinking-a guide for teachers. *Computing at School*, 2015.
- [10] Cuny, J., Snyder, L., y Wing, J. M. Demystifying computational thinking for non-computer scientists. Unpublished manuscript, 2010, referenced in <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>
- [11] The Royal Society (2012). Shut down or restart? The way forward for computing in UK schools. Computing in Schools Reports. <https://royalsociety.org/topics-policy/projects/computing-in-schools/report/>
- [12] Shute, V., Dun, C., y Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- [13] Codelearn. (noviembre 2019). ¿Qué es el pensamiento computacional? Codelearn. <https://codelearn.es/beneficios-del-pensamientocomputacional/>
- [14] Ortega, B. (2017). Pensamiento computacional y resolución de problemas. [Tesis doctoral], Universidad Autónoma de Madrid. Repositorio UAM. [https://repositorio.uam.es/bitstream/handle/10486/683810/ortega\\_ruperez\\_beatriz.pdf?sequence=1&isAllowed=y](https://repositorio.uam.es/bitstream/handle/10486/683810/ortega_ruperez_beatriz.pdf?sequence=1&isAllowed=y)
- [15] Sysło, M., y Kwiatkowska, A. Informatics for All High School Students: A Computational Thinking Approach. En I. Diethelm y R. Mittermeir (Eds.), *Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages*, 2013. (43-56). [https://doi.org/10.1007/978-3-642-36617-8\\_4](https://doi.org/10.1007/978-3-642-36617-8_4)
- [16] Zapata-Ros, M. (2019). Computational Thinking Unplugged. *Education in the Knowledge Society*, 20(18). [https://doi.org/10.14201/eks2019\\_20\\_a18](https://doi.org/10.14201/eks2019_20_a18)
- [17] García-Peñalvo, F. J. What computational thinking is. *Journal of Information Technology Research*, 9(3), v-viii, 2016. <https://repositorio.grial.eu/bitstream/grial/679/1/CT.pdf>
- [18] Daniele M., Quintero T., Bavera F. & Buffarini, F. (2019). Marcela Daniele, Teresa Quintero, Francisco Bavera, Flavia Buffarini. Análisis de producciones de docentes de educación primaria con formación en didáctica de las ciencias de la computación, Segundas Jornadas Argentinas de Didáctica de la Programación (JADIPRO). FAMAF y FFYH, Universidad Nacional de Córdoba. 7 y 8 de junio de 2019. Córdoba, Argentina.
- [19] Buffarini F., Rosso F., Bavera F. & Daniele M. (2019). La Entrada a los Procesos de Generalización: Una Clase de Problemas que Favorece su Construcción. Encuentro Regional de la Unión Matemática Argentina (ERUMA). FCEyT, Universidad Nacional de Santiago del Estero. 22, 23 y 24 de mayo de 2019.
- [20] Bavera F., Quintero T., Daniele M. & Buffarini, F. (2020). Computational Thinking Skills in Primary Teachers: Evaluation Using Bebras. In: Pesado P., Arroyo M. (eds) Computer Science – CACIC 2019. Communications in Computer and Information Science, 1184. Springer, Cham. DOI: [https://doi.org/10.1007/978-3-030-48325-8\\_26](https://doi.org/10.1007/978-3-030-48325-8_26)
- [21] Bavera F., Daniele M. & Quintero T. (2021) “Evaluación y Análisis de producciones de docentes de primaria formados en Pensamiento Computacional”. sesión Contenidos y didáctica, Jornadas Argentinas en Didáctica de las Ciencias de la Computación. Universidad Nacional de Quilmes, 4, 5 y 6 de noviembre de 2021. Buenos Aires. Argentina.
- [22] Bavera, F., Quintero, T. & Daniele, D. (2023). Formación en Ciencias de la Computación y su Impacto en las Habilidades del Pensamiento Computacional, 7º Encuentro de Investigación en Educación en Ciencias Naturales y Tecnología organizado en la Universidad de Buenos Aires, Argentina, desde el 5 al 8 de junio de 2023.
- [23] Kong, SC. (2019). Components and Methods of Evaluating Computational Thinking for Fostering Creative Problem-Solvers in Senior Primary School Education. In: Kong, SC., Abelson, H. (eds) Computational Thinking Education. Springer, Singapore.
- [24] Román-Gonzalez, et al. Test de Pensamiento Computacional: diseño y psicometría general.II Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2015). Octubre 14-16, 2015, Madrid, ESPAÑA
- I[25] A. D. F. Pérez and G. M. Valladares, Development and assessment of computational thinking: A methodological proposal and a support tool, 2018 IEEE Global Engineering Education Conference (EDUCON), Santa Cruz de Tenerife, Spain, 2018, pp. 787-795, doi: 10.1109/EDUCON.2018.8363311
- [26] C. Otero Avila, L. Foss, A. Bordini, M. Simone Debacco and S. A. da Costa Cavalheiro, "Evaluation Rubric for Computational Thinking Concepts," 2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT), Maceio, Brazil, 2019, pp. 279-281, doi: 10.1109/ICALT.2019.00089.

- [27] Sun, L., You, X. & Zhou, D. Evaluation and development of STEAM teachers' computational thinking skills: Analysis of multiple influential factors. *Educ Inf Technol* 28, 14493–14527 (2023). <https://doi.org/10.1007/s10639-023-11777-7>

Versión Preliminar

# **Experiencias Docentes**

Versión Preliminar

# **Uso de Herramientas de Software y Estrategias Didácticas para Facilitar la Comprensión de los Alumnos en la Enseñanza de Compiladores**

Gladis M. Sequeira<sup>1</sup>, Gabriela Del R. Gómez<sup>2</sup>, Cristian A. Kornuta<sup>3</sup>

<sup>1</sup> Facultad de Ciencias Exactas, Químicas y Naturales – Universidad Nacional de Misiones, Posadas, Argentina

<sup>2</sup> Facultad de Ciencias Exactas, Químicas y Naturales – Universidad Nacional de Misiones, Posadas, Argentina

<sup>3</sup> Facultad de Ciencias Exactas, Químicas y Naturales – Universidad Nacional de Misiones, Posadas, Argentina

[gladis.sequeira@gmail.com](mailto:gladis.sequeira@gmail.com), [gabriela.gomez@fceqvn.unam.edu.ar](mailto:gabriela.gomez@fceqvn.unam.edu.ar), [cristian.kornuta@yahoo.com.ar](mailto:cristian.kornuta@yahoo.com.ar)

## **Resumen**

En el presente trabajo, nos propusimos dar cuenta de la experiencia desarrollada con los estudiantes de la cátedra de Teoría de la Computación, perteneciente a las carreras Licenciatura en Sistemas de Información y Profesorado Universitario en Computación de la Facultad de Ciencias Exactas, Químicas y Naturales (FCEQyN) de la Universidad Nacional de Misiones (UNaM). Esta asignatura se encuentra en el primer cuatrimestre del cuarto año de ambas carreras. A partir del análisis de la propuesta, pretendimos mostrar resignificaciones que fue adoptando el proceso de construcción de un compilador durante el ciclo lectivo 2024, mediante la innovación con herramientas tecnológicas aplicadas en el aula.

Cabe destacar que las nuevas estrategias de enseñanza favorecieron a los estudiantes en el desarrollo de actitudes proactivas hacia logros de metas, mediante procesos cognitivos de complejidad creciente. Consideramos que el desarrollo de la experiencia permitió avanzar hacia una mejor comprensión y afianzamiento de los conocimientos en temas teóricos, así como obtener un prototipo funcional por parte de los alumnos al finalizar el cursado.

**Palabras clave:** Compiladores; Teoría de la computación; Estrategias didácticas; Educación universitaria; Computación.

## **1. Introducción**

La cátedra de Teoría de la Computación presenta un gran desafío para los docentes que nos encontramos trabajando de manera constante en la planificación y ejecución de procesos de enseñanza y de aprendizaje. La compleja tarea docente, causada por el grado de abstracción, requiere que se relacionen las unidades de la materia con contenidos aprendidos en cátedras anteriores.

A partir del año 2010, el equipo docente se fue conformando. El primer docente se abocó en su dictado a la enseñanza de los aspectos teóricos vinculados a los principios de diseño de compilación. Se organizó a los estudiantes en grupos cooperativos, con resolución de trabajos prácticos relacionados con cuestionarios teóricos y la creación de mapas conceptuales para abordarlos. En el año 2011 se incorporaron herramientas informáticas de simulación, con la finalidad de que los estudiantes realicen pruebas de sus ejercicios en herramientas de software específicas para este tema. Por su parte, en [1], se menciona: “es posible que la aplicación de refinamiento sucesivo facilite al sistema cognitivo del sujeto la tarea de anticipar el comportamiento del modelo y permita redirigir ese esfuerzo hacia la comprensión de la teoría que respalda la práctica en proceso”.

En dictados posteriores se incorporaron dos herramientas de software específicas para realizar las simulaciones. Por un lado, para trabajar con autómatas se utilizó JFLAP [2], mientras que, para realizar la comprobación del árbol sintáctico,

se llevaron adelante pruebas con ANTRLWorks [3] con la intención de lograr la comprensión de los estudiantes sobre estos temas centrales.

En otros años, se solicitó a los estudiantes como trabajo final, una propuesta de creación de un lenguaje propio a partir de los trabajos prácticos que se fueron desarrollando durante las clases. La actividad final consistió en el diseño de un lenguaje de programación con una finalidad específica elegida por los estudiantes. Por ejemplo: lenguajes para la enseñanza de la programación, lenguajes para personas con discapacidad visual, entre otras propuestas. En este trabajo final debían identificar los tokens, escribir las expresiones regulares, diagramar los autómatas, escribir la gramática de su lenguaje y una propuesta de la interfaz de ese lenguaje específico [4]. Cabe mencionar que, por prioridades de la cátedra, se enfocó en la asimilación de los conceptos dados, dejando de lado la codificación de los proyectos.

Dentro de la cátedra, los contenidos mínimos con los que se trabaja son: sintaxis y semántica, semántica operacional, lenguajes formales y autónomas, minimización de autómatas, expresiones regulares, gramática e isomorfismo. Cabe destacar que, en la Licenciatura en Sistemas de Información, este espacio es la única asignatura donde se aborda compiladores, ya que no contamos con un espacio específico para desarrollar estos temas.

Desde el ciclo lectivo 2023 hasta la actualidad, se implementaron otras estrategias didácticas para el abordaje del trabajo final. La cátedra proporcionó el fragmento de código de un programa en un lenguaje ficticio o real, con el que los grupos de alumnos trabajaron en la implementación de su propio compilador, utilizando librerías de metacompiladores y herramientas de software diversas. En dicho año se utilizó el metacompilador PLY [5], y en el ciclo 2024 las herramientas de metacompiladores elegidas fueron JFlex y CUP [6].

## 2. Compiladores en Teoría de la Computación

Esta sección tiene la finalidad de explicar el trabajo con compiladores en la asignatura Teoría de la Computación, espacio compartido entre las carreras Profesorado Universitario en Computación y Licenciatura en Sistemas de Información. La asignatura se lleva adelante en el primer cuatrimestre del cuarto año en ambas carreras.

Dentro del plan de estudios de la licenciatura no se presentan otros espacios destinados al trabajo con compiladores; por tal motivo, y en concordancia con los contenidos mínimos del programa, en el dictado se abordan los temas relacionados con compiladores.

El conocimiento sobre técnicas de compilación desempeña un papel crucial para los estudiantes que participan en el diseño y desarrollo de software, ya que les permite maximizar el uso de los compiladores que emplean y entender, de esta manera, sus limitaciones. Además, el estudio de la teoría detrás de la generación de construcciones léxicas y gramaticales mejora la comprensión de los lenguajes de programación utilizados, facilitando un uso más eficiente.

La asignatura tiene como objetivo que los estudiantes comprendan cómo se pueden definir matemáticamente los lenguajes naturales (como lenguajes formales) para facilitar su entendimiento, destacando las similitudes y diferencias entre ambas representaciones.

También busca aplicar los conocimientos sobre gramáticas formales en la descripción de lenguajes de programación, lo que permite a los estudiantes desarrollar partes de un compilador o entender su funcionamiento. Asimismo, la asignatura proporciona las bases necesarias para diseñar analizadores léxicos y comprender la amplia variedad de aplicaciones que pueden ser representadas y simuladas con este tipo de módulos.

Se pretende establecer fundamentos para el diseño de analizadores sintácticos dentro de un compilador. Finalmente, se busca familiarizar a los estudiantes con el poder computacional de estas máquinas en relación con la resolución de problemas relacionados con el reconocimiento de lenguajes.

Dentro de la cátedra, los contenidos mínimos relacionados a la experiencia llevada adelante con los estudiantes son los siguientes: Trabajo con sintaxis y semántica, expresiones regulares, fases de un compilador, analizador léxico, errores léxicos, construcción de un analizador léxico, generación automática de analizadores léxicos para determinados lenguajes, analizador sintáctico y manejo de errores.

## 3. Implementación de Simuladores y Metacompiladores

Desde el equipo de la cátedra, se buscó implementar el enfoque constructivista [7] como estrategia didáctica para mejorar la comprensión de los alumnos y apoyar el proceso de enseñanza y de aprendizaje dentro de la asignatura Teoría de la Computación. Según este enfoque, el conocimiento se construye a partir de los esquemas cognitivos previos de cada individuo. Este proceso se desarrolló gradualmente, haciendo énfasis en el aprendizaje personal, permitiendo a los estudiantes adquirir no solo nuevos conocimientos e incorporación de nuevas herramientas de software, sino también competencias que podrán aplicar en situaciones futuras.

El trabajo se enfocó en el desarrollo de un integrador como proyecto final grupal para promocionar la cátedra. La tarea consistió en la implementación de un analizador léxico y un analizador sintáctico para la compilación de una propuesta, donde se trabajó con algoritmos en un lenguaje ficticio, similar al lenguaje C, pero en castellano. El propósito del código fue hallar el factorial de un número ingresado por el usuario a través de un programa principal que, a su vez, llama a una función para tal fin. La estructura del lenguaje contempló las siguientes partes: las secciones de declaraciones de variables, la incorporación de funciones y la sección del programa principal.

Las librerías con las que trabajaron los estudiantes fueron JFlex y CUP en el lenguaje JAVA [6]. Los objetivos de estas actividades realizadas pretendían que los estudiantes de manera grupal, conocieran teóricamente a un compilador y pudieran crear de manera práctica, un analizador léxico y un analizador sintáctico a partir de estos conocimientos obtenidos. Los metacompiladores ofrecieron a los estudiantes la oportunidad de aplicar los conocimientos adquiridos durante el curso, facilitando la implementación de dos fases claves: el analizador léxico y el analizador sintáctico [8].

Para poder llegar a esta instancia, nuestra experiencia nos llevó a utilizar durante la primera parte del cursado herramientas de software como regexpr [9] y otras propuestas de los alumnos para la asimilación de los conceptos de expresiones regulares. En segunda instancia, para los autómatas finitos se utilizó la herramienta JFlap [2]; de la misma forma se abordó para la gramática.

A partir de ello, y conjuntamente con una guía de desarrollo compuesta por tres etapas, los grupos de estudiantes elaboraron su trabajo Integrador final, que consistió en la codificación de dos fases de un compilador: el analizador léxico y sintáctico, además del diseño de una interfaz de pantalla para el manejo del usuario.

La realización del mismo pretendió que se implementarán las primeras fases del trabajo final utilizando la misma herramienta como parte de la documentación del trabajo. Se trabajó en forma grupal considerando que la construcción del conocimiento, según el enfoque utilizado, se produce cuando el alumno interactúa con otros [7]. La consigna incluyó la presentación del producto obtenido.

### 3.1 Implementación de un analizador léxico y un analizador sintáctico

Las actividades que llevaron a cabo los grupos de estudiantes fueron las siguientes:

- Análisis Léxico: Para realizar el análisis léxico del lenguaje de programación específico se optó por utilizar la herramienta JFlex. Esta herramienta es un generador de analizadores léxicos para Java que proporciona diversas características como, la definición de patrones léxicos que permiten definir expresiones regulares que representan los diferentes tokens del lenguaje, la generación automática de código a partir de las definiciones léxicas y la integración con herramientas de análisis sintáctico; en este caso, la herramienta de integración fue CUP.

Para esto, se creó un archivo de reglas léxicas que contiene todas las reglas que el analizador seguirá. Este archivo define las expresiones regulares que corresponden a cada uno de los tokens, del lenguaje. Seguidamente, se creó el archivo de tokens identificando cada una de las reglas léxicas definidas. Por último, se generó el archivo main, que se utilizó para generar el archivo de reglas léxicas a través de la librería de JFlex.

- Una vez generados los archivos necesarios para el análisis léxico, se procedió con la creación del analizador sintáctico. Para este propósito, se utilizó la herramienta CUP (Constructor of Useful Parsers) que es un generador de analizadores sintácticos para Java. El primer paso fue generar el archivo de reglas sintácticas que define las reglas léxicas específicas para el analizador sintáctico, asegurando que el código fuente sea analizado correctamente. A continuación, se creó el archivo de gramática, el cual define todas las reglas gramaticales necesarias para analizar el código. Luego, se adaptó el archivo al main para generar los archivos nuevos.
- Finalmente, se creó la interfaz gráfica en java para mostrar las diferentes opciones de análisis. Esta pantalla incluyó la opción de cargar un archivo desde el ordenador que debe estar en formato texto, o copiar un código para luego pegarlo en la entrada de datos.

Algunos ejemplos de interfaz gráfica se muestran a continuación:

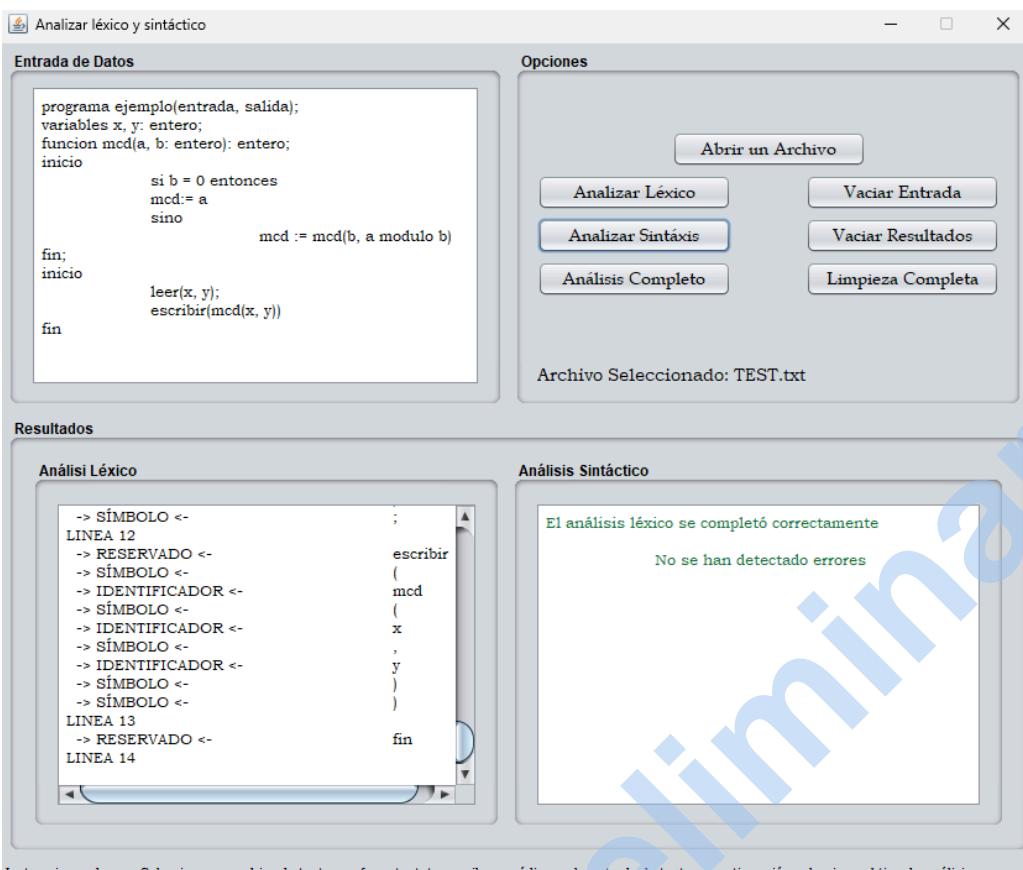


Figura 1. Interfaz gráfica del analizador léxico y analizador sintáctico.

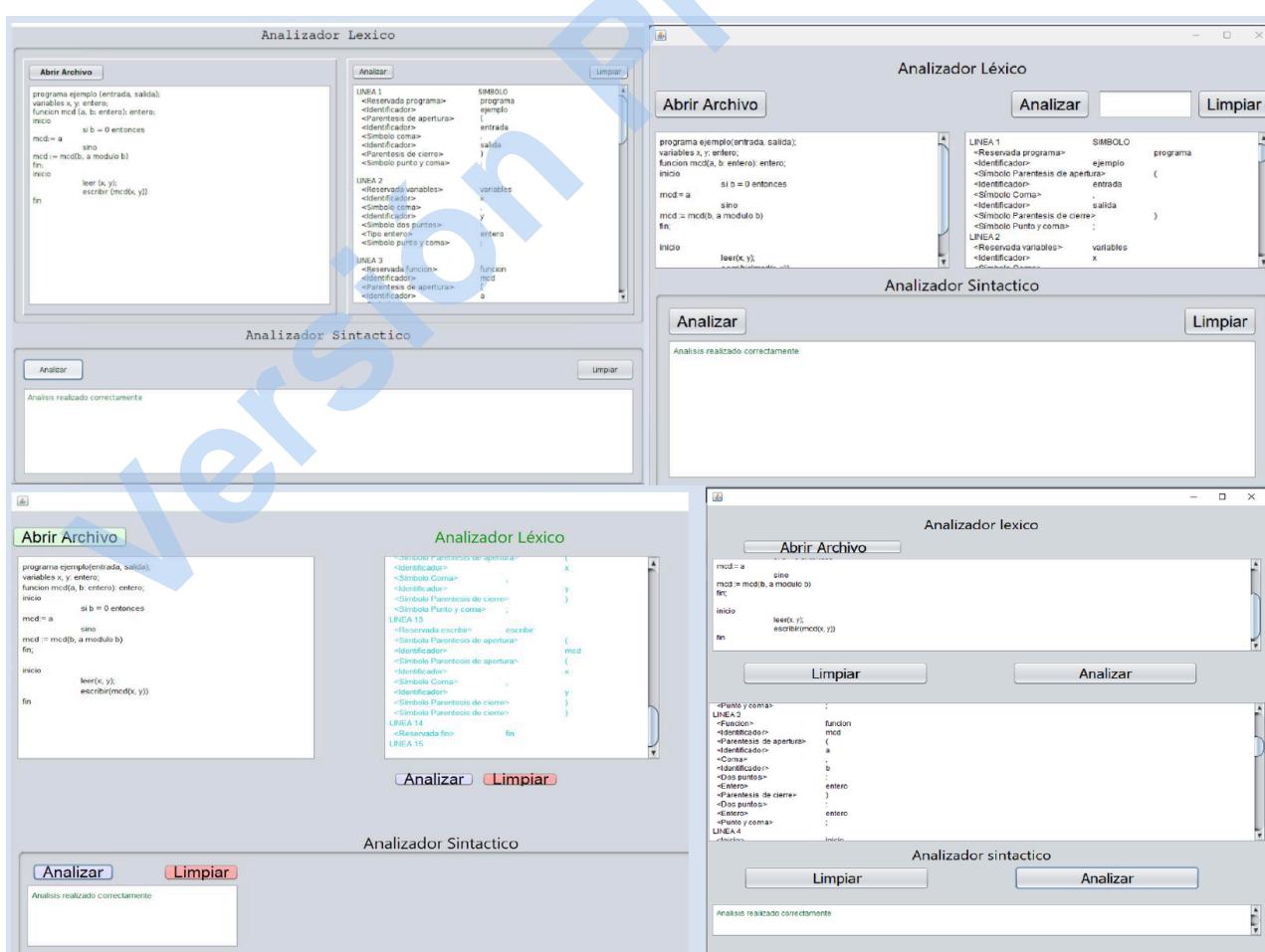


Figura 2. Interfaz gráfica del analizador léxico y analizador sintáctico de algunos grupos de trabajo - Ciclo 2024.

### **3.2. Resultados y Objetivos**

Para obtener los resultados, al finalizar la entrega y defensa de los trabajos integradores grupales, se realizó una encuesta anónima para evaluar la percepción de los estudiantes sobre la utilización de herramientas de software en las simulaciones y la creación de los analizadores léxico y sintáctico. Los resultados revelaron que el 90% de los alumnos, de un total de 18 participantes, consideraron que las prácticas con software fueron fundamentales para consolidar su comprensión de los conceptos teóricos de la asignatura.

Los siete grupos de estudiantes que completaron el curso durante el primer cuatrimestre de 2024 lograron desarrollar, con variados niveles de complejidad, el prototipo requerido para enriquecer el aprendizaje en la asignatura. Este logro subrayó el impacto positivo de la aplicación práctica de los conocimientos teóricos adquiridos en el curso, destacando el compromiso y la capacidad de los estudiantes para integrar la teoría con la práctica en el contexto educativo.

Es importante destacar los aspectos clave que se trabajaron con los estudiantes, relacionados con el desarrollo de habilidades profesionales y el entendimiento del funcionamiento del software y los sistemas informáticos. Se comenzó con la mejora de la capacidad para la resolución de problemas complejos a partir de la traducción de lenguajes, el manejo de errores y la generación de código eficiente. Esto también les permitió obtener un conocimiento profundo en la construcción de compiladores, posibilitando a los estudiantes participar en el diseño de nuevos lenguajes de programación o en la mejora de los actuales.

## **4. Conclusiones**

La incorporación de herramientas de software en la cátedra Teoría de la Computación resultó beneficiosa para ambas partes del proceso. Por un lado, los docentes del equipo de cátedra observamos actividades productivas, tanto grupales como individuales, por parte de los estudiantes; mientras que los alumnos demostraron mayor destreza y confianza al utilizar las herramientas de software de simulación y de programación para crear su propio prototipo de lenguaje, centrándose en este caso, en la codificación de las fases de analizador léxico y sintáctico de un compilador.

Al ser la primera vez que se implementó un trabajo final integrador a partir de las librerías JFlex y CUP con estas características en la cátedra, y al obtener buenos resultados, como líneas futuras esperamos aplicar mejoras en las fases implementadas y, además, incorporar la codificación de otra fase en ciclos lectivos siguientes. El equipo de cátedra reconoció el progreso en la comprensión de los contenidos y el interés en las actividades, lo cual se reflejó en la calidad y originalidad de los prototipos diseñados.

Una encuesta realizada después del trabajo integrador reveló que, para el 90% de los estudiantes, la experiencia fue un desafío motivador que les impulsó a seguir investigando y les permitió integrar los conceptos teóricos de la cátedra.

Por último, queremos destacar que conocer el funcionamiento de los compiladores permitió a los estudiantes tener una perspectiva más amplia del proceso completo, desde la creación de un programa en un lenguaje de alto nivel hasta su ejecución en hardware, elementos esenciales en el ciclo de desarrollo del software.

## **Referencias**

- [1] G. M. Sequeira, S. D. Zajaczkowski, C. A. Kornuta y G. R. Gómez, “Estrategias Didácticas en el uso de Herramientas Software para favorecer la comprensión de los alumnos en la Enseñanza sobre Compiladores”, In *XV Workshop de Investigadores en Ciencias de la Computación*, pp. 285-288.
- [2] S. H. Rodger (2005). JFLAP [Online]. Available: <https://www.jflap.org/>
- [3] T. Parr (1989, Feb. 03). ANTLRWorks [Online]. Available: <https://www.antlr3.org/works/>
- [4] Kenneth, L.: Construcción de compiladores, principios y práctica. International Thomson Ed. (2004)
- [5] PLY (Python Lex-Yacc), <http://www.dabeaz.com/ply/ply.htm>
- [6] Ruiz Catalan, J.: Compiladores TEORÍA E IMPLEMENTACIÓN. International Thomson Ed. (2010)
- [7] M. Carretero (1993). Constructivismo y Educación. Aique
- [8] Aho, A., Lam, M., Sethi, R., Ullman, J., Compilers, Principles, Techniques & Tools. Pearson Addison – Wesley
- [9] G. Skinner (2023, Feb. 03). RegEx [Online]. Available: <https://regexpr.com/>

# El desarrollo de habilidades de programación en el tramo superior de formación de carreras Informáticas en la era de ChatGPT

Paola D. Budán<sup>1,2</sup>, Margarita Álvarez<sup>1,2</sup>

<sup>1</sup>Universidad Nacional de Santiago del Estero, Facultad de Ciencias Exactas y Tecnologías, Argentina

<sup>2</sup>IISII Instituto de Investigación en Informática y Sistemas de Información, UNSE, Argentina

[pbudan@unse.edu.ar](mailto:pbudan@unse.edu.ar), [alvarez@unse.edu.ar](mailto:alvarez@unse.edu.ar)

## Resumen

La tecnología basada en Inteligencia Artificial interpela las habilidades de los docentes para incentivar prácticas de programación, especialmente en el tramo superior de formación de carreras insertas en la rama de la Informática. En este artículo se difunden las estrategias utilizadas en la asignatura Lenguajes de Programación y Compiladores, que se dicta en la Universidad Nacional de Santiago del Estero, en el cuarto año de la carrera Licenciatura en Sistemas de Información. El principal desafío en esta etapa formativa consiste diseñar actividades que propicien la motivación del estudiante en las cuales la tecnología se pueda usar para disparar el pensamiento creativo. Ello requiere que, partiendo de consignas dirigidas a objetivos claros, se fomente en los alumnos la habilidad de argumentar críticamente las decisiones tomadas para el desarrollo del trabajo que se les encarga. A su vez, requiere que el docente plantee criterios de evaluación diferentes a los habituales. A tales fines se han desarrollado experimentalmente actividades en la asignatura Lenguajes de Programación y Compiladores en las que no se restringe el uso de herramientas tecnológicas de ninguna naturaleza, obteniéndose a la fecha resultados favorables y que incentivan nuevas propuestas para el año académico siguiente.

**Palabras clave:** Informática; Argumentación crítica; Habilidades de programación.

## 1. Introducción

En este artículo, presentaremos los resultados obtenidos tras implementar actividades mediadas por tecnologías que, al mismo tiempo, fomentan el pensamiento crítico y la creatividad de alumnos avanzados (cuarto año) en la carrera de Licenciatura en Sistemas de Información, que se ofrece en la Universidad Nacional de Santiago del Estero, Argentina. En general, en estos tramos la cursada recibe pocos alumnos, lo que hace posible transformar el proceso de enseñanza en un proceso de seguimiento exhaustivo y personalizado. Tal como se apreciará en la siguiente sección, la asignatura a la que haremos referencia es *Lenguajes de Programación y Compiladores*, y en ella se introduce una rama de la Informática muy específica, en la que un futuro profesional podría iniciar su formación de posgrado, pero que en los libros de textos existentes siempre se presenta de forma monótona como una seguidilla de algoritmos mecánicos con poco margen de creatividad. Dado que en este tipo de asignaturas los alumnos combinan resolución de ejercicios en papel con actividades de programación, es usual, por ejemplo, que los alumnos le soliciten a ChatGPT un código que solucione un determinado problema, un algoritmo que implemente alguna de las etapas de un compilador, entre otros requerimientos. Entonces, el problema que se genera para el docente es tratar de medir el grado de aprendizaje del alumno con las actividades que se le solicitan, o dicho de otro modo, cómo lograr que los alumnos aprovechen las asignaturas para lograr una formación integral. Responder estas preguntas conllevan a cuestionarse sobre el diseño de los recursos pedagógicos y de los recursos didácticos que se utilizan en las asignaturas del nivel superior de este tipo de carreras.

Lo que se difunde en esta ocasión corresponde a una investigación de tipo descriptiva, en la que principalmente se presentan los resultados de las prácticas diseñadas y llevadas a cabo en el período académico 2023 (2do cuatrimestre). La valoración de estos resultados nos permitirá mejorar las prácticas para el año académico en curso, ayudando en la motivación y el interés por el aprendizaje de esta rama tan específica de la Computación, pero además, ampliando los criterios de evaluación de la asignatura a la luz de las tecnologías existentes.

El aporte de esta experiencia se fundamenta en que es sabido de la existencia de numerosos artículos que relatan la incorporación de chatbots, como ChatGPT<sup>1</sup> en la educación general, y también en la universitaria [1] al [7]. Sin embargo, no se encuentran en la literatura explorada hasta el momento, alguna que relate una experiencia con una asignatura específica de los últimos tramos de formación de una carrera.

Estructuramos el trabajo de la siguiente manera: en primera instancia se describe la experiencia, centrándonos en la propuesta didáctica, situándonos en el contexto de la asignatura. Por ello, para dimensionar los resultados obtenidos, primero presentaremos la asignatura en la que se circunscribe la experiencia, y detallaremos cómo la misma fue llevada a cabo. Por último, presentaremos un análisis de los resultados que se van obteniendo en la formación práctica de esta asignatura y los desafíos prácticos que enfrentaremos durante el presente año académico en el cuatrimestre que recién inicia a modo de conclusión.

## 2. Descripción de la Experiencia

Durante el cursado de la asignatura los alumnos trabajan con trabajos prácticos y un taller o trabajo de integración para desarrollar un intérprete. Los trabajos prácticos poseen ejercicios clásicos de los diferentes temas que se abordan en la teoría, y otros ejercicios en donde el alumno tiene que desarrollar habilidades de programación y trabajo en equipo, que se complementan con lecturas de investigaciones actualizadas sobre el tema bajo estudio. Los trabajos prácticos están diseñados de manera de contribuir con la resolución de problemas específicos de la informática, en el dominio particular de las *herramientas de análisis lexicográfico y sintáctico* [8], a la vez que se complementan con la realización de proyectos plasmadas en actividades en las cuales los alumnos deben programar. Ejemplo de actividades previstas en estos trabajos prácticos son: el desarrollo manual de analizadores léxicos, y luego el desarrollo con alguna herramienta automática como ANTLR<sup>2</sup>, lo que le permite al alumno profundizar aspectos vitales para la práctica como, por ejemplo, el orden en el cual se consignan las expresiones regulares. Este tipo de trabajo se complementa con lectura de investigaciones en las cuales se toma contacto con cómo es usado un analizador léxico en la actualidad, en diferentes dominios prácticos. A continuación, detallaremos la propuesta didáctica del desarrollo del taller.

### 2.1. Propuesta Didáctica

Como trabajo de integración, los alumnos desarrollan el taller denominado *Diseño y construcción de un intérprete*, que se entiende como una actividad de resolución de problemas del mundo real. Los objetivos de este taller son: (i) introducir al alumno en los problemas concretos que se enfrentan durante el desarrollo de programas traductores; (ii) reforzar, desde el punto de vista práctico, los conocimientos adquiridos sobre analizadores lexicográficos, sintácticos y estructura de programas traductores; y (iii) adquirir habilidades para construir compiladores/intérpretes sencillos. Para conseguirlo, los alumnos deben seguir una metodología propuesta por la cátedra, que básicamente plantea: (i) repasar los temas que comprende el trabajo, (ii) cuestiones técnicas puntuales como lo son la especificación de un vocabulario, la elaboración y prueba manual de las gramáticas necesarias para el funcionamiento del intérprete, entre otras actividades, (iii) la codificación del intérprete con generadores automáticos como el ANTLR, (iv) la prueba funcional de lo generado , y por último (v) la documentación completa del trabajo realizado. Durante todo el desarrollo del taller los alumnos pueden utilizar ChatGPT para solicitarle *ayuda* en cuanto al código que deben generar.

A modo de ejemplo, en los últimos años, el trabajo integrador solicitado fue:

- Año 2023: Diseño y desarrollo grupal de un analizador léxico utilizando un lenguaje de programación a elección del grupo de alumnos.
- Año 2022: Diseño y desarrollo grupal de un intérprete de instrucciones en Python.
- Años 2021 y 2020: Diseño y desarrollo grupal de un intérprete de un mini lenguaje a elección del grupo de alumnos.

Una vez que los trabajos prácticos y el enunciado del trabajo integrador se ponen a disposición de los alumnos, comienza la etapa de evaluar los procedimientos mediante los cuales los alumnos van desarrollando sus habilidades en el marco de la asignatura, y si es necesario realizar ajustes al respecto. En este contexto, los responsables de la asignatura elaboran los criterios de evaluación, que principalmente contemplan el pensamiento crítico y la creatividad puesta en práctica por los alumnos, y su capacidad argumentativa en el momento de defender el trabajo. En el caso del informe escrito del trabajo integrador, se evalúa:

- La organización de la información y pertinencia de los temas incluidos, fundamentando en cada caso el porqué se incluye cada tema.

<sup>1</sup> En este artículo, utilizaremos indistintamente chatbot o ChatGPT, aunque el primer término no agota su significado en el segundo.

<sup>2</sup> <https://www.antlr.org/>

- La correcta redacción y uso adecuado del estilo de escritura que facilite la comprensión, incorporando apreciaciones y valoraciones propias sobre la utilidad en la formación de los temas abordados.
- La utilización de ejemplos acordes al tema en cuestión.
- La inclusión de fuentes de información o referencias acorde a las normas APA. Esto es de vital importancia puesto que algunas herramientas como ChatGPT suelen alucinar al momento de indicar las fuentes de las cuales toman la información.

En cuanto al funcionamiento del intérprete, los criterios de evaluación fijados son:

- La adecuación de las herramientas utilizadas (para programar y para consultar).
- Completitud de las operaciones exigidas.
- Correcto funcionamiento del intérprete.
- Detección de errores y emisión de los mensajes correspondientes.
- Facilidad de uso. Diseños de pantalla adecuados y recursos gráficos convenientes.

Cada trabajo integrador se evalúa en un coloquio oral, que es una de las instancias en las que mejor se aprecia la criticidad de los alumnos para argumentar el porqué solucionan los problemas de una determinada manera *y no de otra*. En este sentido, los criterios de evaluación son:

- Solvencia para responder las preguntas, utilizando argumentos técnicos y vocabulario específico de la asignatura.
- Respuesta clara y precisa a los comentarios que se realicen sobre el funcionamiento del programa.
- Claridad para presentar las alternativas de solución planteadas durante el desarrollo del trabajo, y qué factores se consideraron en el momento de evaluar las diferentes alternativas de solución a un problema que se les presentara durante el desarrollo. Por ejemplo, los factores que consideraron en el momento de decidir el lenguaje de programación a utilizar para alcanzar lo solicitado en el Proyecto, o qué tan adecuadas fueron las respuestas a las consultas realizadas a la herramienta de Inteligencia Artificial (IA).

Finalmente, se les pide a los alumnos que presenten una evaluación sobre la utilidad de herramientas basadas en IA en la programación de intérpretes, comentando las ventajas y desventajas detectadas a partir de su experiencia.

## 2.2. Contexto

*Lenguajes de Programación y Compiladores* es una asignatura que se ubica en el trayecto de Ciencias Básicas Generales y Específicas, de la currícula de la *Licenciatura en Sistemas de Información*. Aborda conceptos de los lenguajes de programación y en ella se discuten aspectos referidos al diseño y a la implementación de estos, de modo tal que le permita al estudiante aprovechar su potencial y permitir lograr un uso eficiente de los lenguajes de programación. Otro gran tópico que trata la asignatura es la aplicación de las gramáticas regulares y libres de contexto y, de los autómatas finitos y de pila en los compiladores e intérpretes<sup>3</sup>. Se proporciona la justificación teórica y práctica de los instrumentos necesarios para la construcción de los compiladores. Los conceptos y técnicas correspondientes al tema Compiladores le permiten al alumno desarrollar un intérprete sencillo y además, pueden ser utilizadas en el desarrollo de software de aplicación general.

Los objetivos de esta asignatura son:

- Conocer los diferentes conceptos de los lenguajes de programación.
- Diferenciar los criterios de diseño y de implementación de lenguajes de programación para ser capaces de justificar una elección.
- Construir compiladores/intérpretes sencillos mediante el desarrollo de las fases del proceso de traducción y con el empleo de herramientas generadoras de analizadores léxicos y sintácticos.
- Aplicar procedimientos que permitan la generación de código intermedio y su optimización.

El valor formativo de todos los conceptos desarrollados en la asignatura es esencial en el diseño de Compiladores e Intérpretes y además, le proporciona al alumno una base sólida para el ejercicio profesional en el desarrollo de programas. Al mismo tiempo, complementa su formación informática en un área especializada y que posteriormente puede ser profundizada con posgrados específicos. Al estar ubicada en los tramos finales de la carrera, y luego de que los alumnos obtengan el título intermedio de Analista en Sistemas de Información, es una asignatura con muy baja matrícula, en promedio 8 alumnos por año. Sin embargo, el hecho de que sea pocos alumnos permite que se lleve a cabo un seguimiento personalizado del progreso de cada uno de ellos, tanto en lo que respecta al entendimiento de los conceptos de la asignatura, como al desarrollo de los trabajos prácticos.

<sup>3</sup> Aún cuando un intérprete es técnicamente diferente a un compilador, en este artículo se los usará como sinónimos para no confundir al lector con algunos tecnicismos que agregan complejidad a la lectura

## 2.3. La Experiencia

El desarrollo del taller consistió en el diseño y desarrollo grupal de un *analizador léxico* utilizando un lenguaje de programación a elección del grupo de alumnos y las herramientas basadas en IA que tengan a disposición (como por ejemplo, ChatGPT). Un analizador léxico es la primera etapa de un compilador y se encarga de escanear un flujo de caracteres (código fuente) para identificar y categorizar los tokens individuales, como palabras clave, identificadores, operadores y símbolos. Es por ello que entender su funcionamiento en profundidad completa las habilidades de esta asignatura.

### Consignas:

- Conformar un grupo de trabajo de, a lo sumo, 3 integrantes, acorde a las siguientes pautas:
  - Pauta 1: El equipo debe tener un nombre. En la carátula del informe final se debe incluir tanto el nombre del equipo como sus integrantes.
  - Pauta 2: Deben especificar acuerdos iniciales de trabajo, por ejemplo, las formas en las que se reunirán, cada cuánto lo harán, entre otros acuerdos.
- Realizar la planificación: identificar actividades y recursos de desarrollo, y definir la calendarización de éstas para dar cumplimiento a la fecha de entrega del taller. Consignar los roles de cada integrante. En esta planificación se deberían incluir el estudio/repaso de los conceptos necesarios para llevar a cabo el taller, el desarrollo de las pruebas en cada etapa del analizador, a modo de ejemplo de actividades a considerar. Es deseable que tengan en cuenta herramientas que se basan en IA, como ChatGPT, para el desarrollo del taller.
- Luego de analizar detalladamente la metodología a aplicar, solicitar a los docentes información adicional y/o consultas.
- Respetar todas y cada una de las actividades consignadas en la metodología.

### Metodología:

#### a) Investigación Preliminar sobre:

- a.1. La estructura de un analizador léxico y cómo funciona. Los conceptos de tokens, expresiones regulares y el proceso de escaneo del código fuente
- a.2. Generadores automáticos de Analizadores Léxicos: ANTLR, Flex, Ragel, Coco/R, entre otros.
- a.3. Correcta documentación de la investigación, incluyendo referencias.

b) Descripción del lenguaje y definición de tokens: Describir el lenguaje de programación para el cual generarán el AL, definir los tipos de tokens que el analizador deberá reconocer, como identificadores, números, operadores, paréntesis, etc. En el caso de acceso a sitios web, consignar día de visita al sitio. En el caso de considerar utilizar ChatGPT para esta tarea, especificar los *prompts* utilizados y la calidad de los resultados obtenidos.

c) Especificación de las Expresiones Regulares para cada tipo de token definido. Estas expresiones regulares serán patrones que el analizador usará para identificar los tokens en el código fuente. Pueden ajustarse a la notación utilizada por un lenguaje de programación o pueden definir una notación propia.

d) Generación y prueba de la gramática regular. Definir la gramática en forma correcta y probarla con patrones válidos e inválidos. En el caso de considerar utilizar ChatGPT para esta tarea, especificar los *prompts* utilizados y la calidad de los resultados obtenidos.

e) Implementación del analizador léxico: Programar el analizador léxico utilizando las expresiones regulares definidas. Recordar que: (i) usualmente los AL se implementan con AFD, (ii) el analizador debe ser capaz de leer un archivo de código fuente, escanearlo carácter por carácter y producir una secuencia de tokens identificados. En el caso de considerar utilizar ChatGPT para esta tarea, especificar los *prompts* utilizados y la calidad de los resultados obtenidos.

f) Realización y documentación de pruebas: Crear varios archivos de prueba con fragmentos de código en el lenguaje seleccionado por el grupo. Ejecutar el AL con esos archivos verificando que identifique correctamente los tokens esperados.

- e) Gestión de errores: Implementar mecanismos para manejar errores léxicos, como caracteres desconocidos o patrones no reconocidos.
- h) Manejo de la Tabla de Símbolos: Esto incluye: (i) definir la tabla de símbolos, agregando una estructura de datos (como un diccionario o una lista) para almacenar la información de los identificadores encontrados en el código fuente. Cada entrada podría contener el nombre del identificador, su tipo y cualquier otra información relevante; (ii) actualizar la tabla de símbolos cada vez que se encuentre un nuevo identificador en el código fuente; (iii) verificar que el identificador no exista en la tabla.
- i) Manejo de Buffers:
  - o i.1. Implementación de Buffer de Entrada: En lugar de leer el código fuente carácter por carácter directamente desde el archivo, implementar un buffer de entrada. Esto implica leer una cantidad significativa de caracteres a la vez en el buffer para reducir la cantidad de operaciones de lectura.

- i.2. Avance y Retroceso del Buffer: A medida que el analizador léxico avanza a través del código fuente, avanzará y retrocederá en el buffer según la cantidad de caracteres procesados. Esto puede mejorar la eficiencia al reducir las operaciones de lectura del archivo.
- i.3. Manejo de Fin de Archivo: Implementar un mecanismo para detectar y manejar el fin de archivo (EOF) en el buffer. Cuando el buffer se vacía, cargar nuevos caracteres del archivo en el buffer.

#### **Productos esperados:**

El trabajo que realizará cada grupo de alumnos tendrá como producto un Generador de Análisis Léxico con la documentación solicitada. Para ello, se prevé la entrega de Informes de avances que contemplan las etapas de la metodología y la documentación solicitada en otros puntos, de la siguiente manera:

- Primer Informe: Tareas de documentación: a) hasta d). Fecha: 19-septiembre.
- Segundo Informe: Tareas de programación: e) hasta h). Fecha: 3-octubre.
- Tercer Informe: Tareas i) de programación- Fecha: 17-octubre.
- Informe Final: Documentación integradora de todas las etapas, incluyendo una comparación entre las implementaciones realizadas en las tareas e) e i). Fecha: 31-octubre.

#### **Presentación del Trabajo**

Para el Informe Final se requiere la presentación del trabajo por escrito y el programa.

**Informe escrito:** que contenga la documentación de cada una de las etapas de la metodología aplicada y de la documentación requerida. Debe contener:

- a. Un índice.
- b. El estudio exploratorio de la primera actividad.
- c. La descripción del lenguaje.
- d. La documentación del analizador léxico, con sus respectivas pruebas.
- e. La comparación de las implementaciones presentadas de las tareas e) e i).
- f. Una conclusión, en la que se incluya una opinión sobre el uso de las herramientas empleadas y una valoración sobre el proceso de desarrollo.

#### **Sobre el programa:**

- a. Fuentes.
- b. Ejecutable
- c. Informe final en formato pdf.

**Evaluación:** detallamos este punto en la sección 2.1.

### **3. Resultados**

De acuerdo con la investigación presentada por Firat [9] que trata de explicitar la visión de los alumnos y docentes universitarios sobre el impacto de ChatGPT en la educación, podemos realizar el siguiente *análisis general* sobre el uso de esta herramienta en la práctica integradora diseñada:

- *Impacta en la evolución del aprendizaje y del sistema de educación.* Para el caso de la asignatura detallada, los alumnos pudieron utilizar toda clase de herramienta considerando, por ejemplo, mecanismos para depurar códigos con error, explicando porqué se realizan las implementaciones de una determinada manera, experimentando realizar modificaciones al código proporcionado por la herramienta y documentando el porqué las realizan.
- *Cambia el rol de los docentes.* Con estos talleres integradores, buscamos que los alumnos desarrollen tanto habilidades duras inherentes a su perfil profesional, como habilidades blandas, mediante las cuales se fortalece el pensamiento crítico y creativo.
- *Impacta en las valoraciones y evaluaciones.* Tal como se ha detallado, los criterios de evaluación prevén que se haga uso de cualquier herramienta que facilite el trabajo de los alumnos (pues en el futuro también las usarán), siempre que puedan dar cuenta de las ventajas e inconvenientes técnicos que se les presentan.

Desde un *punto de vista particular*, el trabajo integrador está diseñado para fomentar las siguientes competencias en los alumnos: a) identificación, formulación y resolución de problemas de informática; b) concepción, diseño y desarrollo de proyectos de informática; c) utilización de técnicas y herramientas de aplicación en la informática; d) generación de desarrollos tecnológicos y/o innovaciones tecnológicas; e) fundamentos para el desempeño en equipos de trabajo; f) fundamentos para la comunicación efectiva y; g) fundamentos para el aprendizaje continuo. Como se puede deducir de la lectura, entre las competencias que se desea que sean desarrolladas por los alumnos, se encuentran aquellas que profundizan

en la interacción humana y en el autoaprendizaje, y otras que requieren que el alumno decida críticamente cómo realizar la tarea de programación, de manera que posteriormente pueda dar cuenta del procedimiento empleado con solvencia técnica.

Sin embargo, en lo que respecta al uso de chatbots, se cumple lo afirmado por Fryer et al. [10]: las actividades mediadas por chatbots impactan negativamente en la interacción humana, e incluso disminuyen el interés de los alumnos, en comparación a si lo hicieran con un compañero humano. Así, a pesar del paso del tiempo y del desarrollo constante de tecnologías incidentes en la programación, no se pierde de vista el hecho de solicitar que el desarrollo del trabajo integrador sea grupal y principalmente basado en la propia creatividad. Más aún, los alumnos refirieron que el uso de ChatGPT en particular tornó las actividades propias necesarias para desarrollar un intérprete más complejas. Esto nos alienta a profundizar competencias de interacción humana que, según Dempere [11], son necesarias para el desenvolvimiento profesional futuro en equipos de desarrollo. En este caso en particular, los alumnos propusieron una solución al taller que consistió en programar un analizador léxico para sentencias de Pascal, utilizando el lenguaje de programación JAVA y el entorno NetBeans<sup>4</sup>. Lo más interesante es que por iniciativa propia (sin intervención de la IA) entendieron al Analizador Léxico como una clase que interactúa con otras 3 clases: buffer, token, tabla de símbolos, lo cual demuestra que pusieron en juego su creatividad para resolver el problema planteado, además de aplicar los conceptos técnicos proporcionados durante el cursado.

Como estadística general de la asignatura, se tiene que la totalidad de los alumnos aprueban y promocionan la obligación curricular (el 100% o un valor muy cercano a él) y no influye en ello el uso de un chatbot. Sin embargo, no se puede ignorar el avance tecnológico en la práctica docente y creemos conveniente ir incorporándolo, variando las estrategias.

## 4. Conclusiones y Trabajo Planificado

Desde *Lenguajes de Programación y Compiladores* entendemos que cada año académico que inicia es una oportunidad de mejorar el servicio que se le presta a los alumnos. En tal sentido, tomando como base lo propuesto por Rahman et al. [12] y considerando que es una herramienta que inexorablemente está presente en las actividades educativas y que su utilización podría ser provechosa, durante el nuevo cursado de la asignatura en este segundo cuatrimestre de 2024, se propondrán las siguientes modificaciones en el enunciado del taller integrador:

- Cada grupo le consultará a ChatGPT o herramienta similar sobre qué lenguaje es conveniente iniciar el proyecto de construcción de un intérprete. Cada grupo deberá argumentar (brindar razones a favor o en contra) si está de acuerdo o no con la sugerencia de la herramienta.
- Le solicitarán una introducción sobre el tema. En este caso, deberán analizar la confiabilidad de la información que la herramienta les proporciona.
- Le solicitarán la metodología para llevar a cabo el intérprete. Y la pondrán en práctica. Para ello, podrán determinar si todos los términos técnicos provistos por la herramienta son comprensibles y si fueron abordados en las clases teóricas y prácticas, o si se requiere alguna explicación adicional por parte de los docentes. Además, deberán en esta instancia evaluar las ventajas y desventajas de llevar a cabo los pasos indicados en forma grupal o en forma individual, decidiendo la más conveniente y justificando la decisión.

Otras prácticas ideadas para el año académico en curso a partir de la experiencia realizada son:

- Análisis de la especificidad de algún tema de la asignatura. Se propondrá que los estudiantes elijan un tema, realicen un cuestionario y luego analicen la profundidad en las respuestas vertidas por el ChatGPT. Así, el estudiante pondrá a prueba el pensamiento crítico y reflexivo para evaluar la profundidad de las respuestas de acuerdo con la siguiente escala: básica, básica pero hay que pensar la respuesta, compleja, poco entendible.
- Indagación en otros contextos. Se propondrá una actividad usando ChatGPT, en la que se investigue sobre las expresiones regulares y gramáticas libres de contexto en dominios prácticos diferentes a los estudiados en la asignatura. El estudiante pondrá a prueba el pensamiento crítico y reflexivo para identificar otros contextos de uso de la herramienta.
- Descubrimiento de las técnicas de optimización usadas por ChatGPT. El optimizador de código es una fase del compilador que tiene como función principal obtener código que se ejecute más rápidamente y que ocupe menos espacio de memoria. Para la comprensión de los conceptos involucrados en el tema y la experimentación mediante la realización de programas, se propone:
  - (i) Elaborar ejemplos de programas que contengan defectos que deben ser optimizados, como por ejemplo, subexpresiones comunes, expresiones de copias y código inactivo, código que calcula valores invariantes en un bucle, etc .
  - (ii) Realizar manualmente las transformaciones a los ejemplos.
  - (iii) Ingresar los programas sin las trasformaciones en ChatGPT. Con los resultados obtenidos determinar qué técnicas de optimización aplicó la IA y comparar las dos transformaciones. Por lo tanto, el estudiante pondrá en marcha sus herramientas técnicas para determinar qué conceptos y técnicas a usado para la optimización.

<sup>4</sup> Machado Ana Paula; Oller Mauricio; Orellana Carlos Agustín. Taller Integrador. Noviembre 2023.

Con estas nuevas propuestas se espera que los alumnos desarrollen las siguientes competencias: capacidad para identificar, plantear y resolver problemas, actuar en nuevas situaciones, generar nuevas ideas (creatividad), capacidad de investigación, de abstracción, de análisis y síntesis, de comunicación oral y escrita y, habilidades para tomar decisiones argumentadas y para trabajar en forma autónoma.

## Referencias

- [1] Lo, C. K. (2023). What is the impact of ChatGPT on education? A rapid review of the literature. *Education Sciences*, 13(4), 410.
- [2] Grassini, S. (2023). Shaping the future of education: exploring the potential and consequences of AI and ChatGPT in educational settings. *Education Sciences*, 13(7), 692.
- [3] Montenegro-Rueda, M., Fernández-Cerero, J., Fernández-Batanero, J. M., & López-Meneses, E. (2023). Impact of the Implementation of ChatGPT in Education: A Systematic Review. *Computers*, 12(8), 153.
- [4] Halaweh, M. (2023). ChatGPT in education: Strategies for responsible implementation. *Contemporary Educational Technology*, 15(2), ep421. <https://doi.org/10.30935/cedtech/13036>.
- [5] Firaina, R., & Sulisworo, D. (2023). Exploring the usage of ChatGPT in higher education: Frequency and impact on productivity. *Buletin Edukasi Indonesia*, 2(01), 39-46.
- [6] Chen, L., Chen, P., & Lin, Z. (2020). Artificial intelligence in education: A review. *Ieee Access*, 8, 75264-75278.
- [7] Chen, X., Zou, D., Xie, H., Cheng, G., & Liu, C. (2022). Two decades of artificial intelligence in education. *Educational Technology & Society*, 25(1), 28-47.
- [8] Aho, A. V.; Lam, M.; Sethi, R.; Ullman, J. D. (2008). *Compiladores: Principios, Técnicas y Herramientas*. Addison-Wesley. 2008.
- [9] Firat, M. (2023). What ChatGPT means for universities: Perceptions of scholars and students. *Journal of Applied Learning and Teaching*, 6(1).
- [10] Fryer, L., Ainley, M., Thompson, A., Gibson, A., and Sherlock, Z. (2017). Stimulating and sustaining interest in a language course: an experimental comparison of chatbot and human task partners. *Comput. Human Behav.* 75, 461–468. doi: 10.1016/j.chb.2017.05.045
- [11] Dempere, J., Modugu, K., Hesham, A. & Ramasamy, L.K. (2023). The impact of ChatGPT on higher education. *Front. Educ.* 8:1206936. doi: 10.3389/feduc.2023.1206936
- [12] Rahman, M., Terano, HJR, Rahman, N., Salamzadeh, A., Rahaman,S.(2023). ChatGPT and Academic Research: A Review and Recommendations Based on Practical Examples. *Journal of Education, Management and Development Studies*, 3(1), 1-12.

# **El pensamiento computacional como propuesta tecnopedagógica centrada en la resolución de problemas para el aprendizaje de la geometría en cuarto de primaria**

Maestrando: Marcos Ariel Conte

Directora: Dra. Fabiana Justina Gilardoni

Adaptación de Trabajo Final para Maestría en Enseñanza en Escenarios Digitales

AUSA. UNCuyo, Mendoza, Argentina

[maconte@mendoza.edu.ar](mailto:maconte@mendoza.edu.ar), [fjgilardoni@fhvcs.unam.edu.ar](mailto:fjgilardoni@fhvcs.unam.edu.ar)

## **Resumen**

En el siguiente trabajo, se presenta y aplica una propuesta tecnopedagógica, cuyo objetivo implica la utilización del Pensamiento Computacional (PC), como forma de potenciar la resolución de problemas en entornos digitales, así como otros componentes que se despliegan del mismo, en el aprendizaje de cuadriláteros para cuarto grado de nivel primario. Para lograr esto, dicho proyecto consta de cuatro actividades que incluyeron la construcción de mapas conceptuales como forma de abstraer los conceptos principales, la construcción de figuras geométricas en entorno de programación por bloques, siguiendo consignas pre establecidas, la elaboración de juegos tipo puzzle o trivia, que hagan referencia a los cuadriláteros y conceptos adyacentes, a partir de aplicar estrategias y componentes sobresalientes del PC. Finalmente alude a contextos interactivos y lúdicos para motivar a los estudiantes en la reflexión y evaluación de la propuesta educativa. De esta manera, se adapta una dinámica de trabajo que implica cambios en la concepción de los roles tradicionales, fomentando el trabajo colaborativo, apostando al uso de tecnologías y pedagogías emergentes, registrándose como una iniciativa de notable mérito motivacional, lúdico, colaborativo y creativo para potenciar habilidades tanto del ámbito de la geometría como para la formación de ciudadanos digitales.

**Palabras clave:** Pensamiento computacional; Resolución de problemas; Geometría; Educación primaria.

## **1. Introducción**

La institución educativa del que este trabajo se expide, es la escuela de gestión estatal de educación primaria N° 1-449 Manuel Ignacio Molina, del distrito de Pedro Molina, en el departamento de Guaymallén, provincia de Mendoza. Dicha ubicación la dispone en un ámbito urbano-marginal, atendiendo a un contexto de alta vulnerabilidad, bajo el lema “Todos los mundos en la escuela y la escuela en el mundo”. En el mencionado centro educativo, se comienza a analizar en el transcurso del año 2.021, bajo una experiencia de estudio en contexto, tanto aspectos intrínsecamente institucionales, como particulares que atañen al cuarto grado de segundo ciclo, entendiéndolo como un segmento inflexible en el trayecto escolar del estudiantado de nivel primario.

### **1.1. Descripción del problema**

El punto de partida del presente trabajo, nace del análisis del contexto por medio de entrevista a directora y docente de grado, observación de clase y lecturas de documentos, a través de los cuales se encontró falencias en el desarrollo de comprensión de textos, así como de resolución de problemas en los estudiantes, lo cual se refleja en un bajo rendimiento en informes diagnósticos y resultados de exámenes estándares nacionales, como una constante en los últimos años, así mismo se percibió falta de motivación en el aprendizaje en clase de matemáticas.

Es en este último campo, donde la implementación de resolución de problemas puede potenciar el aprendizaje sobre figuras geométricas, de manera que los alumnos logren apropiarse de los conceptos intervenientes mediante un proceso de construcción, reflexión y abstracción de saberes, fuera de apreciar las figuras como productos reales. Ante tal motivo, se ve la necesidad de incentivar habilidades en resolución de problemas, pero encarándolo desde un punto de vista menos tradicional, que favorezca a la predisposición y motivación de los estudiantes, de manera de aprovechar el vasto equipamiento tecnológico con el que cuenta la escuela. No se pretende incluir escenarios digitales sin un sentido pedagógico específico, sino potenciar nuevas formas de aprender que facilitan y despiertan otras oportunidades educativas atravesando de manera transversal las áreas curriculares, desprendiendo nuevas dinámicas, vinculaciones y roles entre los involucrados en el acto educativo.

Ante este escenario, se propone integrar las TIC en el aprendizaje cuadriláteros, siendo parte de la currícula de cuarto grado de primaria dentro del ámbito de la geometría, para enriquecer y acercar las propuestas educativas al estudiante del siglo XXI, lo cual no sólo delimita en la simple utilización de recursos tecnológicos como meras herramientas sino que lleva implícito, un cambio de roles de los actores educativos, y la correspondiente necesidad de renovación de dinámicas y formas de trabajo en el aula. En este sentido, se plantea la utilización de pedagogías y tecnologías emergentes, haciendo foco en fomentar destrezas y habilidades incluidas en la aplicación del Pensamiento Computacional (PC), resaltando la resolución de problemas, para la puesta en marcha de una propuesta tecnopedagógica

Dicho lo anterior, se aprovecha la formación inicial de la docente de grado, quien en entrevista expresa que ha participado de jornadas de capacitación en programación por bloques, además se tiene en cuenta la participación de alumnos de cuarto grado en capacitación en programación por bloques, quienes recibieron talleres en entornos de programación Scratch y Pilas Bloques, por el Auxiliar de Informática Educativa (AIE) en el primer cuatrimestre del año 2.022, de manera que dicho aprendizaje, lejos de permanecer aislado de la currícula oficial de primaria, pueda complementarse e integrarse en nuevos conocimientos.

Finalmente, para concretar las actividades formuladas se propone el trabajo en pareja pedagógica de docente de grado y AIE, siendo el acompañamiento de este último a lo largo de la implementación de la propuesta, de manera de auxiliar en aspectos técnicos, como, asesor docente y guía en los trayectos de aprendizaje que impliquen interacción con TIC. En este sentido, se distribuyen a disposición del proyecto una hora más de la obligatoria tribuida institucionalmente de informática educativa pudiendo ampliarse en caso de ser necesario a 3HR semanales a disposición del proyecto aplicado de manera presencial.

## 1.2. Marco Teórico

Las tecnologías de la información y la comunicación (TIC), pueden constituirse en medios que ayuden al mejoramiento de los procesos de enseñanza y aprendizaje. Por esta razón, desde hace algún tiempo, se ha venido dotando a las escuelas con ordenadores, software educativo y acceso a Internet [1]. Siguiendo esta línea, en los establecimientos de enseñanza primaria de la República Argentina se implementaron los planes “Primaria digital” y “Aprender Conectados”, planteando la inclusión de infraestructura tecnológica específica para las escuelas, de manera de facilitar el desarrollo de competencias en alfabetización digital, así como de habilidades para afrontar el mundo actual (Ministerio de Educación, Cultura, Ciencia y Tecnología). Esto se ve reflejado en muchas prácticas educativas que brindan experiencias resultantes en la optimización de la interacción y uso que los estudiantes hacen de la tecnología para la resolución de problemas que afectan a su entorno [2].

Particularmente, el papel de la incorporación de las TIC en la enseñanza de las ciencias exactas, ofrece, como plantea Coll [3], la posibilidad de brindar entornos atractivos a los alumnos y favorables al docente, en un proceso de enseñanza aprendizaje más dinámico y activo. De esta manera se puede promover las habilidades en resolución de problemas, abordándose de manera innovadora con tecnologías emergentes que favorezcan no solo a la motivación del alumno, rompiendo estructuras tradicionales de organización de clase, sino también, promoviendo ciertas habilidades, comportamientos y destrezas por parte de estudiantes y docentes, agregándole un valor a la integración de los recursos y entornos digitales para posibilitar aprendizajes significativos. Es así como la utilización de ciertos recursos digitales, acompañados de modelos de aprendizajes no tradicionales y tecnologías emergentes, que incluyen el desarrollo del pensamiento computacional, y la gamificación, pueden posibilitar experiencias de aprendizajes ricas, contextualizadas y significativas [4].

El PC, término acuñado por Wing [5], describe a “los procesos de pensamiento involucrados en la formulación de un problema y la expresión de su/s solución/es de tal manera que una computadora (humana o máquina) pueda llevarlo a cabo de manera efectiva”. En este punto se rescatan otras características asociadas al PC, clasificadas por Basogain et al. [6] en base a las definiciones de Wing, que se pueden detallar en:

- Reformular un problema a uno parecido que sepamos resolver por reducción, encuadrarlo, transformar, simular
- Pensar Recursivamente
- Procesar en Paralelo
- Interpretar código como datos y datos como código
- Generalizar análisis dimensional
- Reconocer ventajas y desventajas del solapamiento
- Reconocer coste y potencia de tratamiento indirecto y llamada a proceso
- Juzgar un programa por simplicidad de diseño

- Utilizar Abstracción y descomposición en un problema complejo o diseño de sistemas complejos
- Elegir una correcta representación o modelo para hacer tratable el problema
- Seguridad en utilizarlo, modificarlo en un problema complejo sin conocer cada detalle
- Modularizar ante múltiples usuarios
- Prefetching y caching anticipadamente para el futuro
- Prevención, protección, recuperarse de escenario peor caso
- Utilizar razonamiento heurístico para encontrar la solución
- Planificar y aprender en presencia de incertidumbre
- Buscar, buscar y buscar más
- Utilizar muchos datos para acelerar la computación
- Límite tiempo/espacio y memoria/potencia de procesado

Las características antes descriptas, implican que el centro del pensamiento computacional, no está exclusivamente en el desarrollo de programas y uso de lenguajes de programación, sino en el desarrollo de habilidades del pensamiento, que implican brindar soluciones en base a la formulación de instrucciones secuenciadas, siendo esto una forma de resolución de problemas característica de los tiempos contemporáneos [7].

La inclusión de dichas instrucciones secuenciadas, implícitas en la experiencia de programar, abre a la escuela una oportunidad para incorporar distintas maneras de organizar la enseñanza, con otras formas de acercarse a la construcción de los saberes; un encuentro creativo y dinámico, con lugar para el despliegue de la inteligencia personal y colectiva, en un ambiente lúdico donde la resolución de problemas se lleva a cabo mediante la programación de juegos [4]. Así mismo, posibilita estimular el aprendizaje por medio de la curiosidad, el juego, el descubrimiento y la posibilidad de sumergirse en escenarios multidimensionales [8].

Tal es el caso del entorno de programación gráfico Scratch, que fue desarrollado por el Instituto Tecnológico de Massachusetts (MIT), con la intención de iniciar a niños y jóvenes como pensadores computacionales, aprendiendo conceptos básicos de informática y matemática algorítmica, así como diseñar, resolver problemas y colaborar entre pares, facilitando el desarrollo de diversas habilidades multidisciplinares que se ponen en marcha cuando se elaboran pequeños programas. Por su lado, el entorno PilasBloques, integra secuencias didácticas que involucran la programación para la resolución de desafíos planteados con complejidad creciente, diseñados por docentes e investigadores de diferentes universidades argentinas en colaboración con Program.Ar y Fundación Sadosky, de manera que el alumno tenga un acercamiento a la programación por bloques. De esta manera, en una interfaz similar a Scratch, se puede seleccionar y arrastrar los bloques, siguiendo un orden lógico para cumplir con la tarea asignada [6] [9].

Con lo expuesto se trata de destacar, tal como lo afirma Gamboa [4], “la importancia del uso de la tecnología como un medio que permite al estudiante obtener conclusiones y realizar observación”, no como mero reemplazo de tecnologías tradicionales como el lápiz y el papel, sino que brindando un valor agregado, que se traduzca en potenciar nuevas experiencias de aprendizaje.

## **1.2. Objetivos**

En base a lo descripto anteriormente en este trabajo, se plantea diseñar e implementar una propuesta de enseñanza para cuarto grado con centro en la matemática pero que articule con otras áreas del conocimiento, como lengua, por medio de la implementación de tecnologías y pedagogías emergentes, como el desarrollo del pensamiento computacional, apoyado del uso de entornos educativos para el aprendizaje de la programación con Pilas Bloques y Scratch que involucran espacios gamificados como motor motivacional y para contextualizar experiencias de aprendizaje [10]. De esta forma, se pretenden potenciar saberes de la rama de la geometría, particularmente el tema “cuadriláteros”, en el cuarto año de educación primaria [1]. Por ello, surge el interrogante:

**¿Cómo la propuesta tecnopedagógica despliega el aprendizaje de cuadriláteros mediante la resolución de problemas a partir de actividades que estimulen el desarrollo del PC en estudiantes de 4º grado?**

## **1.3. Objetivos**

### **Objetivo general**

- Diseñar e implementar una propuesta tecnopedagógica para la enseñanza de cuadriláteros, que involucre el desarrollo del pensamiento computacional junto con sus habilidades correlacionadas, resaltando entre ellas la resolución de problemas, en cuarto grado de primaria.

### **Objetivos específicos**

- Diseñar y desarrollar secuencias de actividades que promuevan el despliegue de habilidades inherentes al PC para el aprendizaje de cuadriláteros.
- Promover el trabajo colaborativo tanto en alumnos como de docente de grado y AIE.
- Incentivar la utilización de pedagogías y tecnologías emergentes para enriquecer el proceso de enseñanza.
- Estimular el uso de entornos digitales para la resolución de problemas, de forma creativa.
- Recabar resultados cualitativos de la implementación pautada, en base a la comunicación con los sujetos intervenientes en el proyecto.

## 2. Descripción de la Experiencia

En este apartado, se describe para la modalidad educativa presencial, la inclusión de un proyecto que comprende aspectos multidisciplinares interárea, presentando una secuencia tecnopedagógica, que abarca cuatro actividades, resaltando e incluyendo ejercicios pedagógicos desde el enfoque socioconstructivista, orientados a promover la resolución de problemas en el área de matemática, haciendo foco en el desarrollo del PC, por medio de la creación de producciones multimediales en entornos de programación por bloques. Con este objeto, se presentan las actividades pautadas para el aprendizaje de geometría, adentrándose, más precisamente, en el tema de cuadriláteros, en el cuarto grado de nivel primario, como se expone en la Tabla 1.

**Tabla 1.** Actividades que incluyen el proyecto educativo para el aprendizaje de cuadriláteros.

Nº	Descripción de la actividad	Tiempo
1	Conceptos de cuadriláteros, armado de un mapa conceptual en aplicación para presentación de diapositivas.	1 clase semanal de 2HR.
2	Construcción de figuras geométricas en respuesta a actividades prediseñadas del Entorno de programación por bloques PilasBloques.	1 clase semanal de 2HR.
3	Diseño y construcción de un mini juego tipo puzzle o trivia, sobre cuadriláteros, en entorno de programación por bloques Scratch.	3 clases (1 por semana) de 2HR.
4	Actividad gamificada tipo caza del tesoro, utilizando códigos QR y cuestionarios sobre temas vistos en el proyecto, enriquecido con Realidad aumentada (RA).	1 clase semanal de 2HR.

Las actividades enunciadas en la Tabla 1, incluyeron un acercamiento teórico del tema cuadrilátero en la clase 1, como base para armar un mapa conceptual que acompañó como guía de aplicación en el resto de las secuencias aplicadas, siguiendo por la construcción de figuras geométricas, aprovechando las actividades prediseñadas para el entorno PilasBloques, en donde los estudiantes, pudieron aplicar conceptos de ángulos, lados y características de los cuadriláteros, en un contexto de despliegue de habilidades del PC, intrínsecas en el desarrollo de la actividad 2. De igual manera, la tercera actividad promovió el despliegue de habilidades del PC, realzando la creatividad de los estudiantes al crear juegos interactivos cortos, aplicando conceptos de cuadriláteros, finalmente se hizo un cierre que hizo uso de la RA como entorno gamificado de aprendizaje, para repasar conceptos adquiridos, como paso previo a una reflexión grupal final en la formación de una nube de palabras, como se detalla a continuación en el diseño instruccional de la propuesta.

## 2.1. Propuesta Didáctica

**Tabla 2.** Diseño instruccional (DI) de actividades.

	Tema	Objetivos	Materiales	Actividades
Actividad1	Cuadriláteros, conceptos y clasificación.	<p>Reconocer los distintos tipos de cuadriláteros y sus características.</p> <p>Relacionar conceptos, clasificando y jerarquizando el conocimiento.</p> <p>Promover la participación grupal y el trabajo colaborativo</p> <p>Utilizar los entornos digitales para la resolución de problemas, en colaboración con los pares.</p>	<p><a href="#">Guía de clase 1</a> presentada con asistencia de proyector multimedia y pizarra digital</p>	<p><i>Inicio</i> Observación y análisis de <a href="#">video</a> sobre cuadriláteros.</p> <p><i>Desarrollo</i> Armado de mapa conceptual en presentador de diapositivas, utilizando netbooks y pizarra digital</p> <p><i>Cierre</i> Autoevaluación con <a href="#">REA interactivo de preguntas y respuestas</a></p>
Actividad2	Identificación y construcción de cuadriláteros en entornos de programación por bloques.	<p>Reconocer los distintos tipos de cuadriláteros.</p> <p>Aplicar conceptos adquiridos en la construcción de figuras, reflexionando sobre los pasos utilizados para la concreción de los mismos.</p> <p>Promover la participación grupal y el trabajo colaborativo</p> <p>Utilizar los entornos digitales para la resolución de problemas, en colaboración con los pares.</p>	<p><a href="#">Guía de clase 2</a> presentada con asistencia de proyector multimedia y pizarra digital</p>	<p><i>Inicio</i> Recuperación de saberes con el apoyo de <a href="#">REA interactivo</a>.</p> <p><i>Desarrollo</i> Resolución de problemas geométricos en entorno Pilas Bloques</p> <p><i>Cierre</i> Reflexión y metacognición de lo aprendido en clase.</p>
Actividad 3	Aplicación de contenidos aprendidos sobre cuadriláteros en la construcción de juegos en entornos de programación por bloques.	<ul style="list-style-type: none"> <li>• Afianzar conocimientos en algoritmos y programación.</li> <li>• Identificar y resolver problemas del campo de la geometría de forma creativa, desde el campo de la programación.</li> <li>• Reconocer características de los cuadriláteros.</li> <li>• Fomentar habilidades de compartir experiencias y la elaboración de estrategias mediadas por entornos digitales para la resolución de problemas en colaboración con sus pares, en un marco de respeto y valoración de la diversidad.</li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">Guía de clase 3</a> presentada con asistencia de proyector multimedia y pizarra digital</li> <li>• Entorno de programación Scratch 3.0</li> </ul>	<p><i>Inicio</i></p> <ul style="list-style-type: none"> <li>• Recuperación de saberes aprendidos sobre cuadriláteros y programación por bloques, <a href="#">presentación de las actividades</a> a realizar.</li> </ul> <p><i>Desarrollo</i></p> <ul style="list-style-type: none"> <li>• Creación de juego interactivo tipo puzzle o trivia, en grupos de trabajo, incluye como actividad previa el diseño de un breve guión del mismo</li> </ul> <p><i>Cierre</i></p> <ul style="list-style-type: none"> <li>• Co-evaluación de los juegos creados.</li> </ul> <p>Reflexión grupal.</p>

<b>Actividad 4</b> Reflexión, metacognición y evaluación de la propuesta implementada.	<ul style="list-style-type: none"> <li>• Reflexionar sobre el propio aprendizaje.</li> <li>• Repasar conceptos adquiridos a través del juego y el uso de tecnologías emergentes.</li> <li>• Promover la participación grupal y el trabajo colaborativo</li> <li>• Utilizar los entornos digitales para la resolución de problemas, en colaboración con los pares</li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">Guía de clase 4</a> presentada con asistencia de proyector multimedial y pizarra digital</li> <li>• Entorno interactivo de RA Metaverse</li> <li>• Entorno de construcción de nubes de palabras en portal <a href="http://abcya.com">abcya.com</a></li> </ul>	<p><i>Inicio</i></p> <ul style="list-style-type: none"> <li>• Recuperación de saberes aprendidos enproducciones interactivas en entornos de programación por bloques, <a href="#">presentación</a> de la actividad con RA.</li> </ul> <p><i>Desarrollo</i></p> <ul style="list-style-type: none"> <li>• Actividad gamificada tipo caza del tesoro utilizando tablets, lectura de códigos QR escondidos por la escuela, con enlace a cuestionarios en aplicación de RA Metaverse como autoevaluación</li> </ul> <p><i>Cierre</i></p> <ul style="list-style-type: none"> <li>• Reflexión, metacognición y evaluación de la propuesta pedagógica en forma grupal. Formación de <a href="#">nube de palabras</a> con palabras representativas de la experiencia.</li> </ul>
---	---	--	---

## 2.2. Contexto

La secuencia expuesta se desarrolló con alumnos/as de cuarto grado, correspondiente al segundo ciclo de educación primaria cuyo rango etario osciló entre los 9 y 10 años, constando de un grupo reducido de 16 alumnos, quienes fueron partícipes, en la primer mitad del año, de un taller de programación en entornos por bloques.

Institucionalmente el personal involucrado, incluye a la Docente de grado en su rol de guía educativo, Docentes de áreas especiales (AIE, Educación física) en roles de guía y/o soporte, Personal directivo en la supervisión de las actividades propuestas y la figura del maestrando en la coordinación e implementación pedagógica de la propuesta sumada a su función escolar como AIE

La dinámica organizativa de trabajo se distribuyó en clases presenciales durante el mes de noviembre de 2.022, dentro del establecimiento escolar, aprovechando los recursos tecnológicos disponibles, particularmente en el aula de informática, incluyendo netbooks, tablets, proyector multimedial y pizarra digital.

## 2.3. La Experiencia

La implementación de la experiencia educativa implicó fomentar el trabajo entre pares, abriendo espacios de autoevaluación y coevaluación, en el despliegue y acompañamiento de competencias y habilidades de resolución de problemas, como parte de los componentes que fomenta el PC, bajo la guía y seguimiento de la docente de grado junto con el AIE.

Por otro lado, si bien, el PC intervino de manera protagónica, fomentando el desarrollo de habilidades que acompañan al mismo en todas las actividades, la propuesta se complementó con recursos educativos abiertos (REA), bajo licencia Creative Commons (CC), que facilitó un marco legal para usar y compartir el conocimiento producido, respetando los derechos de autor, utilizando bienes culturales ya existentes para enriquecerlos y adaptarlos a las necesidades del usuario. De esta manera se aprovechó el conocimiento colectivo en la generación de riqueza cultural, lo cual resulta muy útil en el campo educativo [11]. También se incluyeron secuencias en las que, si bien no intervino la programación informática, se involucró el despliegue de algún elemento del PC.

Tal fue el caso de la **clase 1**, que inició con la proyección de un video sobre cuadriláteros, siendo el disparador para encausar un diálogo maestra – alumnos, evocando conocimientos previos con los expuestos en el video, que devino en la propuesta de armar grupalmente un mapa conceptual de la temática, por medio del uso de software de presentación de diapositivas y asistidos de pizarra digital. Esto permitió visualizar relación entre diferentes conceptos, desarrollar el pensamiento crítico, todo en un entorno que promovió la construcción colaborativa de conocimiento. Por su parte los estudiantes replicaron la producción de la pizarra, trabajando en las netbooks, pudiendo adaptar aspectos de formato y diseño a su gusto, siendo posteriormente impresas para incorporar a sus carpetas. En el desarrollo de las producciones el AIE guió y acompañó en el uso del software, siendo ya conocido por el grupo de estudiantes. Finalmente la clase

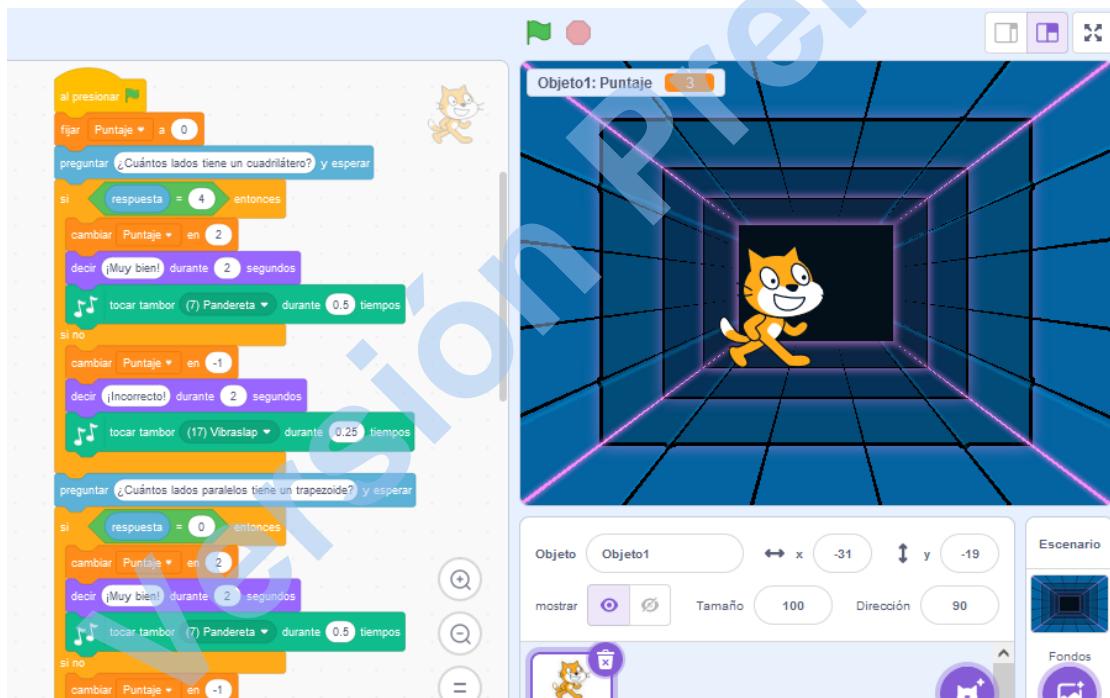
culminó con la propuesta de acceder a un cuestionario interactivo de autoevaluación, cuyo acceso se facilitó desde el apartado del weblog escolar, disponible para el proyecto.

En la segunda clase, se promovió la resolución de problemas, recurriendo en el inicio a un juego interactivo para retomar la atención de los estudiantes, en el que pudieron formar robots conformados por cuadriláteros, para luego, de la mano del entorno de programación Pilas Bloques y parte de su cartilla de ejercicios programados para el segundo ciclo de educación primaria, construir mediante secuencias de programación gráfica, diferentes figuras, teniendo en cuenta aspectos como ángulos y lados. De esta manera, siguiendo los lineamientos previstos por el MECCyT [1], se integró la educación digital y la programación para posibilitar múltiples oportunidades de enseñanza y aprendizaje, habilitando diversos modos de comunicación y creación, a través del juego. Una vez más, la docente y el AIE actuaron como facilitadores, brindando orientación y apoyo durante este proceso, culminando en el análisis de las características que vislumbradas de las figuras geométricas encontradas, tales como, cantidad de lados y comparativa de estos, cantidad de ángulos y clasificación. En este espacio se compartieron los diferentes algoritmos construidos para llegar a la solución requerida, de manera de poder apreciar diferentes caminos para llegar a la misma.

Llegando a la actividad 3, luego de un espacio de diálogo para recuperación de saberes, con soporte de presentación interactiva, se promovió desde el área de Lengua, la confección de un breve guion de un juego corto, del tipo trivia o puzzle, de manera de demostrar lo aprendido sobre cuadriláteros y conceptos subyacentes, que posteriormente en parejas de trabajo se codificaron en el entorno de programación gráfico Scratch 3. Dichas producciones, luego fueron testeadas entre los diferentes equipos de trabajo, para su coevaluación, en base a una rúbrica específica para dicha tarea. En esta actividad particularmente se pudo apreciar el desarrollo de componentes del PC en las producciones de los alumnos, lo cual devino en una diversificación de soluciones, enmarcado en un ambiente de motivación, creatividad, intercambio, acompañamiento y guía del aprendizaje tanto de los docentes como entre los pares.

El desarrollo de los juegos interactivos resalta de forma explícita la firma que cada grupo de trabajo, deja en los detalles incluidos en la actividad, lo cual hace apreciar la creatividad que los estudiantes incluyen en la resolución del problema propuesto, como se observa en la Figura 1.

**Figura 1.** Fragmento de bloques de juego de preguntas.



En la Figura 1, se observa un fragmento del juego de preguntas, cuya dinámica incluye al personaje principal de Scratch, un gato llamado Scratchy, quien por globos de diálogo realiza preguntas al usuario, el cual puede responder por medio del uso del teclado, así mismo, por cada respuesta correcta, se agrega un puntaje. Por otro lado las preguntas diseñadas por los alumnos, atañen a saberes teóricos de identificación de las figuras geométricas que corresponden a algún tipo de cuadrilátero, buscando respuestas concretas y breves.

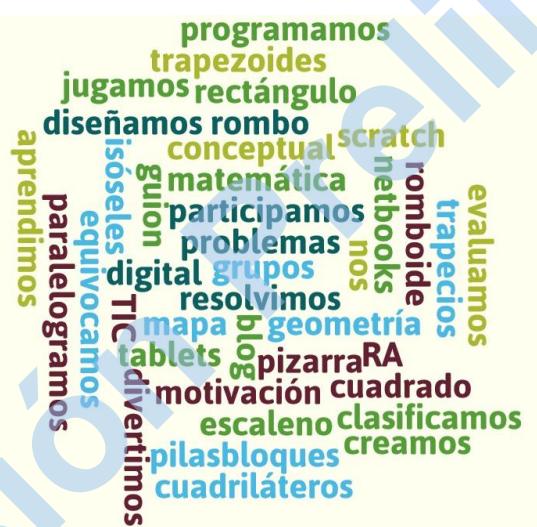
Finalmente, la **actividad de cierre** del proyecto, comenzó con una reflexión grupal guiada tanto por la docente de grado como el AIE, de lo trabajado las últimas semanas, dando pie a la autoevaluación de los mismos, mediante el uso de tablets y la aplicación Metaverse, contando con la asistencia de la profesora de educación física, en la exploración del entorno buscando códigos QR. Al escanear cada código, accedieron a cuestionarios de RA relacionados con los cuadriláteros y la construcción de algoritmos como se muestra en la Figura 2. Dicha actividad fue muy apreciada por los alumnos, quienes se vieron muy incentivados para cumplimentarla.

**Figura 2.** Captura de Cuestionario interactivo con RA.



La propuesta concluyó de manera grupal, construyendo una nube de palabras mediante el uso del portal abcyah.com, como de proyector y pizarra digital. Esta última producción demostró muy resumidamente lo aprendido y vivenciado en esta experiencia, a modo de metacognición, tal como se aprecia en la Figura 3. Como última observación, se decidió que las producciones y experiencias realizadas se publican en el weblog escolar, para su registro y difusión con la comunidad educativa.

**Figura 3.** Nube de palabras.



### 3. Resultados

En este trabajo se diseñó e implementó una propuesta tecnopedagógica para la enseñanza de cuadriláteros en cuarto grado de primaria, que involucró el desarrollo del pensamiento computacional y sus habilidades correlacionadas, siendo una de ellas la resolución de problemas, la cual es considerada por la institución educativa como una de las habilidades que emerge reforzar en el estudiantado.

Se destacaron de dicha propuesta, los componentes propios del PC, que los estudiantes pudieron desplegar, al momento de resolver las actividades presentadas durante las diferentes situaciones educativas pautadas, haciendo énfasis en el desempeño aplicado en el trabajo en entornos de programación por bloques orientados específicamente para el público infantil, materializándose en la producción creativa de soluciones multimediales, bajo la mediación y guía de un equipo docente compuesto por la maestra del grado junto con el AIE del establecimiento, quienes se desempeñaron colaborativamente tanto en la implementación tecnopedagógica como en el diseño, selección de materiales y entornos de trabajo.

En la integración áulica de pedagogías y tecnologías emergentes que acompañaron a lo largo del proyecto, se propició un contexto motivador de trabajo entre los actores participes, quienes aceptaron de buen grado los REAs, la producción y coevaluación de las soluciones multimediales creadas mediante la aplicación del PC, así como el trabajo colaborativo respetuoso entre los pares y la interacción con la RA en un entorno gamificado para la autoevaluación de los conocimientos y habilidades incorporados, que incluyeron un recorrido interárea, más allá del implícito por el campo de la geometría, en un ambiente lúdico que promovió la creatividad, motivación e incentivó y acercó a los estudiantes a un aprendizaje activo.

## 4. Conclusiones

La experiencia educativa descripta, resultó de gran valor motivador para los estudiantes que participaron de manera entusiasta, permitiendo adquirir conocimientos específicos conceptuales geométricos como de áreas involucradas de manera adyacente en el proyecto educativo, además de fortalecer habilidades y competencias involucradas al aplicar el PC, en las que se destaca la resolución de problemas y el despliegue de su creatividad y trabajo colaborativo, contribuyendo a una formación y desarrollo integral de los estudiantes del siglo XXI.

En pos de iniciativa de mejoras a tener en cuenta para implementaciones de proyectos a futuro, se reflexiona ahondar en aspectos que pueden generar cambios a través de determinaciones institucionales, tales como los siguientes:

- Establecer mecanismos institucionales de capacitación docente, tanto en el uso de herramientas digitales, como su reflexión para la implementación pedagógica en la práctica docente, fundamentalmente relacionadas al PC, para la apertura de nuevas formas de creación e interacción de contenidos por y para estudiantes.
- Entablar puentes de colaboración profesional entre docentes de diferentes áreas, resaltando el compartimiento de experiencias relevantes para el trabajo en proyectos.
- Extender la integración del PC a todas las áreas del currículo, de manera de establecer un vínculo entre los talleres de programación y su aplicación pedagógica en el aprendizaje áulico.

## 5. Agradecimientos

Un agradecimiento especial a la comunidad educativa de la Escuela 1-449 Manuel Ignacio Molina, en donde se desarrolló la propuesta descripta, durante el mes de noviembre de 2022.

A la Dra. Fabiana Justina Gilardoni, quien dirigió el trabajo de base para su presentación como Trabajo Final de la Maestría en Enseñanza en Escenarios Digitales, de la Asociación de Universidades Sur Andinas, para la sede UNCuyo, Mendoza, Argentina.

## Referencias

- [1] Jaramillo-Marín, P. (2005). Uso de Tecnologías de Información en el Aula. *¿Qué saben hacer los niños con los computadores y la información?* Revista de Estudios Sociales, 20 (6), 27-44.[https://www.researchgate.net/publication/28176664\\_Uso\\_de\\_tecnologias\\_de\\_informacion\\_en\\_el\\_aula\\_que\\_saben\\_hacer\\_los\\_ninos\\_con\\_los\\_computadores\\_y\\_la\\_informacion](https://www.researchgate.net/publication/28176664_Uso_de_tecnologias_de_informacion_en_el_aula_que_saben_hacer_los_ninos_con_los_computadores_y_la_informacion)
- [2] Ministerio de Educación, Cultura, Ciencia y Tecnología (2018). NAP Educación digital, programación y robótica. República Argentina. <https://www.educ.ar/recursos/150123/nap-de-educacion-digital-programacion-y-robotica>
- [3] Coll, C. (2008). Aprender y enseñar con las TIC: expectativas, realidad y potencialidades. Boletín de la Institución Libre de Enseñanza, 72, 17-40. <https://www.educ.ar/recursos/70819/aprender-y-ensenar-con-las-tic-expectativas-realidad-y-potencialidades>
- [4] Gamboa, R. (2007). Uso de la tecnología en la enseñanza de las matemáticas. Revista Cuadernos de Investigación y Formación en Educación Matemática. 2 (3), 11-44.: [http://cimm.ucr.ac.cr/cuadernos/cuaderno3/cuaderno3\\_c1.pdf](http://cimm.ucr.ac.cr/cuadernos/cuaderno3/cuaderno3_c1.pdf).
- [5] Wing, J.M. (2017) Computational thinking's influence on research and education for all. Italian Journal of Educational Technology, 25(2), 7-14. doi: 10.17471/2499-4324/922
- [6] Basogain, X.; Olabe-Basogain, M. yOlabe-Basogain, J. (2015). Pensamiento Computacional a través de la Programación: Paradigma de Aprendizaje. Revista de Educación a Distancia (RED), (46).<https://revistas.um.es/red/article/view/240011>
- [7] Bordignon, F., e Iglesias, A. (2020). Introducción al pensamiento computacional [libro digital]. UNIPE. Editorial Universitaria. EDUCAR S.E.<http://biblioteca.clacso.edu.ar/gsdl/collect/ar/ar-050/index/assoc/D14927.dir/introduccion-pensamiento-computacional.pdf>
- [8] Coicaud S. (2019) Potencialidades didácticas de la inteligencia artificial, videojuegos, realidad extendida, robótica y plataformas. Mediaciones tecnológicas para una enseñanza disruptiva. Novedades Educativas.
- [9] Astudillo, G. J., Bast, S. G., yWillging, P. A. (2016). Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación. Virtualidad, Educación Y Ciencia, 7(12), 125–142. <https://revistas.unc.edu.ar/index.php/vesc/article/view/14739>
- [10] ArtilesFontales, C. G. (2019). Pilas Bloques. Aprende a programar jugando. Observatorio de tecnología educativa (18), 1-8. doi: 10.4438/2695-4176\_OTE\_2019\_847-19-121-5
- [11] [1BuenTip]. (2015, november 10) CreativeCommons | Qué es, como funciona y licencias CC. [Archivo de video]. <https://www.youtube.com/watch?v=Mf9ACZy4O18>

# Algoritmia y juego en la escuela primaria. Miradas sobre una experiencia desconectada

Martín Goin<sup>1</sup>, Edith Lovos<sup>2</sup>, María de la Trinidad Quijano<sup>3</sup>

<sup>1</sup>Universidad Nacional de Río Negro Sede Andina CIEDIS, San Carlos de Bariloche , Argentina

<sup>2</sup>Universidad Nacional de Río Negro Sede Atlántica CIEDIS, Viedma , Argentina

<sup>3</sup>Universidad Nacional de Río Negro Sede Andina; Universidad Nacional del Comahue, San Carlos de Bariloche, Argentina

{mgoin, elovos,mquijano}@unrn.edu.ar

## Resumen

El interés por la enseñanza y aprendizaje de la programación desde edades tempranas no es un tema novedoso, sin embargo la implementación en el aula se enfrenta a diversos desafíos, entre ellos, la disponibilidad y el acceso a recursos como computadoras. Una alternativa para atender estas situaciones, pueden ser las actividades desconectadas y en particular los juegos de mesa. En este trabajo, por una parte, se presenta el juego de mesa denominado JAM, desarrollado en el marco de un proyecto de extensión cuyo objetivo es propiciar un acercamiento a la resolución de problemas a través de la construcción de algoritmos en forma lúdica. Y por otra, se presentan los resultados obtenidos en diferentes experiencias de uso del juego, de las que han participado niños y docentes de algunas instituciones educativas de nivel primario de la ciudad de San Carlos de Bariloche. Entre los aportes del juego, los niños valoran la posibilidad de jugar en equipos, en tanto los docentes consideran que el mismo puede ser un aporte para el acercamiento, el repaso y/o la evaluación de conceptos básicos de programación evitando la ansiedad que puede provocar la interacción con la máquina y fomentando la reflexión.

**Palabras clave:** Juego de mesa, Aprendizaje, Algoritmia,

## 1. Introducción

En una sociedad cada vez más permeada por las tecnologías de la información y la comunicación (TIC), la enseñanza y aprendizaje de la programación y/o la robótica forman parte de las estrategias para promover el desarrollo del pensamiento computacional desde los inicios de la escolaridad. A través del pensamiento computacional, es posible el desarrollo de habilidades para resolver problemas complejos de diversas disciplinas haciendo uso de conceptos vinculados a la ciencia informática (abstracción, descomposición funcional, reconocimiento de patrones y diseño de algoritmos), y así esta forma de pensamiento forma parte de las competencias básicas para la ciudadanía del siglo XXI [1]. Sin embargo, como sostiene Vázquez Uscanga [2], a partir de un trabajo de investigación que incluye entre otros países de Latinoamérica a Argentina, la integración y el desarrollo de este tipo de pensamiento en el ámbito educativo, continúa planteando desafíos. Uno de ellos, asociado a repensar los procesos de enseñanza y aprendizaje desde el inicio de la escolaridad. Por otra parte, el autor, remarca que el pensamiento computacional ha derivado en áreas específicas como la programación, aún cuando su definición sea más compleja. En el caso del aprendizaje de la programación, un estudio documental reciente, llevado adelante por Molina [3] destaca que avanzar desde edades tempranas en la interacción humano-máquina ayuda a reforzar el pensamiento lógico matemático, promueve el desarrollo de habilidades blandas como el trabajo en equipo, la creatividad y el desarrollo de soluciones en forma estructurada a la vez que posibilita madurar habilidades cognitivas. Y por otra parte, identifica en el aprendizaje desconectado una oportunidad para abordar la brecha de acceso a recursos tecnológicos por parte de los estudiantes, e instituciones educativas. Otros autores como [4], sostienen que este tipo de aprendizaje, puede convertirse en una herramienta para trabajar la formación docente sobre pensamiento computacional y avanzar hacia una integración del mismo de forma contextualizada, interdisciplinar e inclusiva. En particular sobre el uso de juegos no

digitales para trabajar pensamiento computacional, en [5] se señala la importancia de los mismos para atender a los estilos de aprendizaje kinestésico que puede ser común en los niños de edad escolar. Otros estudios, como el propuesto por [6] sostienen que a través de actividades desconectadas y lúdicas, es posible mejorar las habilidades de resolución de problemas de los niños de corta edad (5-6 años). Volviendo sobre la formación docente, en Colombia, Carmona y Cardoso [7] dan cuenta de una experiencia con estudiantes de profesorado de matemáticas usando el juego de mesa conocido como CLUE y a través del cual los futuros docentes han podido desarrollar habilidades de descomposición, reconocimiento y generalización de patrones.

La provincia de Río Negro, donde tuvo lugar la experiencia que se presenta en este trabajo, se ubica en la región patagónica y abarca una superficie de 203013 km<sup>2</sup>, extendiéndose desde la cordillera al océano atlántico y atravesada de esta forma por diferentes condiciones climáticas y geográficas. Solo para exemplificar entre la ciudad capital, Viedma y ciudad cordillerana San Carlos de Bariloche existe una distancia de más de 800 km, en los que es posible encontrar pueblos y parajes para los cuales las TIC se convierten en una valiosa herramienta para la educación, el trabajo y la comunicación en general. Así, la provincia cuenta con antecedentes de inclusión de TIC en el sistema educativo, tanto en lo que refiere a la gestión [8] como a la inclusión en prácticas educativas en sus diferentes niveles, tal es el caso del programa Aulas Digitales Móviles implementado desde el año 2008 o los Centros de Educación Media Rural en modalidad virtual implementados desde 2010 [9]. En el caso específico del nivel primario, la Ley Provincial de Educación Nro. 4819 busca “promover los usos de las tecnologías de la información y la comunicación y la recepción crítica de los discursos mediáticos” (art. 32 inciso k) y, en relación con los derechos, señala que la educación mediada por TIC se desarrollará transversalmente para todos los niveles y modalidades (art. 94). Por otra parte, el Consejo Federal de Educación a través de la resolución 263/15, declara la importancia estratégica que tienen la enseñanza y el aprendizaje de programación para fortalecer el desarrollo socioeconómico de la Nación. En base a estos lineamientos, y buscando un acercamiento a la algoritmia de forma desconectada (como estrategia para abordar la barrera de acceso) y en forma lúdica, es que surge el juego de mesa denominado JAM.

El juego se alinea con el enfoque de aprendizaje constructivista, y ha sido diseñado ad-hoc en el marco de un proyecto de extensión en el contexto de la Universidad Nacional de Río Negro (UNRN) del cual participan docentes que llevan adelante prácticas de enseñanza en cursos de programación tanto para carreras vinculadas a ingeniería, informática y/o ciencias económicas.

En este trabajo, se comparten y discuten datos cuantitativos y cualitativos recolectados a partir de experiencias áulicas con el juego de mesa JAM en diferentes instituciones educativas de nivel primario de la ciudad de San Carlos de Bariloche.

## 1.1 Juego Algorítmico de Mesa

El juego que se presenta en este trabajo, se denomina JAM (Juego Algorítmico de Mesa) y propone la resolución de desafíos (problemas) de manera grupal y cooperativa, a través de la construcción de un algoritmo usando cartas y de acuerdo a reglas pre establecidas. JAM se diseñó durante 2020, en el marco de un proyecto de extensión ejecutado en la Universidad Nacional de Río Negro, y acreditado y financiado por la Secretaría de Políticas Universitarias. El juego, busca trabajar la resolución de problemas a través de la construcción de algoritmos en forma unplugged o desconectada, tomando como base dos lenguajes visuales de programación como son DaVinci concurrente [10] y la iniciativa basada en bloques conocida como Scratch [11].

En cuanto a sus componentes, JAM incluye 3 tableros, 72 cartas, 53 fichas (robots, números, obstáculos y premios), 11 objetivos, un reglamento y una etiqueta con un código QR para acceder a un video de presentación, donde se explican las instrucciones y reglas del juego. En la figura 1 se muestran algunas de las cartas de JAM y el modelo de tablero donde el robot se ha ubicado en la posición fila 6, columna 1.

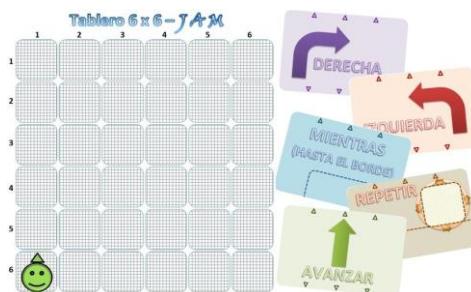


Figura 1. JAM. Cartas y tablero modelo.

Los símbolos que contienen las cartas representan: los 3 movimientos posibles del robot (avanzar un paso hacia adelante, girar a la izquierda 90° y girar a la derecha 90°), y las estructuras de control (repeticIÓN e iteraciÓN). Sobre estas últimas, es

possible observar una línea punteada, la misma se utiliza para trabajar el concepto de bloque de acciones y construcción por indentación, como se puede observar en la figura 2.



Figura 2. JAM. Bloque de acciones

El propósito principal del JAM es promover, a partir de la resolución de problemas (desafíos del juego), el desarrollo de habilidades y competencias necesarias para la construcción de algoritmos, buscando con ello un aporte a la alfabetización digital en el nivel primario. Esto último se alinea a lo propuesto por la autora Marina Bers [12] quien otorga a la experiencia de enseñanza y aprendizaje de la programación con niños de escolaridad primaria o menores (hasta 7 años), la potencialidad de la escritura, al señalar que al igual ésta “*es una nueva forma de expresarnos y crear distintos productos*” y en relación con el proceso agrega “*resolver un problema es una manera de expresar una solución por lo que está enmarcada en una meta más global de creación, la posibilidad de expresar y desarrollar una idea*”. En el caso de JAM el lenguaje es reducido, lo cual puede ser una ventaja para dar los primeros pasos en el proceso de construcción de algoritmos y hacer un aporte al desarrollo del pensamiento computacional.

En cuanto a la dinámica de juego, JAM está pensando con la intención de trabajar en equipo, preferentemente se propone jugarlo con 4 integrantes, divididos en 2 parejas que construyen colaborativamente el algoritmo a medida que pasa la ronda de turnos. Ambos equipos deberán alcanzar un mismo objetivo. La primera pareja en lograrlo gana la partida del juego. Cada objetivo propuesto en el juego, consiste en mover una ficha (robot de nombre JAM) hasta un casillero determinado dentro del tablero. En la figura 3 se presentan algunos de los desafíos que incluye el juego.

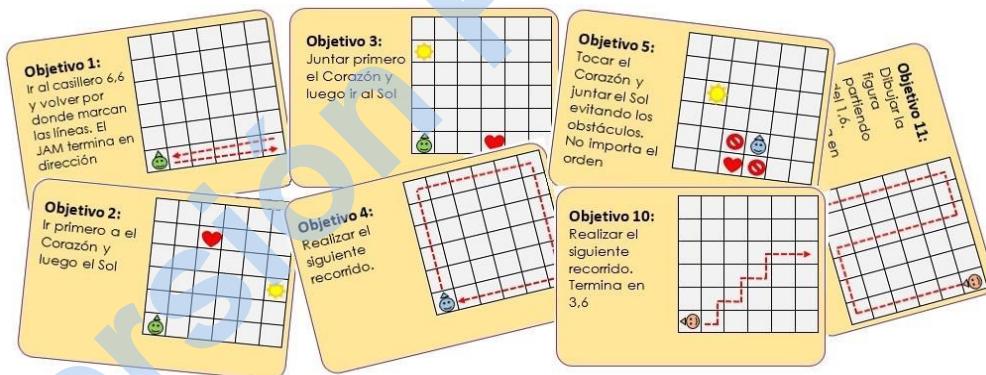


Figura 3. JAM. Tarjetas objetivos

Al comienzo de una partida, cada pareja del equipo dispone de un tablero con su robot, que por lo general se ubica en la esquina inferior izquierda. Como segundo paso, en equipo, eligen un objetivo en común (tarjetas objetivos) o bien pueden idear sus propios objetivos. Luego, se reparten tres cartas por jugador (pareja), dejando una carta destapada cerca del mazo. Cada pareja juega de manera alternada por turnos, y en cada turno es posible tomar y dejar una carta, de manera de tener siempre 3 cartas por pareja. Así, es posible bajar una carta, ya sea para descartarla en el mazo o para usarla en la construcción del algoritmo. Si bien está permitido que el equipo pueda dialogar para analizar una jugada, está prohibido el intercambio de cartas entre las parejas. Una partida finaliza cuando una de las parejas, o ambas, alcanzan el objetivo, dando lugar a la etapa de prueba de la solución. Así, cada pareja pone a prueba su propio algoritmo y el del equipo contrario, para corroborar si efectivamente cumple con el desafío. Se trata de construir una solución con la menor cantidad de instrucciones posibles (cartas), teniendo en cuenta que los caminos para llegar al objetivo deseado pueden ser distintos, todo depende del azar y de la estrategia utilizada para resolver el problema.

JAM presenta 11 desafíos que abarcan, desde los más simples como desplazar al robot desde un punto a otro del tablero (con o sin obstáculos) o juntar un objeto y llegar a otro; hasta otros más complejos, como realizar un recorrido que permita la construcción de una figura determinada. Así también, es posible que un equipo pueda crear libremente desafíos promoviendo de esta forma la creatividad de los jugadores.

## 2. Descripción de la Experiencia

A continuación, se presenta una descripción resumida de la experiencia de uso del juego JAM en diferentes instituciones educativas de nivel primario entre los años 2022 y 2023. Las instituciones se ubican en el ejido urbano y suburbano de la ciudad de San Carlos de Bariloche (Río Negro).

### 2.1. Propuesta Didáctica

La intención de la propuesta didáctica, es una invitación a la formación compartida entre los docentes y estudiantes, teniendo en cuenta que en todos los casos JAM es un juego nuevo para los participantes (docentes/estudiantes) aún cuando tenga similitudes con otros juegos de mesa [13, 14].

En cuanto a los contenidos incluidos en la propuesta, se trabajan: la noción de secuencia de instrucciones, las estructuras de control (iteración condicional y la repetición) y el concepto de construcción de bloques de instrucción a partir de la indentación. A la vez que habilidades como el trabajo colaborativo y cooperativo, y la autoevaluación.

Para la evaluación de la propuesta, se propuso la construcción de un instrumento (cuestionario) ad-hoc con preguntas cerradas y abiertas que permitieran recuperar información de tipo demográfica, experiencias previas con juegos (de diversos tipos), y en particular de la experiencia de juego con JAM. El diseño de este instrumento se presenta en dos versiones: docentes y estudiantes, con las modificaciones necesarias de acuerdo a los roles y edades. Es importante señalar que su construcción estuvo acompañada por los aportes de una especialista en educación de la UNRN Sede Andina. Sumado a esto, se propuso realizar una observación participante en la sesión de juego, por parte del equipo extensionista. Y a posteriori realizar entrevistas semiestructuradas a los docentes participantes con la intención de conocer cuestiones vinculadas al uso posterior del juego (se continuó utilizando, en qué grados, qué impacto se observa en el aprendizaje, etc.).

### 2.2. Contexto

De la experiencia de uso del juego JAM, al momento de presentar este trabajo, han participado seis instituciones educativas de nivel primario de la ciudad de San Carlos de Bariloche (Río Negro), de las cuales tres son escuelas que pertenecen a la gestión pública provincial, dos son colegios privados y uno es un colegio público de gestión privada. Es importante señalar, que tres de las instituciones se ubican en espacios urbanos (dos colegios y una escuela) y el resto en zonas suburbanas. En total, de estas experiencias han participado 200 estudiantes del tercer ciclo de nivel primario y 14 docentes.

De las 3 escuelas participantes, solo en 2 de ellas se aborda el aprendizaje de la programación desde el 6to grado. En el caso de los colegios, en uno lo hacen desde el 3er grado y en los restantes desde el 5to grado. Además en uno de los colegios se trabaja nociones de robótica desde el nivel inicial. Esta información permite inferir que los estudiantes de los colegios podrían tener mayores saberes en relación a la programación que los estudiantes de las escuelas primarias qué participaron de la experiencia. En la Tabla 1, se presenta la información sobre saberes previos de los docentes qué participaron de la experiencia, categorizada por institución.

Tabla 1. Instituciones, grados y docentes participantes.

Institución	Grados participantes	Docentes participantes / Conocimientos de programación
Colegio 1 - privado	5to	- Prof. Inglés con conocimientos de programación - Docente de Matemáticas sin conocimientos
Colegio 2 - privado	6to	- Docente de Matemáticas con conocimientos* - Referente TIC con conocimientos
Colegio 3 - público gestión privada	7mo	- Referente TIC con conocimientos - Docente de Matemáticas con conocimientos* - Docente de Lengua sin conocimientos
Escuela 1 (No abordan contenidos de programación)	7mo	- Docente de Matemáticas con conocimientos - Maestra integradora sin conocimientos

Escuela 2	6to	-Directora con conocimientos -Secretario con conocimientos -Docente de Ciencias Naturales sin conocimientos
Escuela 3	6to	-Prof. de Folclore sin conocimientos -Docente de Inglés sin conocimientos -Docente de Ciencias Naturales con conocimientos

En relación a los saberes previos de los docentes, los datos dan cuenta que en su mayoría, tienen conocimientos de programación, vinculados en general al uso de la programación visual. Sobre la información presentada, es importante, resaltar qué en las experiencias con JAM en los colegios 2 y 3, el docente de matemáticas es el mismo (identificado en la Tabla 1 con el símbolo \*).

### 2.3. La Experiencia

La experiencia de uso del juego JAM estuvo destinada a estudiantes de nivel primario que cursan el último ciclo (de 5to a 7mo grado) y se desarrolló en un espacio de 80 minutos que corresponden a 2 módulos de clases. Durante ese tiempo se llevó adelante la presentación del proyecto de extensión, seguidamente se realizó la exhibición del juego usando una presentación interactiva y una simulación del mismo. Al finalizar, se procedió, con la asistencia de los docentes, a la conformación de equipos entre los niños y niñas presentes en el aula, y se inició el espacio de juego. De este espacio participan además de los estudiantes, el docente responsable del mismo y, en algunos casos, se contó con el acompañamiento del referente TIC, de la maestra integradora, del secretario y del director de la escuela participante. Es importante señalar que cada institución, después de concluida la experiencia, tomó posesión de los kits del juego utilizados, de manera que pudieran seguir experimentando con JAM y compartiendo con el resto de la comunidad educativa.

## 3. Resultados

A continuación se presentan algunos de los resultados obtenidos a partir del análisis del cuestionario realizado a los estudiantes y docentes, y de las entrevistas realizadas a algunos de los docentes.

Respondieron al cuestionario 200 estudiantes, de los cuales 85 pertenecen a escuelas y 115 a colegios. Categorizados por edad, la mayor parte (107 estudiantes) tienen entre 10 y 11 años y el resto (93 estudiantes) se ubican en el rango de 12 a 13 años.

Consultados los estudiantes respecto a los juegos que usan habitualmente, se han obtenido los siguientes resultados: el 55% indica jugar videojuegos, el 18% señala que utiliza juegos de mesa y el 34% indica otro tipo de juegos. Se indagó además acerca de la modalidad de juego que prefieren, aquí el 11% indicó que prefiere jugar en forma individual, el 31% prefiere los juegos en parejas y el 58% en grupo.

Con relación a qué aspectos del juego JAM les resultaron de interés a los estudiantes a partir de la experiencia en el aula, se han obtenido los datos que se presentan en la Tabla 2, categorizados por edad.

Tabla 2. JAM aspectos de interés para los estudiantes categorizados por edad.

Aspecto de interés	Entre 10 y 11 años	Entre 12 y 13 años
La semejanza con otros juegos	4,67%	1,09%
El trabajo en equipo	18,69%	18,48%
Distintos objetivos a cumplir	15,89%	17,39%
El aprendizaje de algoritmos	15,89%	22,83%
El azar en el juego	5,61%	8,70%
La variedad de movimientos del robot	22,43%	20,65%
El material - estética	5,61%	1,09%
En general todo	13,08%	9,78%

Se puede observar como aspecto relevante, la variedad de movimientos del robot, seguido por el trabajo en equipo, esto último en línea con las preferencias de modalidad de juego indicada por los estudiantes en una pregunta precedente. En el caso de los estudiantes más grandes, se destaca el valor del juego para el aprendizaje de algoritmos. Consultados los estudiantes acerca de las dificultades que se les presentaron durante el desarrollo de las partidas del juego, se han obtenido los resultados que se presentan en la Tabla 3, continuando con la categorización por edad.

*Tabla 3.* Dificultades indicadas por los estudiantes al jugar JAM categorizados por edad.

Dificultades	Entre 10 y 11	Entre 12 y 13	Todos
Ninguna dificultad	34,58%	41,11%	37,56%
La ubicación de las cartas de movimiento dentro de las estructuras de control	26,17%	22,22% %	24,37%
Movimientos y estrategias para llegar al Objetivo			
La dinámica del juego para hacer valer el reglamento	25,23%	30% %	27,41%
No poder ganar, no haber tenido suerte	10,28%	11,11%	10,66%

Los resultados indican que quienes menos dificultades encontraron son los estudiantes más grandes, lo cual puede estar asociado a los saberes previos. A partir de los datos recolectados se presenta a continuación una comparación entre dos grupos de 7mo grado, uno perteneciente a la escuela primaria donde al momento de realizar la experiencia no se trabajaba en ningún grado la enseñanza de la programación (Escuela 1) y otro a un colegio donde la programación se aborda desde el 5to grado.

*Tabla 4.* Comparativa dificultades indicadas por los estudiantes al jugar JAM categorizados por institución.

Dificultades	Colegio 3 (aborda la programación desde 5to grado)	Escuela 1 (no aborda la programación)
Ninguna dificultad	27,02%	61,29%
La ubicación de las cartas de movimiento dentro de las estructuras de control	32,43%	16,13%
Movimientos y estrategias para llegar al Objetivo		
La dinámica del juego para hacer valer el reglamento	16,21%	22,58%
No poder ganar, no haber tenido suerte	24,32%	3,22%

Aquí se observa que para quienes menos dificultades presenta el juego, son aquellos estudiantes para los cuales los conceptos de algoritmos qué plantea JAM no son temas que se aborden en las prácticas áulicas. Es interesante observar qué para el grupo de estudiantes con saberes previos en programación, la ubicación de las cartas de movimiento dentro de las estructuras de control (indentación) resulta un desafío, y esto tal vez pueda estar asociado a la forma de llevar adelante esta acción en las herramientas de programación visual.

A continuación, se presentan algunos resultados del cuestionario aplicado a los docentes e información proveniente de las entrevistas realizadas. Estas últimas, se realizaron durante el primer cuatrimestre de 2024. Respecto al cuestionario, se obtuvieron 9 respuestas del total de los docentes que participaron de la experiencia. En cuanto a la antigüedad docente, la mayoría (55,6%) indicó contar con entre 10 a 20 años de antigüedad. Consultados acerca de si utilizan habitualmente juegos durante las clases, la mayoría (66,7%) respondió afirmativamente. En cuanto a la facilidad de uso del juego JAM durante la experiencia en el aula, 6 de los 9 docentes indicaron que necesitaron asistencia para jugarlo. Sin embargo, todos consideraron que el mismo es un aporte al aprendizaje de la lógica de programación y que lo recomendarían a otros colegas.

A continuación se presentan algunos resultados obtenidos a partir de dos entrevistas en profundidad realizadas a docentes de dos colegios privados. Una de las entrevistas corresponde a un docente que cuenta con más de 20 años de antigüedad

en el ejercicio de la práctica docente y una fuerte vinculación en el área de matemáticas, en su propias palabras “*tuve toda una formación sobre lo que es Olimpiada Matemáticas. Actualmente, ya hace unos cinco años, soy también jurado regional de la Olimpiada Mate Clubes.*” y al momento de realizar la experiencia con JAM estaba a cargo de sexto y séptimo grado para Matemáticas y Ciencias Naturales. La otra entrevista, corresponde a una docente que ocupa desde 2018, el rol de referente en innovación digital educativa, y entre otras funciones lleva adelante la capacitación de los docentes del establecimiento para abordar la enseñanza de la programación en todos sus niveles. La misma es Licenciada en Diseño y Comunicación Transmedia y alcanzó previo a ello, el segundo año de Licenciatura en Sistemas en una institución universitaria de la provincia de Córdoba, carrera a través de la cual reconoce haber iniciado con la programación. Sobre su experiencia en la enseñanza de programación a niños, señala qué comenzó utilizando el “*lenguaje Scratch, con algunos kits de robótica, aprendiendo en la marcha.*” y en Bariloche continuó su experiencia docente en el tema en una institución educativa no formal vinculada específicamente a la robótica y la programación. Esta información da cuenta que son docentes con formación y experimentados en relación a la programación.

A través de las entrevistas, se les consultó a estos docentes acerca del impacto que consideran puede tener el conocimiento de lenguajes visuales como Scratch en el uso de JAM. Sobre este punto, la docente entrevistada respondió que JAM en estos casos brinda un aporte al desarrollo de la lógica, invitando a los estudiantes a realizar acciones más reflexivas y controlando la ansiedad (en comparación con el accionar prueba-error que se da en el aprendizaje conectado), donde se pone en valor el trabajo en equipo y la evaluación por pares. En palabras de la docente: “*les ayuda un poquito más a reflexionar previamente antes de pasar simplemente a poner los bloques. A veces pasa, por la ansiedad misma del alumno, decir ya lo quiero terminar y listo..., pero el juego lo lleva a que obviamente tengan que pensar en hacerlo bien, que cuentan con su compañero, que el otro equipo también te está controlando si está bien lo que están haciendo, entonces como que te lleva a una reflexión previa que creo que está bueno y los ayuda.*”

Por otra parte, el docente entrevistado valoró el apoyo de JAM para trabajar en forma tangible y concreta con estudiantes que ya tienen incorporada la lógica de la programación y señaló “*...el tema de que sea palpable la carta. Sexto o séptimo grado, es ideal, porque todavía uno se da cuenta que necesitan siempre lo concreto y el poder mover con sus propias manos las cartas y acomodarlas.*” inclusive agregó que estudiantes que habían jugado JAM en la experiencia 2022, al retomar el uso del juego un año después, lejos de desmotivarse “*te pasaban el trapo, no había forma de ganarles porque habían encontrado muchísimas formas de allanar el camino y con muchísimo menos cartas poder hacer los mismos recorridos*”.

Consultados ambos entrevistados respecto a si han observado diferencias de género durante las instancias de juego con JAM, la docente señaló que si bien no hay diferencias en el proceso de resolución de los desafíos y en cuanto al efecto competencia, las niñas son más ordenadas e incluso más tranquilas, lo cual ayuda al proceso de construcción del algoritmo.

En relación con la apropiación del juego JAM, uno de los entrevistados, que al momento de la experiencia tenía actividades de docencia y al momento de la entrevista se encontraba en la función de director y vicedirector en dos de las instituciones donde se llevó adelante la experiencia, señaló que a partir de charlas con los docentes de grado de esas instituciones, decidieron utilizar JAM como recurso para avanzar con actividades de programación en diferentes cursos, tanto en la actividad escolar como en talleres extraescolares. Y destaca el valor del trabajo colaborativo en el proceso de inclusión de JAM en las propuestas áulicas, sobre todo con los docentes para los cuales el juego resulta novedoso. Consultado acerca de las estrategias que les permiten mantener la motivación, además de la dosificación del tiempo y frecuencia de uso de JAM, el docente señaló haber implementado la modalidad de juego denominada “*Todos contra el profesor*” donde como expresa el docente “*Les dejaba un desafío y ellos tenían que resolver el algoritmo. Si perdían se les descontaban minutos de recreo, entonces llegó un momento en que siempre me ganaban, me di cuenta de la evolución del grupo*”.

Consultado el docente acerca de cuál es la estrategia de enseñanza usando JAM, señaló que con los estudiantes más pequeños (tercer grado) utilizan una estructura de control por vez, comenzando por la carta de repetición, luego por la de iteración y por último presentan la carta de selección, de manera que puedan comprender de cada una la funcionalidad y aportes al proceso de resolución de desafíos que les presenta el juego. Así mismo, el docente señaló que uno de los aspectos complejos que presenta el JAM es la delimitación de bloques de código usando la indentación.

En cuanto al diseño de los materiales del juego, el mismo docente indicó que las cartas “*podrían ser más gruesas*” a la vez que resaltó el cuidado de los estudiantes con la manipulación de los componentes del juego.

Por lo expuesto, es posible señalar que el JAM ha sido bien receptionado, tanto por los estudiantes como por los docentes que han participado de la experiencia. En el caso de los docentes, y a partir de las entrevistas, es posible reconocer una apropiación del mismo como herramienta para dar los primeros pasos en la construcción de algoritmo y/o como recurso para el repaso de conceptos trabajados en propuestas didácticas conectadas, promoviendo un espacio de trabajo más reflexivo.

## 4. Conclusiones

El objetivo de este trabajo es presentar en forma resumida un juego de mesa denominado JAM y algunos resultados incipientes alcanzados a partir del desarrollo de experiencias áulicas para trabajar con el mismo en diferentes instituciones educativas de nivel primario de la ciudad de San Carlos de Bariloche. En el caso de los estudiantes que han participado de la experiencia, son niños cuyas edades oscilan entre los 10 y 13 años y transitan el tercer ciclo de la escolaridad primaria. En el caso de los docentes entrevistados, ambos cuentan con experiencia en la enseñanza de la programación y lógica tanto en forma conectada como desconectada.

Si bien el juego ha sido receptionado positivamente tanto por los estudiantes, quienes destacan como positivo la posibilidad de jugar en equipos con independencia de la edad, como por los docentes para quienes el recurso se presenta utilidad tanto para ser incluido en actividades educativas de repaso como de evaluación de saberes vinculados a la programación, a futuro, se espera poder avanzar con la implementación de la experiencia en otras instituciones educativas de nivel primario de la provincia de Río Negro, principalmente en aquellas donde los estudiantes y/o docentes no han tenido experiencias previas de aprendizaje de la programación en forma conectada. Asimismo, se propone avanzar en el diseño, planificación y evaluación del impacto de experiencias de aprendizaje usando JAM destinada a estudiantes de formación docente principalmente en institutos de formación docente del contexto en el que se inserta la UNRN, tomando como base investigaciones previas realizadas en el contexto nacional y regional [14,15]

En cuanto a las orientaciones para llevar adelante una experiencia como la que se presenta en este trabajo, consideramos que resulta fundamental contar con el acompañamiento institucional y el deseo de innovar en las prácticas docentes, en palabras de uno de los docentes entrevistados “*A veces los directivos están interesados por una propuesta nueva innovadora, pero si el profe o la maestra no están enganchados la propuesta no prospera, salvo en instituciones que se pongan muy firmes.... A veces sucede que el camino se da entre colegas.*” . Asimismo, y como señaló otro docente, es importante sobretodo con los niños más pequeños o con aquellos que no tienen ningún conocimiento previo, avanzar en forma incremental utilizando de a una estructura de control por vez (repetición iteración y por último la selección).

## 5. Agradecimientos

A las instituciones educativas qué han participado de la experiencia y a los docentes que nos brindaron su tiempo para las entrevistas. A los colegas docentes de la UNRN Sede Andina que han contribuido con su conocimiento para el desarrollo de los instrumentos de evaluación.

## Referencias

- [1] J. M. Wing. (2006). “Computational thinking” Communications of the ACM, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- [2] E.A. Vázquez Uscanga, J. Bottamedi, M.L. Brizuela. "Pensamiento computacional en el aula: el desafío en los sistemas educativos de Latinoamérica." RiiTE Revista Interuniversitaria de Investigación en Tecnología Educativa, Nro. 7, (2019).
- [3] J. A. C. Molina. “Lenguaje de programación para niños y niñas: perspectivas conectadas y desconectadas en la educación básica”. Revista Internacional de Pedagogía e Innovación Educativa, Vol. 3 Nro.1, pp.,45-66. 2023
- [4] Casali, Ana, et al. "Pensamiento computacional y programación en la formación de docentes del nivel primario." XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste).. 2018.
- [5] Huang, W. and Looi, C. (2021). A critical review of literature on "unplugged" pedagogies in K-12 computer science and computational thinking education. Computer Science Education, 31(1), 83-111.
- [6] Hufad, A., Faturrohman, M. and Rusdiyani, I. (2021). Unplugged Coding Activities for Early Childhood Problem-Solving Skills. Journal Pendidikan Usia Dini, 15(1), 121-140.
- [7] Carmona-Mesa, J. A., & Cardona, M. (2019). Formación en el Pensamiento Computacional a través de juegos de mesa. In XV Conferencia Interamericana de Educación Matemática, 1-8.
- [8] R. Bilbao, A. Rivas. “Las provincias y las TIC: avances y dilemas de política educativa. Documento de trabajo”, vol. 76, 2011
- [9] Iuri, T. “Educación Secundaria Rural Virtual en Río Negro, Argentina: una experiencia de más de diez años”. Praxis educativa, Vol. 26 Nro 3, pp., 77-98, 2022
- [10] Depetris, B., Aguil Mallea, D., Pendenti, H., Tejero, G., Feierherd, G. E., y Prisching, G. "La enseñanza y el aprendizaje de la programación y la programación concurrente con DaVinci Concurrente." X Congreso sobre Tecnología en Educación & Educación en Tecnología (TE & ET)(Corrientes, 2015). 2015.

- [11] <https://scratch.mit.edu/>
- [12] Bers, M. "Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom. New York, USA: Routledge. 2017
- [13] Bogliolo, A., "Unplugged language-neutral card games as an inclusive instrument to develop computational thinking skills". In INTED2015 Proceedings (pp. 7609-7615). 2015, IATED.
- [14] Gresse von Wangenheim, C., Araújo e Silva de Medeiros, G., Missfeldt Filho, R., Petri, G., da Cruz Pinheiro, F., Ferreira, M. N., & Hauck, J. C. (2019). SplashCode--A Board Game for Learning an Understanding of Algorithms in Middle School. *Informatics in Education*, 18(2), 259-280.
- [15] Bavera, F., Daniele, M., Quintero, T., & Buffarini, F. (2019). Habilidades de Pensamiento Computacional en docentes de primaria: evaluación usando Bebras. In XXV Congreso Argentino de Ciencias de la Computación (CACIC)(Universidad Nacional de Río Cuarto, Córdoba, 14 al 18 de octubre de 2019).

Versión Preliminar

# Iniciación a la programación en contextos de encierro con Arduino

Daniela Macario Cabral<sup>1</sup>, Christian Cossio-Mercado<sup>1</sup> y Gonzalo Pablo Fernández<sup>2</sup>

<sup>1</sup> Universidad de Buenos Aires. Facultad de Ciencias Exactas y Naturales. Departamento de Computación. Buenos Aires, Argentina.

<sup>2</sup> Universidad Nacional de Quilmes. Departamento de Ciencia y Tecnología. Buenos Aires, Argentina.

mmacario@dc.uba.ar, ccossio@dc.uba.ar, gpfernandez@dc.uba.ar

## Resumen

Con el fin de garantizar el derecho a la educación superior para las personas privadas de su libertad, la Universidad de Buenos Aires desarrolló el programa UBAXXII, a través del cual brinda la posibilidad de cursar carreras de grado y cursos de formación profesional a internos de algunas unidades penitenciarias del país. Dentro del marco de este programa se diseñó un curso introductorio de programación usando placas Arduino y programación por bloques, realizado en dos etapas. Los conceptos elementales se dictaron con el uso de la plataforma Pilas Bloques y la puesta en práctica se implementó con el uso de placas Arduino y la herramienta Arduino en la Escuela. Aunque se decidió usar la robótica como herramienta tangible que permite promover la participación y la motivación de los estudiantes, el objetivo del curso es la resolución de problemas basada en el pensamiento computacional y la programación por bloques. Ambas etapas del curso recibieron buenas evaluaciones por parte de los participantes al final de su dictado. Además, se pudo notar que una adecuada gestión de los materiales contribuyó con el compromiso de los estudiantes con sus proyectos y mejoró su rendimiento. Se espera que esta experiencia sea de interés y utilidad para docentes que quieran implementar nuevas estrategias para la enseñanza de la programación en contextos de vulnerabilidad.

**Palabras clave:** Contexto de encierro; Programación; Robótica; Arduino; Personas privadas de la libertad

## 1. Introducción

El acceso a la educación es considerado un derecho humano. En particular, la educación en contextos de encierro contribuye significativamente a la reinserción social de los internos [1]. Además, a pesar de las limitaciones y los desafíos que implica el contexto, la enseñanza de cursos universitarios en prisiones puede ser exitosa y tener un impacto positivo en la vida de los estudiantes [2]. El Sistema Nacional de Estadística sobre Ejecución de la Pena de la República Argentina reporta en su último informe que, de 105.053 personas privadas de su libertad, sólo 384 alcanzaron a completar el nivel universitario, 26.779 tienen incompleto el nivel secundario y 35.292 tienen completo el nivel primario, habiendo además 23.545 personas con nivel primario incompleto [3]. Por otro lado, la ley 26.695, promulgada en el año 2011, establece el derecho a la educación pública para las personas privadas de la libertad y la responsabilidad indelegable del estado de proveer prioritariamente una educación integral, permanente y de calidad. Además, en ella se crea un régimen que pretende estimular el interés de los internos por el estudio al permitirles avanzar en forma anticipada en el régimen progresivo de ejecución de la pena, a partir de sus logros académicos [4].

Para garantizar el derecho humano a la educación, algunas universidades nacionales llevan adelante programas de integración universitaria en unidades penitenciarias. Algunos proyectos de educación superior en cárceles incluyen a la Universidad Nacional de La Plata, con su Dirección de Acompañamiento Universitario en Cárcel, la Universidad Nacional del Litoral con su Programa de Educación Universitaria en Prisiones, la Universidad Nacional de Córdoba con su Programa Universitario en la Cárcel, la Universidad Nacional de San Martín con su Centro Universitario San Martín y la Universidad de Buenos Aires con su programa UBAXXII. El programa UBAXXII tiene su origen en el año 1985 con el objetivo de garantizar el acceso a la educación superior a personas privadas de su libertad. Tiene lugar en el Complejo Penitenciario Federal de la CABA y en los Complejos Penitenciarios Federales 1 y 4 de Ezeiza, entre otros [5, 6]. Varias unidades académicas participan en el programa: el Ciclo Básico Común, la Facultad de Ciencias Económicas, la Facultad de Ciencias Exactas y Naturales, Facultad de Derecho, Facultad de Filosofía y Letras, Facultad de Psicología y Facultad de Ciencias Sociales.

En particular, la Facultad de Ciencias Exactas y Naturales participa de este programa desde el año 1991 a través del dictado de talleres de formación profesional. Estos talleres suelen estar dirigidos al uso de las computadoras, ya que el uso básico de una computadora y de programas de edición de texto y hojas de cálculo son temas de enseñanza con alta demanda. Entre otras cosas, se enseña el uso de herramientas para diseño gráfico, edición de video y Python como lenguaje de programación y herramienta para análisis de datos. Además, se dicta una diplomatura de iniciación a la programación y análisis de datos.

### 1.1. Sobre el curso

En este marco, a partir del año 2023 se tomó la decisión de impartir formaciones en programación utilizando placas Arduino<sup>1</sup>. Se propuso un curso piloto separado en dos etapas cuatrimestrales en el que se introdujeran los conceptos

<sup>1</sup>Un microcontrolador de hardware abierto utilizado para el desarrollo de proyectos de robótica en diversos contextos[7].

fundamentales de la programación y el pensamiento computacional, con el uso de la robótica como herramienta pedagógica.

En particular, se hizo énfasis en el trabajo con pensamiento computacional, en tanto este engloba diversas competencias como la identificación de patrones, la descomposición de problemas en subproblemas, el pensamiento abstracto y el diseño de algoritmos, entre otras [8], que desde su aparición ha sido ampliamente usado y citado en educación. Más allá de que la definición del término y su alcance no está libre de discusiones [9, 10], creemos que los aprendizajes que implica son muy relevantes.

Por otro lado, una de las herramientas más elegidas para la enseñanza de conceptos de computación es la robótica. En particular, el uso de placas Arduino, gracias a su diseño de hardware abierto y a la amplia variedad de recursos disponibles, presenta múltiples ventajas en la enseñanza de la programación ya que permite programar comportamientos con efectos en el mundo físico. Este aspecto facilita el desarrollo del pensamiento computacional [11] e incrementa la motivación y el trabajo creativo al permitir resolver problemas de la vida cotidiana [12]. Finalmente, su uso promueve el trabajo con proyectos y, en particular, la construcción y prototipado de pruebas de concepto que, de otra forma, requerirían un entendimiento técnico mucho más profundo.

En este trabajo se describe la experiencia en el desarrollo de dos cursos de programación de placas Arduino para enseñar conceptos fundamentales de la programación en contextos de encierro. Los conceptos elementales se introdujeron con Pilas Bloques, una plataforma de programación por bloques con una propuesta pedagógica basada en el aprendizaje basado en la indagación, la descomposición, los desafíos cortos y la construcción de abstracciones [13]. Además, se utiliza Arduino en la Escuela [14], un entorno de programación basada en bloques para Arduino, desarrollado para su utilización en ámbitos educativos, caracterizado por su portabilidad y su utilidad para la enseñanza de la programación, especialmente en contextos de baja disponibilidad de recursos [15].

## 2. Descripción de la experiencia

A lo largo del año 2023 se dictaron dos cursos de duración cuatrimestral, uno como continuación del otro, que se desarrollaron en los centros universitarios de Ezeiza (CUE) y Devoto (CUD). Las clases fueron semanales y tuvieron 2 horas de duración. El temario de los talleres fue el mismo en ambos centros, aunque con adaptaciones específicas derivadas de la experiencia con cada lugar.

### 2.1. Contexto

Los cursos se dictaron como talleres de formación profesional de la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires (FCEyN-UBA), dentro del marco del programa UBAXXII de educación en cárceles. Se dictaron simultáneamente en el Centro Universitario del Complejo Penitenciario Federal 1 de Ezeiza (comúnmente llamado CUE) y en el Centro Universitario del Complejo Penitenciario Federal de la CABA en Devoto (comúnmente llamado CUD). Ambos complejos están habitados por hombres adultos.

#### 2.1.1. Participantes

El público objetivo del curso fue la población interna del penal que estuviera en condiciones de acceder al centro universitario. Como no existen requisitos académicos para cursar talleres de formación profesional, esto deriva en que se puede contar con participantes con una carrera universitaria así como otros que aún no completaron el nivel primario. En ambos complejos se debe otorgar al interno una autorización especial para cada día de cursada en Centro Universitario (habitualmente llamada *bajada*). Es común que los internos tengan problemas para acceder a su *bajada* cuando ocurren eventos en su residencia, como requisas, visitas, turnos médicos, entrega de alimentos o asistencia a instancias judiciales. En estos casos se habla de un *corte de la bajada*.

Por lo expuesto en el punto anterior, es importante considerar la residencia de cada asistente al curso. Cada complejo penitenciario está compuesto por módulos de residencias con distintos niveles de seguridad. En el caso del complejo de Ezeiza, los módulos residenciales de procedencia de la mayoría de los estudiantes de este curso eran de mínima o media seguridad. Esto es así porque en los módulos de alta seguridad el acceso a la *bajada* puede ser más difícil, tanto por cuestiones de conducta o por otros criterios aplicados por el Servicio Penitenciario que los docentes desconocen. En el caso de Devoto, las residencias de origen fueron heterogéneas, por lo que hubo estudiantes provenientes de módulos residenciales de mínima, media y alta seguridad. En esta sede los problemas referidos al *corte de la bajada* suelen tener que ver con cuestiones administrativas que son desconocidas para los docentes. En líneas generales, además, es más complicado para un interno del penal de Ezeiza acceder al CUE que para un interno del penal de Devoto acceder al CUD.

Respecto a las edades de los participantes<sup>2</sup>, en Ezeiza participó alrededor de un 50 % de mayores de 60 años y de 20 % de menores de 30 años. Por otro lado, en Devoto, para el primer cuatrimestre alrededor del 60 % de estudiantes estaba en la franja de 30 a 40 años y cerca de un 20 % de menores de 30. En el segundo cuatrimestre hubo un 65 % de estudiantes en la franja de 30 a 40 años y cerca de un 10 % de menores de 30.

El nivel educativo de los participantes en ambas sedes fue heterogéneo, ya que había desde estudiantes de primaria a profesionales universitarios. Por último, en lo respectivo a la asistencia, es de destacar que, si bien el acceso a la *bajada* puede ser un problema en sí mismo, la asistencia a los talleres también está sujeta a decisiones personales de cada interno.

<sup>2</sup>Los porcentajes son estimaciones de la docente, ya que no fue posible acceder a esa información con precisión.

### **2.1.2. Centro Universitario de Ezeiza (CUE)**

El CUE es un espacio pequeño en el que conviven varias facultades de la UBA. Sus estudiantes se caracterizan por el compromiso con el espacio y abogan por el bienestar de sus compañeros. Además, son muy aplicados en su estudio: en general, si un estudiante recibe su *bajada*, asiste a la clase. El curso se dictó en el laboratorio de informática, equipado con 15 computadoras donadas por la Fac. de Ciencias Exactas y Naturales (FCEN) de la UBA. El espacio es pequeño pero las mesas están dispuestas contra las paredes del aula, por lo que queda un pequeño espacio central. La docente puede proyectar diapositivas en una de las paredes del aula y hacer uso de una pizarra.

### **2.1.3. Centro Universitario de Devoto (CUD)**

El CUD es un espacio más amplio que el CUE en superficie. Los estudiantes cuentan con más libertades dentro del espacio y se caracterizan por ser más dispersos: si un estudiante recibe su bajada, puede que no asista a clases por otros motivos. El curso se dictó en el aula de informática, equipado con quince computadoras donadas por la FCEN. El aula es muy amplia. Las mesas están dispuestas en filas de tres escritorios, una detrás de la otra, con un pizarrón en el frente. Hay un espacio libre para colocar sillas y mesas adicionales. La docente da la clase desde el frente y puede proyectar diapositivas sobre el pizarrón.

## **2.2. Propuesta didáctica**

Para el diseño del curso se tomó como base un marco didáctico-pedagógico integral para la enseñanza de la programación, el pensamiento computacional y otros conceptos de las Ciencias de la Computación llamado PRENDER. Esta propuesta incluye definiciones para la planificación y seguimiento de actividades de clase, enfoques para el acercamiento al conocimiento, por medio de la indagación y el trabajo con problemas, la organización de cursos y un enfoque de la evaluación como una instancia más de enseñanza [16].

### **2.2.1. Objetivo del curso**

Si bien los cursos están enfocados en el uso de herramientas de robótica, el objetivo final es el aprendizaje de conceptos fundamentales de programación. Se espera que los estudiantes lleguen al final de la primera etapa habiendo podido aplicar los conceptos aprendidos en el diseño de un sistema sencillo de comportamiento automático utilizando un sensor y una respuesta básica, programado con bloques. Para el final de la segunda etapa, se espera que los estudiantes hayan podido avanzar en la complejidad programando el comportamiento de un robot, lo que incluye el diseño, el diagrama y la ejecución de un proyecto más elaborado que el realizado como proyecto final durante la primera etapa, usando la misma herramienta de programación por bloques.

### **2.2.2. Herramientas utilizadas**

A lo largo del curso se utilizaron distintas herramientas de software, como Pilas Bloques y AelE, y de hardware, como computadoras, placas Arduino y componentes electrónicos compatibles. Además, se utilizó ampliamente el pizarrón para las puestas en común de las actividades propuestas, entre otras cosas. Respecto a las computadoras, se trata de equipos poco potentes que cuentan con Windows 10 instalado y están conectadas a una red del Servicio Penitenciario Federal. Así, tienen acceso a internet limitado por bloqueos de seguridad a sitios no autorizados, y los docentes deben solicitar la autorización a los sitios a los que necesitan acceder cada cuatrimestre. Además, no tienen permisos de administrador, por lo que se dispone de un técnico que se encarga de reparar los equipos e instalar el software necesario.

Los materiales electrónicos se repartieron según los requerimientos de cada práctica durante la primera etapa, mientras que se entregaron en forma de kit grupal durante la segunda etapa. Con respecto al kit de materiales, el mismo se compone de una placa Arduino UNO con su correspondiente cable USB, un buzzer activo, un breadboard de 170 puntos, un breadboard de 400 puntos, una fuente para breadboard, diez leds de 5mm de varios colores, dos leds RGB de 5mm, un módulo joystick, un módulo RFID, un potenciómetro lineal de 10k, diez pulsadores, un sensor de humedad de suelo, un sensor infrarrojo, un sensor ultrasónico, un acelerómetro giróscopo de 9 ejes, una matriz de leds 8x8, un mini speaker, un módulo semáforo, un sensor de movimiento, un servomotor y un paquete de 60 cables con ficha dupont.

### **2.2.3. Diseño del curso**

El curso completo se organizó en dos etapas fundamentales, cada una de duración cuatrimestral.<sup>3</sup>

#### **Primera etapa**

Abarcó el primer cuatrimestre y consistió de una introducción a los alcances del taller, cinco módulos de aprendizaje (divididos en 10 a 13 clases según la comisión) y un proyecto final con aplicación de los conceptos aprendidos en los módulos anteriores. El enfoque de las clases fue mayoritariamente práctico, mientras que el contenido teórico se dictó mediante actividades lúdicas o espacios de discusión para construir los conceptos en forma comunitaria.

<sup>3</sup>El diseño completo del curso está disponible en <https://bit.ly/xxii>.

En la introducción se fomentó un espacio de discusión alrededor de la idea de “qué es un robot” y los alcances de la robótica como herramienta científica y de trabajo. Además se plantearon ejemplos del uso de la robótica en la cotidianidad. En el primer módulo se introdujo el concepto de Pensamiento Computacional y se discutieron ideas personales respecto de los conceptos de Algoritmo y Programa. Además se presentó la herramienta Pilas Bloques, con la cual se descubrieron los conceptos de comando primitivo, procedimiento, repetición simple y alternativa condicional. Para las actividades prácticas de este módulo se propuso comenzar con la actividad del autómata humano en la que la docente cumplió el rol de “autómata” y se le debieron dar instrucciones específicas para que realizara la tarea de salir del aula. Luego se realizaron actividades en la plataforma Pilas Bloques para fijar los conceptos aprendidos. Además, se explicó el concepto de búsqueda binaria mediante una actividad en pizarrón de adivinar un número.

En el segundo módulo se presentó la placa Arduino y la plataforma AeIE, con un ejercicio básico de demostración participativa (un blinker). Se mostró además un simulador de placas arduino para enseñar su composición y se explicaron algunos algoritmos de ordenamiento mediante un juego en pizarrón consistente en ordenar listas de números. Los dos primeros módulos fueron diseñados para no usar material electrónico en sus clases, considerando las demoras habituales al gestionar la aprobación de su ingreso al penal.

Para el tercer módulo se introdujeron conceptos básicos de electrónica con el fin de promover el cuidado de las placas Arduino y sus componentes. Se implementaron dos ejercicios prácticos para entender la función de una fuente de energía, una resistencia y la puesta a tierra. En el primero se pidió encender un diodo haciendo uso de un LED de color, una resistencia y un breadboard con fuente, mientras que el segundo consistió en modificar lo anterior, sumando un potenciómetro lineal para variar su intensidad.

En el cuarto módulo se introdujeron los conceptos de sensor y actuador y se los relacionó con los sentidos y las extremidades humanas. Se programaron comportamientos con varios sensores y actuadores por separado y se discutió su utilidad y uso en la vida cotidiana. Se trabajó con un semáforo (primero, usando tres diodos, una resistencia, un breadboard y una placa Arduino UNO, luego reemplazando los diodos por un módulo semáforo y finalmente reemplazando el módulo semáforo por un LED RGB), la lectura de sensores (primero el ultrasónico y luego el LDR) a través del monitor serial y, por último, con el uso de actuadores (primero el buzzer activo y luego el servomotor).

Para el quinto módulo se introdujo el concepto de comportamiento automático y se programaron artefactos que requirieron combinar el uso de sensores y actuadores, a partir del planteo de problemas aplicables a la vida cotidiana. Las actividades propuestas consistieron en una luz que se enciende cuando hay oscuridad (usando diodos LED, un LDR y la placa Arduino UNO) y una alarma de proximidad (usando un LED RGB o un buzzer activo, un sensor ultrasónico y la placa Arduino UNO). Para ambas actividades fue necesario además usar un breadboard de 400 puntos, conectores y resistencias, en caso que fuera necesario usar diodos.

En el proyecto final se dio a elegir a cada grupo entre varios proyectos prediseñados, donde se planteaba una problemática para la cual debían encontrar una solución. La lista de proyectos incluía una alarma para casa (usando LEDs, un buzzer activo y un sensor de proximidad), un analizador de humedad para macetas (usando un display LCD de 16x2, LEDs y un sensor de humedad del suelo), una pequeña estación meteorológica (usando un display LCD de 16x2 y un sensor de temperatura, humedad y presión), una cerradura electrónica (usando un servomotor y un módulo RFID), un tacho de basura automático (usando un sensor ultrasónico y un servomotor), unos ojos biónicos (usando dos servomotores y un acelerómetro) y un juego de alambre y zumbador (usando alambres de cobre, LEDs y un buzzer). Todos los proyectos debieron usar una placa Arduino UNO, cables y breadboard de 170 o 400 puntos. Además, según los proyectos elegidos se usaron elementos de librería para diseñar los contenedores de los proyectos.

## Segunda etapa

Se desarrolló a lo largo del segundo cuatrimestre y consistió en un taller libre en el que los estudiantes en grupo debieron desarrollar un proyecto y trabajar en él a lo largo del cuatrimestre con la guía de la docente. Se introdujeron nuevos conceptos como pseudocódigo, variables locales y globales y se conversó sobre los distintos roles que hay en un equipo de trabajo en la industria del software, sugiriendo aplicar algunos de estos roles a la dinámica del trabajo en equipo para optimizar el tiempo de clase. La metodología de esta etapa fue puramente práctica y el proyecto propuesto fue lo suficientemente amplio como para que cada grupo pudiera desarrollar sus propias ideas. La guía personalizada propuesta por la docente consistió, con las diferencias propias de cada proyecto, en dividir el trabajo en varias etapas. En primer lugar, luego de la conformación de los grupos, se propuso una instancia de decisión, consistente en una discusión grupal de la idea del proyecto a realizar, un boceto del diagrama del comportamiento de la mascota (un esquema de cuadros con flechas que detalle estímulos recibidos y reacción a esos estímulos) y la identificación de los sensores y actuadores que debían usarse para ese fin. Luego, se propuso que cada grupo asignara roles de responsabilidad, para que pudieran separar las tareas de diseño del circuito, programación y diseño visual del robot. A continuación de esta etapa, se propuso el trabajo sobre el pseudocódigo, haciendo uso del diagrama de comportamientos previamente trabajado, para luego pasar a la “traducción” del pseudocódigo a bloques en AeIE. En paralelo se propuso, además, el diseño visual del proyecto y el testeo de los componentes a utilizar y las pruebas de los comportamientos por separado. Con estos temas resueltos, se propuso el armado definitivo del circuito del proyecto y pruebas con el código, para finalmente integrar lo armado al diseño visual elegido.

## **2.2.4. Evaluación y aprobación del curso**

Desde un punto de vista general de la evaluación, pensándola como una instancia más de aprendizaje, a lo largo del curso se desarrollaron actividades de evaluación continua, por medio de comentarios breves y realimentaciones durante el trabajo de laboratorio con las computadoras. Adicionalmente, se hizo uso intensivo de las puestas en común de la solución a cada problema, tanto para que cada grupo contara qué era lo que había hecho, como para que la docente pudiera dar algunos lineamientos de la solución que se esperaba. En ambas etapas se evaluó diariamente el trabajo manual (con los componentes electrónicos) y en computadora (actividades de programación) de los grupos. Durante la segunda etapa, al tratarse de proyectos personalizados, se reservó un espacio de 15 minutos al final de cada clase para que cada grupo pudiera contar a los demás el estado de su proyecto y opinar o hacer aportes sobre otros.

En lo respectivo a la aprobación del curso, al tratarse de un taller experimental y con el fin de promover la participación, el criterio se basó en la observación sobre el trabajo grupal en las actividades prácticas y la participación individual en las actividades de discusión. Un estudiante aprobaría el taller si se registraba su participación y su trabajo grupal en más de la mitad de las clases asistidas.

## **2.3. La experiencia**

### **Primera etapa**

Cada clase contó con el apoyo visual de diapositivas proyectadas para guiar su estructura. Se fomentaron espacios de discusión y prácticas grupales con los materiales electrónicos. Se propuso una lista de proyectos finales para trabajar en las últimas clases del curso, donde se planteó una lista de problemas a resolver con el uso del comportamiento automático. Así, cada grupo podía elegir el proyecto de su gusto o inventar el propio. En este último caso, se planteó que ellos definieran los materiales electrónicos a utilizar.

En ambos centros los materiales se repartieron según los objetivos de cada clase a grupos de 4 o 5 personas. Sin embargo, por cuestiones de espacio disponible, hubo una diferencia respecto de la implementación de los métodos en ambos centros universitarios. En el CUE cada grupo debió trabajar en su mesa y, por la disposición del aula, fue fácil para la docente observar el trabajo de cada grupo y acercarse a responder consultas. En el CUD, si bien al principio se aplicó la misma metodología, la disposición del aula limitaba la participación de los estudiantes de las últimas filas. Para resolverlo se armó una mesa grande en el espacio libre del aula en la que todos trabajaron a la par de la docente. Luego, con el proyecto físicamente armado, volvían a sus mesas para trabajar con el entorno de programación por bloques. Esta nueva disposición también se implementó para las actividades de discusión.

### **Segunda etapa**

Se comenzó con un repaso general de lo visto el cuatrimestre anterior. Aunque no se planificó impartir nuevos contenidos teóricos, se enseñaron conceptos nuevos a cada grupo según las necesidades que les surgieran. Cada equipo recibió un kit de elementos de robótica, identificable por un color en particular, y sólo podía ser usado por ellos. Para comenzar se dedicaron dos clases al reconocimiento y testeо de cada material y a realizar un inventario del mismo. Al final de cada clase, cada grupo debía reportar el estado en una hoja provista por la docente, haciéndose responsable por el mismo.

El proyecto propuesto fue el diseño, armado y programación de una mascota robot. Se definió este proyecto debido a su versatilidad: cada grupo debía definir cómo sería su mascota robot ideal y cómo debía comportarse. Se pidió que cada grupo definiera al menos un comportamiento para su mascota y que lo esquematizara identificando qué sensores y actuadores sería necesario utilizar para simularlos. De esta forma, se trabajó en pizarrón el concepto de pseudocódigo como paso previo a la programación en AelE, y cada grupo debió programar los comportamientos de sus mascotas de la misma manera.

Si bien cada grupo contaba con total libertad creativa, sólo limitada por los componentes del kit asignado, se diseñó una guía personalizada para cada grupo, y al final de cada clase los grupos debían compartir sus ideas y avances. Complementariamente, se sugirió la asignación de roles para optimizar el tiempo de trabajo.

## **2.4. Documentación de la experiencia**

### **2.4.1. Satisfacción de los participantes con el curso**

Al final de ambas etapas del curso se pidió que los participantes completaran una encuesta anónima de valoración del curso, de las cuales se listan las preguntas más relevantes para este trabajo:

- ¿Cuál fue tu nivel de experiencia previa en robótica, programación y pensamiento computacional antes de comenzar el taller? (Nada, poca, algo, bastante, ya era un experto)
- ¿Consideras que los contenidos del taller fueron adecuados para tu nivel de conocimiento? (Sí, No, No estoy seguro)
- ¿Cómo calificarías la herramienta “Arduino en la Escuela” para aprender programación por bloques? (Excelente, buena, regular, mala)
- ¿Cuál fue tu impresión general sobre las prácticas de robótica que realizamos con Arduino? (Me encantaron, me gustaron, estuvieron bien, no me gustaron)
- ¿Qué tan útiles te resultaron las clases teóricas sobre robótica, programación y pensamiento computacional? (Muy útiles, útiles, algo útiles, poco útiles, nada útiles)

- ¿Tuviste la oportunidad de aplicar los conocimientos adquiridos en el taller en tu proyecto final de robótica? (Sí, pude aplicarlos completamente; Sí, pude aplicarlos en parte; No pude aplicarlos; No realicé el proyecto final)

En resumen, las encuestas consultan sobre el nivel de experiencia previa en robótica, programación y pensamiento computacional antes de comenzar el taller. También plantean una evaluación sobre el entendimiento de los contenidos vistos, el disfrute de las actividades prácticas, el trabajo de la docente y la utilidad de los conocimientos adquiridos al momento de realizar los proyectos integradores, así como de la herramienta de software Arduino en la Escuela.

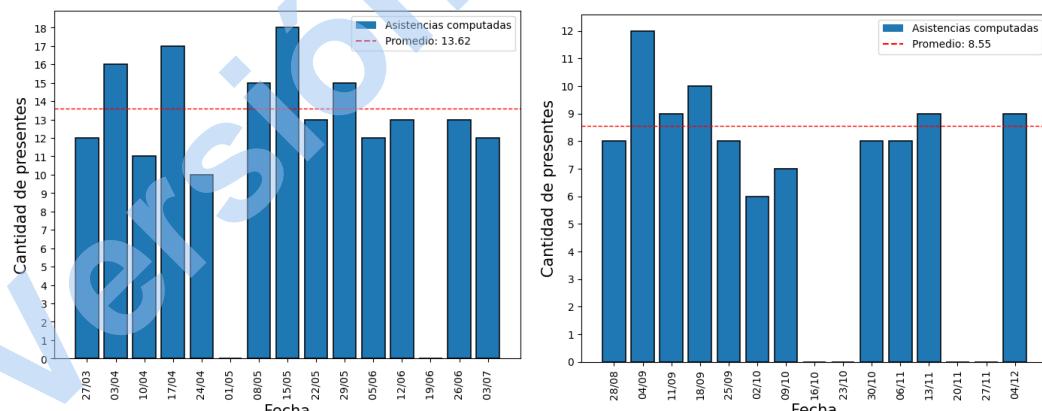
Para el caso de la segunda etapa, se consultó además sobre la sensación de seguridad respecto a los contenidos vistos el cuatrimestre anterior y sobre la utilidad del kit de materiales provisto a cada grupo:

- ¿Sentiste que arrancaste bien preparado este segundo cuatri? (Sí, no, no estoy seguro/no cursé)
- ¿Cómo calificarías la herramienta “Arduino en la Escuela” para aprender programación por bloques? (Excelente, buena, regular, mala)
- ¿Cuál fue tu impresión general sobre las prácticas de robótica que realizamos con Arduino? (Me encantaron, me gustaron, estuvieron bien, no me gustaron)
- ¿Qué tan útiles te resultaron las clases del cuatri pasado sobre robótica, programación y pensamiento computacional? (Muy útiles, útiles, algo útiles, poco útiles, nada útiles)
- ¿Tuviste la oportunidad de aplicar los conocimientos adquiridos en el taller en tu proyecto final de robótica? (Sí, completamente; si, en parte; no; no realicé el proyecto final)
- ¿Qué te pareció el kit de componentes de robótica con el que trabajaste este cuatri? (Me pareció útil y completo; Me pareció útil pero le faltaban cosas; No me pareció útil)
- ¿Recomendarías este taller a otras personas interesadas en aprender sobre robótica y programación? (Sí, definitivamente; si, con algunas mejoras; no estoy seguro; no)

### 3. Resultados

Cuadro 1: Datos de inscriptos, asistentes reales y personas aprobadas en el curso.

Complejo/ Etapa	Inscriptos	Asistentes	Aprobados
Ezeiza - E1	39	29	13 (44,83 %)
Ezeiza - E2	18	14	9 (64,29 %)
Devoto - E1	121	63	37 (58,73 %)
Devoto - E2	56	33	20 (60,61 %)



(a) 1er. cuatrimestre.

(b) 2do. cuatrimestre.

Figura 1: Asistencia por día al curso en el CUE.

#### 3.1. Asistencias y aprobación del curso

A lo largo de las cursadas se registraron las asistencias. Al final se consideraron aprobados a los estudiantes que hubieran cumplido con los objetivos del curso. Como se puede observar en la tabla 1, tanto en el CUD como en el CUE el porcentaje de aprobación creció en la segunda etapa con respecto a la primera. Se debe tener en cuenta que, en muchos casos, cuando un estudiante no recibe bajada no la reclama y deja de asistir (o no logra asistir nunca). Estas circunstancias están fuera del radar de los docentes y no es posible identificar fehacientemente el motivo de un abandono del curso.

En los gráficos de la figura 1 se observa el conteo de asistencias por fecha de clase para el CUE. Considerando que para el primer cuatrimestre los días 01/05 y 19/06 fueron feriados y para el segundo cuatrimestre los días 16/10 y 20/11

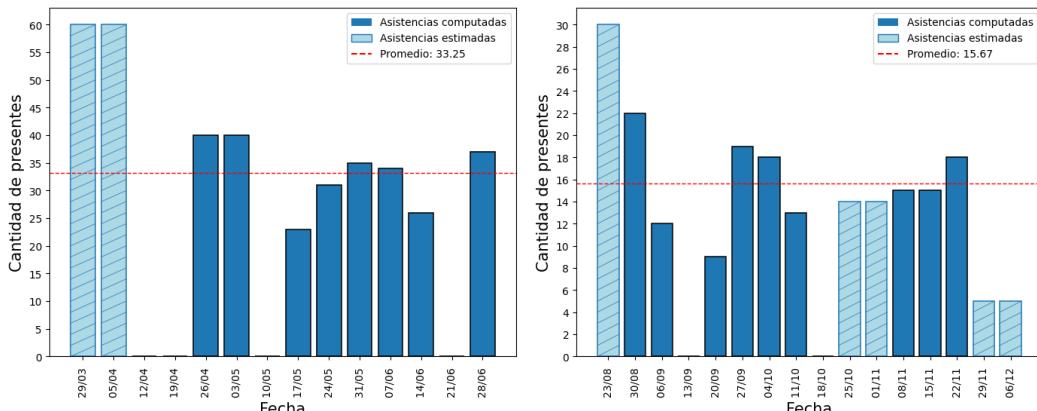


Figura 2: Asistencia por día al curso en el CUD.

fueron feriados, el 23/10 fue una ausencia programada y el 27/11 ningún estudiante obtuvo su bajada, se nota que los estudiantes de Ezeiza mantienen una asistencia regular. La asistencia promedio en ambas etapas coincide con la cantidad de estudiantes aprobados al final del curso. La mayoría de las bajas estuvieron relacionadas con cortes de bajada. Como valoración personal de la docente, los estudiantes de Ezeiza mostraron un perfil altamente colaborativo y se comprometieron con el desarrollo de la clase. En los gráficos de la figura 2 se observa el conteo de asistencias por fecha de clase para el Complejo de Devoto.

En este caso, hubo mayores irregularidades. Respecto al primer cuatrimestre, las dos primeras clases hubo un desborde debido a la alta cantidad de inscriptos por lo cual no se pudo contabilizar la asistencia aunque se estima que fue de alrededor de 60 personas. En consecuencia se reajustaron las listas y se tomó la decisión de no computar asistencias. El 12/04 se suspendió la actividad por asuntos judiciales en el penal y el 19/04 hubo una baja masiva debido a lo mismo. Los días 10/05 y 21/06 hubo suspensión de actividades por enfermedad de la docente. Respecto al segundo cuatrimestre, la primera clase no se computaron asistencias por la reorganización de los turnos pero se estima que asistieron 40 personas. El 13/09 y 18/10 hubo suspensión de actividades por enfermedad de la docente. Los días 25/10 y 01/11 hubo actividades programadas sin supervisión docente y no se computaron asistencias, pero se estima que asistieron 15 personas. Los días 29/11 y 06/12 hubo problemas administrativos debido a los cuales no se tramitó la mayoría de las bajadas y se decidió no computar asistencias, pero se estima una asistencia de 5 personas en ambos casos.

Si bien los datos de asistencia recogidos en el CUD no son lo suficientemente consistentes para un análisis preciso, se pueden destacar algunos puntos. En primer lugar, son notables los conflictos administrativos en este Centro Universitario. Las inscripciones pueden ser masivas y saturar los espacios de estudio. En consecuencia, los estudiantes se desmotivan y dejan de asistir a clase, aún si bajan al complejo de cualquier manera.

Un dato a tener en cuenta es que en ambas etapas, cuando la docente reportó su ausencia por enfermedad, hubo una baja significativa en la asistencia a la clase siguiente. Esto ocurrió porque los estudiantes asumieron que la docente abandonó su cargo y el taller se cancelaría. Esta baja no se repitió luego de la segunda ausencia por enfermedad porque la docente explicó que no abandonaría el curso. Como valoración personal de la docente, los estudiantes de este Centro pueden ser colaborativos en clase y demostrar interés, pero no son constantes en su desempeño.

### 3.2. Encuestas de fin de curso

#### 3.2.1. Etapa 1

La encuesta fue respondida por 12 estudiantes en el CUE y 37 estudiantes en el CUD. Todos los encuestados realizaron el proyecto final. Se puede observar en el Cuadro 2 (b) que la mayoría de los estudiantes considera que los contenidos del taller fueron adecuados para su nivel de conocimiento. Se debe tener en cuenta, sin embargo, que según el Cuadro 2 (a) en ambos centros la mayoría de los estudiantes reporta haber comenzado con al menos un mínimo conocimiento en el tema. Respecto de las herramientas pedagógicas, se observa en el Cuadro 2 (c) que la mayoría de los encuestados calificó al software Arduino en la Escuela de manera positiva y hubo una aceptación generalizada sobre las prácticas de robótica (ver Cuadro 2 (d)). En consonancia con esto, se observa en el Cuadro 2 (e) que las clases teóricas fueron mayormente consideradas de utilidad y según se observa en el Cuadro 2 (f), la mayoría de los encuestados considera haber podido aplicar los conocimientos adquiridos en el proyecto final de robótica.

#### 3.2.2. Etapa 2

La encuesta fue respondida por 7 estudiantes en el CUE y 15 estudiantes en el CUD. Todos participaron en un proyecto integrador. Según lo visto en el Cuadro 3 (d), respecto de los contenidos del taller anterior, la mayoría considera que fueron de utilidad para esta segunda etapa. En el Cuadro 3 (a) se puede observar, además, que consideran haber comenzado

Cuadro 2: Respuestas a las preguntas de la encuesta final de la primera etapa en ambos complejos

(a) ¿Cuál fue tu nivel de experiencia previa en robótica, programación y pensamiento computacional antes de comenzar el taller?

Complejo	Nada	Poca	Algo	Bastante	Ya era un experto
Ezeiza	3	4	3	2	0
Devoto	8	7	12	10	0

(b) ¿Consideras que los contenidos del taller fueron adecuados para tu nivel de conocimiento?

Complejo	Si	No	No estoy seguro
Ezeiza	12	0	0
Devoto	36	0	1

(c) ¿Cómo calificarías la herramienta “Arduino En La Escuela” para aprender programación por bloques?

Complejo	Excelente	Buena	Regular	Mala
Ezeiza	7	5	0	0
Devoto	20	17	0	0

(d) ¿Cuál fue tu impresión general sobre las prácticas de robótica que realizamos con Arduino?

Complejo	Me encantaron	Me gustaron	Estuvieron bien	No me gustaron
Ezeiza	5	5	2	0
Devoto	13	17	7	0

(e) ¿Qué tan útiles te resultaron las clases teóricas sobre robótica, programación y pensamiento computacional?

Complejo	Muy útiles	Útiles	Algo útiles	Poco útiles	Nada útiles
Ezeiza	8	3	0	0	0
Devoto	20	12	5	0	0

(f) ¿Tuviste la oportunidad de aplicar los conocimientos adquiridos en el taller en tu proyecto final de robótica?

Complejo	Completamente	En parte	No pude	No realicé el proyecto
Ezeiza	8	3	0	0
Devoto	16	20	0	0

preparados. Sobre el proyecto, todos los encuestados afirman haber disfrutado las prácticas de robótica, según el Cuadro 3 (c), además de que pudieron, en su mayoría, aplicar los contenidos vistos en la realización del mismo, como se aprecia en el Cuadro 3 (e). Sobre este último apartado, tanto en Ezeiza como en Devoto una persona afirma no haber podido aplicarlos.

Es considerable además observar en el Cuadro 3 (b) que la herramienta AelE continúa teniendo mayoría de opiniones positivas, considerando que para esta etapa los proyectos han sido más complejos y por lo tanto lo han sido también sus respectivos programas.

Respecto del kit de materiales, se observa en el Cuadro 3 (f) que la mayoría considera que fue “útil y completo”. En Devoto, dos encuestados respondieron “Me pareció útil pero le faltaban cosas”. Finalmente, es observable en el Cuadro 3 (g) que todos los encuestados afirman que recomendarían este taller a otras personas interesadas en aprender sobre robótica y programación. Para este punto es importante aclarar que se explicó verbalmente que esta pregunta estaba referida a personas de los complejos penitenciarios con el fin de evaluar la continuidad del taller en futuras ediciones para el mismo contexto.

### 3.3. Trabajo con materiales electrónicos

Por último, unos de los aspectos que nos interesó evaluar de la experiencia de trabajo con los participantes del curso es la forma en qué pudieron desempeñarse con los componentes electrónicos. Si bien las prácticas y los proyectos se realizaron con materiales de bajo costo, el presupuesto requerido para conseguirlos es significativo y el cuidado de los mismos es fundamental.

Durante la primera etapa, los materiales se repartían según los requerimientos de cada clase. Al final del cuatrimestre hubo faltas significativas: de las 10 placas Arduino UNO dos se quemaron por un mal armado de los circuitos y dos cables de alimentación y datos se perdieron. A lo largo de cada clase se quemó al menos un sensor o actuador y se perdieron además innumerables diodos y conectores (cables de ficha dupont).

Durante la segunda etapa, cada grupo debió usar y responsabilizarse por un mismo kit de materiales. Al final de cada clase debieron completar una hoja de reporte para informar el estado de los mismos. No se planteó ningún tipo de penalización si dentro del kit faltaba o se averiaba algo. Sin embargo, al final del curso ningún kit sufrió una falta significativa y no se quemó ningún sensor.

Cuadro 3: Respuestas a las preguntas de la encuesta final de la segunda etapa en ambos complejos

(a) ¿Sentiste que arrancaste bien preparado este segundo cuatri?

Complejo	Si	No	No estoy seguro/ No cursé
Ezeiza	7	0	0
Devoto	14	0	0

(b) ¿Cómo calificarías la herramienta “Arduino En La Escuela” para aprender programación por bloques?

Complejo	Excelente	Buena	Regular	Mala
Ezeiza	6	1	0	0
Devoto	13	2	0	0

(c) ¿Cuál fue tu impresión general sobre las prácticas de robótica que realizamos con Arduino?

Complejo	Me encantaron	Me gustaron	Estuvieron bien	No me gustaron
Ezeiza	7	0	0	0
Devoto	10	4	1	0

(d) Qué tan útiles te resultaron las clases del cuatri pasado sobre robótica, programación y pensamiento computacional?

Complejo	Muy útiles	Útiles	Algo útiles	Poco útiles	Nada útiles
Ezeiza	7	0	0	0	0
Devoto	7	5	2	0	0

(e) ¿Tuviste la oportunidad de aplicar los conocimientos adquiridos en el taller en tu proyecto final de robótica?

Complejo	Completamente	En parte	No pude	No realicé el proyecto
Ezeiza	6	0	1	0
Devoto	9	4	1	0

(f) ¿Qué te pareció el kit de componentes de robótica con el que trabajaste este cuatri?

Complejo	Me pareció útil y completo	Me pareció útil pero le faltaban cosas	No me pareció útil
Ezeiza	7	0	0
Devoto	13	2	0

(g) ¿Recomendarías este taller a otras personas interesadas en aprender sobre robótica y programación?

Complejo	Si, definitivamente	Si, con algunas mejoras	No estoy seguro	No, no lo recomendaría
Ezeiza	7	0	0	0
Devoto	15	0	0	0

## 4. Conclusiones

En la primera etapa de este curso se buscó enseñar conceptos fundamentales de la programación con el uso de una herramienta que resultara divertida y motivadora. Si bien no hay datos iniciales para contrastar con los resultados de la encuesta final, se destaca que los contenidos del taller fueron adecuados y útiles para la realización del proyecto final. De acuerdo a la valoración de la docente, se considera que los estudiantes fueron receptivos en todas las clases y su participación se mantuvo en el tiempo, aún con los altibajos característicos de cada Centro Universitario.

En la segunda etapa se buscó que los estudiantes pudieran trabajar en un proyecto independiente con una guía mínima de la docente. Además, se les asignó un kit de materiales con el que debieron trabajar a lo largo de todo el cuatrimestre. Para este proyecto los estudiantes consideraron de utilidad lo visto en la etapa anterior y respondieron positivamente a la implementación del nuevo kit. La docente considera que en esta nueva etapa los estudiantes se mostraron mucho más comprometidos y se dedicaron afectuosamente a sus proyectos. Además, consideramos que la responsabilidad sobre el kit de materiales contribuyó al cuidado de los mismos y a una actitud protectora de los estudiantes con sus proyectos.

Respecto al objetivo del taller, todos los estudiantes aprobados en la primera etapa pudieron resolver problemas de automatización con el uso de una herramienta de programación con bloques y en la segunda etapa pudieron diseñar, diagramar y programar un proyecto complejo, por lo que la docente lo considera cumplido. Sin embargo, debido a las dificultades del contexto muchos estudiantes no pudieron mantener la constancia o llegar al final del curso. Este problema es ajeno al trabajo docente pero se espera que una mejoría sobre estas dificultades pueda implicar una mejoría en el porcentaje de aprobación respecto de la cantidad de inscriptos con al menos una asistencia.

El marco en el cual se desarrollan estos cursos es interesante para educadores y otras personas relacionadas al ámbito educativo, tanto formal como informal, ya que, más allá de que la educación en cárceles tiene características particulares. Así, algunos de sus desafíos y complejidades también aparecen en otros contextos como, por ejemplo, en el desarrollo de actividades en barrios populares, debido a que los participantes pueden ausentarse por diversos motivos o sin ninguna justificación, no se dispone de un contexto totalmente adecuado para el desarrollo de las tareas del curso, además de la

poca disponibilidad de recursos. Por otro lado, en el contexto de encierro el uso de herramientas electrónicas puede verse limitado por las medidas de seguridad, ya que su ingreso al penal requiere autorizaciones especiales que demoran en su tramitación y se pueden suspender por motivos ajenos al trabajo docente.

Creemos que el diseño de los cursos y las experiencias registradas pueden ser de mucha utilidad para otras personas que deban dar clases en contexto de encierro, aunque creemos que también son relevantes para contextos de vulnerabilidad en general.

## 5. Agradecimientos

A todo el equipo de UBA XXII de Exactas. En particular, a Nair Repollo y a los coordinadores y estudiantes de los Centros Universitarios de Devoto y Ezeiza.

## Referencias

- [1] H. Rangel Torrijo y M. De Maeyer. «Education in prison: A basic right and an essential tool». En: *International Review of Education* 65 (2019), págs. 671-685. URL: <https://doi.org/10.1007/s11159-019-09809-x>.
- [2] T. Bretl. «Teaching Undergraduate Courses on Robotics and Control in Prison». En: *ASME. Mechanical Engineering* 140.9 (2018), S11-S16. URL: <https://doi.org/10.1115/1.2018-SEP8>.
- [3] D. N. de Política Criminal en materia de Justicia y Legislación Penal. *Sistema Nacional de Estadísticas sobre Ejecución de la Pena (SNEEP) - Informe 2022*. <https://www.argentina.gob.ar/justicia/politicacriminal/estadisticas/sneep>. 2022.
- [4] M. A. Lescano. «Estímulo educativo en contexto de encierro. La ley 26.695, un nuevo instrumento legal que garantiza y asegura el acceso a la educación de los internos del sistema penitenciario». En: *Revista Pensamiento Penal* 145 (2012).
- [5] M. Laferriere. *La universidad en la cárcel: Programa UBA XXII*. Buenos Aires, Editorial Universidad, 2006.
- [6] M. Laferriere. *Programa UBA XXII: herramienta para la construcción de ciudadanía*. Universidad de Buenos Aires. Facultad de Ciencias Sociales. Carrera de Trabajo Social, 2019.
- [7] A. D'Ausilio. «Arduino: A low-cost multipurpose lab equipment». En: *Behavior research methods* 2.44 (2012), págs. 305-313.
- [8] J. M. Wing. «Computational Thinking». En: *Communications of the ACM* 49.3 (2006), págs. 33-35. ISSN: 0001-0782. DOI: 10.1145/1118178.1118215. URL: <https://dl.acm.org/doi/10.1145/1118178.1118215>.
- [9] E. Nardelli. «Do we really need computational thinking?». En: *Communications of the ACM* 62.2 (2019), págs. 32-35.
- [10] M. B. Bonello y F. Schapachnik. «Diez preguntas frecuentes (y urgentes) sobre pensamiento computacional». En: *Virtualidad, Educación y Ciencia* 20.11 (2020), págs. 156-167. ISSN: 1853-6530. URL: <https://revistas.unc.edu.ar/index.php/vesc/workflow/index/27453/5>.
- [11] A. Juškevičienė, G. Stupurienė y T. Jevsikova. «Computational thinking development through physical computing activities in STEAM education». En: *Comput Appl Eng Educ* 2021.29 (2020), págs. 175-190. DOI: 10.1002/cae.22365.
- [12] F. Kalelioglu y S. Sentance. «Teaching with physical computing in school: the case of the micro:bit». En: *Education and Information Technologies* 25.4 (2020), págs. 2577-2603. ISSN: 1573-7608. DOI: 10.1007/s10639-019-10080-8.
- [13] A. Sanzo, F. Schapachnik, P. Factorovich y F. Sawady O'Connor. «Pilas Bloques: A scenario-based children learning platform». En: *2017 Twelfth Latin American Conference on Learning Technologies* (2017), págs. 1-6.
- [14] G. P. Fernández y C. Cossio-Mercado. «AeIE: a versatile tool for teaching programming and robotics using Arduino». En: *50a Conferencia Latinoamericana de Informática (CLEI)* (2024).
- [15] G. P. Fernández, M. B. Ticona Oquendo y C. Cossio-Mercado. «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* (2021), págs. 32-44.
- [16] C. Cossio-Mercado y G. P. Fernández. «PRENDER: Una propuesta didáctico-pedagógica para la enseñanza de las Ciencias de la Computación». En: *Jornadas Argentinas de Didáctica de las Ciencias de la Computación 2023* (2023). URL: [https://www.researchgate.net/publication/380173508\\_PRENDER\\_Una\\_propuesta\\_didactico-pedagogica\\_para\\_la\\_ensenanza\\_de\\_las\\_Ciencias\\_de\\_la\\_Computacion](https://www.researchgate.net/publication/380173508_PRENDER_Una_propuesta_didactico-pedagogica_para_la_ensenanza_de_las_Ciencias_de_la_Computacion).

# **Inteligencia Artificial Generativa como herramienta para el desarrollo de habilidades de investigación en la asignatura toxicología de los alimentos**

Maria Laura Vranic<sup>1,2</sup>, Carina Inés Kramer<sup>1</sup>, Viviana Beatriz Panattu<sup>1</sup>, José Ignacio Petraglia<sup>1</sup>, Verónica Luna<sup>1</sup> y Liliana Lujan Lezcano<sup>1</sup>.

<sup>1</sup>Universidad Nacional de Luján, Luján, Argentina

<sup>2</sup>Universidad Tecnológica Nacional, Medrano, Argentina

[mlvranic@gmail.com](mailto:mlvranic@gmail.com), [lilianalujanlezcano@yahoo.com.ar](mailto:lilianalujanlezcano@yahoo.com.ar)

## **Resumen**

Es fundamental desarrollar sólidas habilidades de investigación para que un estudiante de toxicología de alimentos de la carrera de Ingeniería de Alimentos se convierta en un profesional competente y capaz de contribuir a la seguridad alimentaria. Estas habilidades le permitirán mantenerse actualizado en un campo en constante evolución, tomar decisiones y comunicarse de manera efectiva con otros profesionales. La inteligencia artificial (IA) está transformando rápidamente diversos campos, incluida la educación. Su capacidad para simular el pensamiento humano, aprender y adaptarse, ofrece un enorme potencial para mejorar los procesos de enseñanza y aprendizaje (1). En el contexto de la asignatura se implementó un proyecto innovador utilizando IA Generativa (IAGen) como herramienta de investigación. El proceso involucró una exploración inicial de las percepciones de los estudiantiles sobre la IA mediante una encuesta, seguida de la asignación de una tarea práctica individual donde los estudiantes debían generar la documentación necesaria para exportar un aditivo alimentario a la Unión Europea, mediante la utilización de herramientas de la inteligencia artificial. Se evaluaron los trabajos, incluida una evaluación por pares. Para recoger la opinión de los estudiantes sobre esta experiencia se realizó otra encuesta. Los estudiantes lograron producir trabajos de calidad y valoraron positivamente la experiencia.

**Palabras clave:** Inteligencia Artificial; Toxicología; Ingeniería en Alimentos

## **1. Introducción**

La Inteligencia Artificial (IA), al igual que la humana, es un concepto complejo de definir, no existe una definición formal y universalmente aceptada. La IA Generativa (IAGen) es una tecnología de inteligencia artificial (IA) que genera contenidos de forma automática en respuesta a instrucciones escritas en interfaces conversacionales de lenguaje natural (prompts). En vez de limitarse a conservar las páginas web existentes, IAGen produce nuevos contenidos. El contenido puede presentarse en formatos que abarcan todas las representaciones simbólicas del pensamiento humano: textos escritos en lenguaje natural, imágenes (incluyendo fotografías, pinturas digitales y caricaturas), videos, música y código de software. La IAGen se entrena utilizando datos obtenidos de páginas web, conversaciones en redes sociales y otros medios en línea. Genera su contenido analizando estadísticamente la distribución de palabras, píxeles u otros elementos en los datos que ha asimilado e identificando y repitiendo patrones comunes (por ejemplo, que palabras siguen habitualmente a otras).

Aunque la IAGen puede producir nuevos contenidos, no puede crear nuevas ideas o soluciones a los retos del mundo real, ya que no entiende objetos reales ni las relaciones sociales que sustentan el lenguaje. Además, a pesar de su fluidez y su impresionante capacidad de producción, no se puede confiar en la precisión de la IAGen. De hecho, hasta el proveedor de ChatGPT reconoce que “aunque herramientas como ChatGPT pueden generar frecuentemente respuestas que suenan

razonables, no se puede confiar en que sean certeras". La mayoría de las veces, los errores pasarán desapercibidos a menos que el usuario posea sólidos conocimientos sobre el tema en cuestión. La IAGen irrumpió en la escena pública a finales de 2022 con el lanzamiento de ChatGPT, que se convirtió en la aplicación de más rápido crecimiento en la historia. Con el poder de emular las capacidades humanas para generar resultados en formatos como texto, imágenes, videos, música y códigos de software, estas aplicaciones de IAGen han causado sensación. Millones de personas utilizan ahora la IAGen en su vida diaria, y el potencial de adaptar los modelos a aplicaciones de IA de dominios específicos parece ilimitado. Este amplio rango de capacidades de procesamiento de información y de producción de conocimiento tiene, potencialmente, grandes implicancias para la educación, dado que replica el pensamiento de orden superior que constituye la base del aprendizaje humano (2). Nos encontramos en una coyuntura histórica marcada por la incipiente pero decisiva presencia de la Inteligencia Artificial (IA) en todos los ámbitos de nuestra sociedad. Este fenómeno no es ajeno al sector educativo, en especial a las universidades, instituciones que han sido y deben continuar siendo faros de conocimiento, innovación y transformación social (3).

El objetivo de este trabajo es compartir una experiencia realizada en el marco de la asignatura Toxicología de los Alimentos donde los estudiantes debían generar la documentación necesaria para exportar un aditivo alimentario a la Unión Europea, mediante la utilización de herramientas de inteligencia artificial.

## 2. Descripción de la Experiencia

### 2.1. Participantes

Los 10 participantes de la experiencia están cursando la asignatura Toxicología de alimentos de la carrera de Ingeniería en alimentos de la Universidad Nacional de Luján. El plan de estudios se compone de 50 asignaturas y tiene como objetivo formar profesionales altamente calificados capaces de gestionar, investigar, asesorar y peritar en el ámbito de la industria alimentaria, con un fuerte énfasis en la calidad, la seguridad y la eficiencia. La asignatura se encuentra en el primer cuatrimestre del último año de la carrera. Los estudiantes tienen un mínimo de 27 materias aprobadas y un máximo de 42, 9 estudiantes trabajan.

#### 2.1. La experiencia

El trabajo fue individual con el uso de la IA como compañero de estudio. Para el desarrollo de la experiencia se realizaron los siguientes pasos:

**2.1.1. Encuesta Inicial:** con el objetivo de comprender el nivel de conocimiento, uso y percepción que tienen los estudiantes sobre la inteligencia artificial (IA) se realizó una encuesta anónima. La encuesta se divide en cuatro secciones: Datos demográficos, Conocimiento y uso de la IA, Actitudes y creencias sobre la IA y se incluyó una sección para comentarios adicionales. Se incluyeron preguntas cerradas de opción múltiple, dicotómicas y de escala. Para permitir al encuestado expresar sus opiniones o experiencias en sus propias palabras se incluyeron preguntas abiertas de respuesta libre.

**2.1.2. Asignación tarea:** Con la finalidad de activar la atención de los estudiantes y establecer el propósito, se presentó una infografía que hablaba de la "tareonomía de la IA generativa en la educación". Inmediatamente se les preguntó a los estudiantes ¿Qué entienden por tareonomía?, como nadie conocía la palabra se le preguntó a las IAs más utilizadas: Chat GPT, Gemini y Copilot. Luego, se compararon las respuestas y se observó que no todas las respuestas eran válidas. A continuación, se describió la tarea a realizar por los estudiantes: La industria alimentaria es un sector dinámico y competitivo que exige a los productores estar a la vanguardia en cuanto a innovación y cumplimiento de regulaciones. La generación de documentación para la venta de productos alimenticios, tanto en el mercado interno como internacional, puede ser un proceso complejo y laborioso, que requiere de un conocimiento profundo de las regulaciones vigentes y de habilidades técnicas específicas.

**Descripción del problema:** Trabajo en una empresa del rubro alimenticio que produce un aditivo alimentario, necesito ampliar el volumen de ventas llegando a mercados como el europeo, ¿cómo tengo que proceder para poder vender en la comunidad europea? ¿Qué documentación necesito?

**Objetivo de la tarea:** Generar la documentación necesaria para exportar un aditivo alimentario a la Unión Europea, mediante la utilización de herramientas de la inteligencia artificial.

**Guía de trabajo:**

- Pregúntale a Chat GPT, Gemini y Copilot: ¿Qué países de la Unión Europea utilizan el aditivo alimentario en la elaboración de alimentos? Chatea hasta obtener toda la información necesaria para establecer una estrategia de mercado.
- En base a la respuesta obtenida seleccione uno de los tres asistentes de inteligencia artificial (IA), fundamenta tu respuesta. Analiza las ventajas y desventajas de cada uno y fundamenta tu elección.
- Luego, se habló de la importancia de mejorar las preguntas para obtener una respuesta más precisa, se definió qué es un prompt, cómo funcionan, cuáles son las buenas prácticas que se deben seguir para generar un prompt efectivo para mejorar las preguntas (4). Repensando el prompt: teniendo en cuenta las buenas prácticas y recomendaciones para generar prompts; reformule los prompts utilizados.
  - Acceder al prompt Perfect y generar un nuevo prompt.
  - Utilice este nuevo prompt en la IA seleccionada: Chat GPT, Gemini o Copilot.
- Generar un documento: Investigue dentro de la IA y obtenga los datos necesarios para generar la documentación.
- Verifique la veracidad de la información.
- Con la ayuda de alguna de las siguientes IA genere un video: Pictory, Synthesia, Invideo AI, Muse.ai, Rephrase.ai, Rawshorts, Lumen 5.

Los docentes seleccionaron los 10 aditivos alimentarios de uso permitido en la Comunidad Europea de la base de datos del EFSA, aleatoriamente fueron asignados a los estudiantes, vía mail. Se establecieron las fechas de pre-entrega y entrega final. Como material deberían entregar un documento con toda la información necesaria para exportar el aditivo alimentario, el video creado por IA, las citas bibliográficas utilizadas y el registro del camino seguido (entregar los chats utilizados).

**2.1.3. Evaluación:** Los docentes realizaron una doble evaluación: se evaluó la pre-entrega y el trabajo final. Una vez finalizado el trabajo se compartieron los resultados y las experiencias. Cada estudiante presentó el trabajo realizado y contó la trayectoria que siguió hasta la elaboración del documento final. Con la ayuda de un cuestionario elaborado por los docentes donde se tuvieron en cuenta aspectos específicos de los conocimientos a adquirir se realizó una evaluación por pares.

**2.1.4. Encuesta Final:** Se les preguntó cuál de las IA generativas escogió y por qué, se evaluó la eficiencia en el uso de los mejoradores de prompts, nivel de satisfacción con respecto a la veracidad de las respuestas, eficiencia, tiempo requerido, validación de la información, ventajas y desventajas, se incluyó una pregunta abierta que es la siguiente: ¿De qué manera le gustaría que se implementara la IA en el entorno universitario?

### 3. Resultados

#### 3.1 Encuesta Inicial

Todos los estudiantes han oído hablar de la IA, como definiciones de la IA dieron las siguientes: algoritmos que acceden a bases de datos, sistemas que recopilan información y se asemejan al pensamiento lógico humano, herramientas para realizar tareas que requieren inteligencia humana, programas que responden a pedidos complejos, aplicaciones que buscan en bases de datos y responden al instante y grandes bases de datos que realizan acciones a través del análisis de datos electrónicos.

Han utilizado herramientas de IA como ChatGPT, buscadores, televisores inteligentes, plataformas de streaming, mapas, GPS, traductores, Zoom, Uber, entre otros. Con respecto a la frecuencia y propósito del Uso: 4 estudiantes la utilizan a veces, otros 3 raramente; 3 frecuentemente y uno solo constantemente. En general las utilizan para la búsqueda de información o resolver tareas, en menor medida para comunicación o entretenimiento. Otros usos incluyen movilidad, comercio, localización, confirmación de ideas, y búsqueda rápida de datos.

De la IA lo que les gusta es: la eficiencia y productividad, su facilidad de uso y accesibilidad, aprendizaje y descubrimiento de nuevas funcionalidades, centralización de información y recursos, la rapidez y versatilidad en la resolución de problemas. Les preocupa el uso indiscrecional de información privada, el hostigamiento publicitario, que pueda brindar respuestas incorrectas o inventadas y les preocupa la creación de dependencia excesiva.

Con respecto al impacto de la IA en el Aprendizaje y Vida Cotidiana, 8 estudiantes creen que la IA puede mejorar la experiencia de aprendizaje; en la vida cotidiana la utilizan para hacer compras online, reservas de viajes, comunicación, entretenimiento, gestión de finanzas, control de salud. En el trabajo y estudios la utilizan para la búsqueda de información, traducción de idiomas, redacción de textos, análisis de datos.

Un total de 7 estudiantes no creen que la IA sea una amenaza para los empleos, 6 creen que la IA puede mejorar el mundo, 9 creen que la IA puede aumentar la productividad y todos creen que los humanos deben tener control sobre la IA.

Les preocupa que la IA sea indebidamente utilizada y destacan la necesidad de “educar” sobre el uso correcto de la IA.

### 3.2 Tarea

Exportar productos alimenticios a Europa implica cumplir con una serie de requisitos normativos y documentales para garantizar la seguridad alimentaria y el cumplimiento de los estándares europeos. La documentación generalmente requerida incluye el Registro Nacional de Establecimiento (Certifica que tu establecimiento cumple con las normas sanitarias para la elaboración de alimentos), el Registro Nacional de Producto Alimenticio (Identifica cada producto y garantiza que cumple con la normativa vigente), el Certificado Fitosanitario o Sanitario, Etiquetado (las etiquetas deben estar en el idioma del país de destino y cumplir con las regulaciones europeas en cuanto a información nutricional, ingredientes y alérgenos), Factura comercial y documentación aduanera como la declaración de exportación, lista de empaque, entre otros. Los productos deben cumplir con los reglamentos alimentarios de la Unión Europea, que son muy estrictos en cuanto a seguridad alimentaria, higiene y calidad. Se suelen requerir análisis de laboratorio para verificar la composición del producto y la ausencia de contaminantes. Implementar un sistema de gestión de calidad, como HACCP, es altamente recomendable para asegurar la trazabilidad y la seguridad alimentaria.

En general, los trabajos presentados por los estudiantes fueron muy completos y ordenados. Todos compartieron el recorrido realizado en la búsqueda de información. Los estudiantes, al trabajar en el proyecto y recibir retroalimentación de sus pares, reconocieron la importancia de verificar la información y las limitaciones de la IA. Sin embargo, a pesar de encontrar los datos técnicos para el registro europeo, la mayoría descuidó los aspectos legales y documentales para exportar el producto desde Argentina. Con respecto a la elaboración de los videos, todos utilizaron las versiones gratuitas y se repitieron imágenes y trayectorias. Sorprendió gratamente la capacidad de algunos estudiantes para ordenar y presentar la información obtenida.

### 3.3 Encuesta Final:

**Selección IAGen:** Para la búsqueda de la información 7 estudiantes eligieron Chat GPT y 3 eligieron Gemini. Las razones de la elección fueron las siguientes: porque su respuesta fue más clara y completa, porque brindó la información más completa sin restricción de número de interacciones, porque fue la que me dio la información más coherente en cuestión a lo que consultaba, la respuesta más precisa, la elegí porque fue la que me pareció más segura, respuesta más completa, fue la que mejor contestó a las preguntas, sin límite de cantidad ni tener que repetir de forma más específica la pregunta para llegar a la respuesta buscada, la herramienta fue más puntual y contestó lo que había preguntado. No agregó información adicional no consultada, proporcionó las respuestas de manera más clara y porque me brindó las respuestas más completas.

**Eficiencia en el uso de los prompt:** para 4 estudiantes el uso de prompts fue más eficientes, condujo a respuestas más precisas y relevantes por parte de la IA, para 2 estudiantes la IA pudo comprender mejor su intención y proporcionar información más útil, otros 2 estuvieron más satisfecho con el rendimiento general de la IA cuando se utilizaron prompt, un estudiante recomendaría usar prompts más eficientes al interactuar con la IA, solamente a un estudiante no le resultó útil el uso de los prompts.

En general los estudiantes se manifestaron satisfechos (6 estudiantes) o muy satisfechos (1 estudiante) con la veracidad de las respuestas dadas por la IA. Solo 5 estudiantes manifestaron que el uso de la IA redujo el tiempo requerido para completar la tarea comparado con los métodos tradicionales. Todos los estudiantes consideran que es necesario validar la información.

**Información incorrecta o desactualizada:** 6 estudiantes manifestaron que han encontrado alguna información incorrecta o desactualizada, 3 no estuvieron seguros si encontraron alguna información incorrecta o desactualizada y solamente 1 no encontró información incorrecta o desactualizada.

Con respecto a la pregunta: ¿En qué medida está de acuerdo con la siguiente afirmación?: “La IA podría reemplazar eventualmente las sugerencias y devoluciones que recibe de sus pares y docentes”, 4 estudiantes se manifestaron en desacuerdo (Creo que la IA nunca podrá reemplazar completamente las valiosas ideas y perspectivas que brindan los compañeros e instructores), 4 neutral (Veo el potencial de la IA para complementar los comentarios de compañeros e instructores, pero creo que ambas formas de comentarios son importantes) y 2 totalmente en desacuerdo (Si bien la IA puede ofrecer algunos comentarios útiles, todavía prefiero los comentarios personalizados y matizados que recibo de compañeros e instructores).

Entre las ventajas, las respuestas obtenidas destacan: optimización del tiempo (la IA agiliza la búsqueda y el procesamiento de información, reduciendo el tiempo necesario para completar tareas), el acceso rápido a información (Permite obtener respuestas rápidas y de diversos temas sin necesidad de consultar múltiples fuentes), eficiencia en la búsqueda de información y agilización del Trabajo (La IA busca en su memoria y proporciona información rápidamente, evitando la necesidad de revisar muchas fuentes diferentes) y resumen de información (la IA tiene capacidad para sintetizar y resumir grandes volúmenes de datos de manera eficiente) y facilidad de uso (la IA y sus herramientas suelen ser accesibles y fáciles de usar, facilitando la tarea para los usuarios).

Con respecto a las desventajas, se destacan: la posibilidad de obtener información errónea (La IA puede proporcionar respuestas equivocadas o desactualizadas y en general, la IA no cita las fuentes de la información proporcionada, lo que dificulta la verificación de la información), la dependencia de la precisión del prompt (La efectividad de la IA depende de la claridad y precisión del prompt; un prompt poco claro puede llevar a respuestas incorrectas o incompletas), necesidad de verificación (Es crucial verificar la información proporcionada por la IA, ya que puede haber errores o respuestas inexactas), limitaciones en la comprensión (La IA puede tener dificultades para interpretar correctamente algunas

preguntas, lo que puede resultar en respuestas no satisfactorias) y costos potenciales (Algunas aplicaciones de IA requieren pagos para acceder a características o resultados mejorados).

A los estudiantes les gustaría que se implementara la IA en el entorno universitario de la siguiente manera: Que se armen cursos o las materias lo involucren más, se podría implementar a la hora de realizar trabajos de investigación, siempre que se verifique que la información que da la IA sea correcta, creen que es importante aprender a utilizar este tipo de herramientas e incluirlas en diferentes asignaturas ya que puede ayudar mucho en la búsqueda y selección de información, teniendo en cuenta la gran cantidad y variedad existente hoy en la red. Les gustaría que les permitan usar la IA como disparador de búsqueda y después verificar la información, para realizar tareas, consideran que para poder implementarla es fundamental una etapa de aprendizaje, como debería ocurrir con la programación. Después a lo largo de la carrera les gustaría ir utilizándola como herramienta para confeccionar informes, tablas, gráficos, traducciones, etc. Pero si no se enseña a ser crítico con la herramienta va a ser más conflictivo que constructivo. Además, si el profesor no tiene voluntad tampoco va a ser crítico con lo que le confeccionó la IA al alumno y para los trabajos prácticos.

## 4. Conclusiones

Si bien la muestra de estudiantes ( $n=10$ ) es limitada, esta investigación exploratoria nos permite vislumbrar el potencial de la IA en la formación de ingenieros en alimentos. A pesar de que algunos estudiantes ya habían interactuado con la IA en otros contextos, la aplicación de estas herramientas en un proyecto de ingeniería específico resultó novedosa y enriquecedora. Sus experiencias y opiniones ofrecen valiosas perspectivas sobre cómo la IA puede apoyar la resolución de problemas reales en el ámbito profesional.

La IA puede ser una herramienta poderosa en la resolución de tareas, los estudiantes resaltaron la necesidad de manejarla con cuidado y verificar la información proporcionada. Destacaron como ventajas: el ahorro de tiempo, que proporciona acceso rápido a información variada, y facilita la obtención de datos resumidos. Y como desventajas: que la información puede ser errónea, es esencial verificar los datos, y la IA puede tener problemas para entender correctamente algunas consultas. Además, puede haber costos asociados para funciones avanzadas y la elaboración precisa de prompts es fundamental.

Acordamos con [1] en que con grandes oportunidades vienen también grandes retos. La ética de la IA, la brecha digital, la adecuación de los currículos y la capacitación del personal son cuestiones que deben abordarse con seriedad y compromiso. Las universidades han de liderar no solo en la adopción de estas tecnologías, sino también en la reflexión crítica sobre su impacto y en la formación de profesionales capaces de trabajar ética y efectivamente en esta nueva era.

A través del desarrollo del trabajo y la evaluación por pares, los estudiantes comprendieron la relevancia de la validación de información y las limitaciones de la inteligencia artificial. Si bien todos lograron encontrar los datos necesarios para registrar un aditivo alimentario en la UE, la mayoría pasó por alto los cruciales requisitos normativos y documentales para exportar el producto desde Argentina, evidenciando la necesidad de un pensamiento crítico más profundo.

Ante la creciente relevancia de la inteligencia artificial en diversos campos profesionales, los estudiantes demandan activamente capacitaciones específicas, la integración de herramientas de IA en las asignaturas y un acompañamiento personalizado por parte de los docentes para desarrollar las competencias necesarias en este ámbito.

Coincidimos con [5] en que es importante destacar que la integración de la inteligencia artificial en los procesos de aprendizaje-enseñanza debe ser realizada de manera ética y responsable, teniendo en cuenta aspectos como la privacidad y la seguridad de los estudiantes, siempre complementando la interacción y el apoyo docente en el proceso educativo. La inteligencia artificial puede ser una herramienta valiosa para apoyar estos procesos, pero, es esencial que la utilicen de manera adecuada y reflexiva, teniendo en cuenta los objetivos pedagógicos y las necesidades de sus estudiantes. Además, es preciso considerar aspectos relacionados con la equidad en el acceso a las tecnologías, la transparencia en el uso de datos y la formación adecuada del profesorado para utilizar estas herramientas basadas en inteligencia artificial, de manera efectiva y responsable. El costo fue el factor limitante en la selección de la Herramienta con IA para la realización del video.

Aprender a utilizar la IA es una inversión a futuro de los estudiantes universitarios, ya que les proporciona las herramientas necesarias para destacar en un mundo cada vez más digital y tecnológico. Enseñar a utilizar la IA desde las universidades es fundamental porque: prepara a los estudiantes para el futuro del trabajo, fomenta el desarrollo de habilidades clave para el siglo XXI, permite a los estudiantes aprovechar al máximo las oportunidades que ofrece la tecnología y contribuye a formar profesionales más competentes y creativos.

## 4. Agradecimientos

A la UNLu y al Departamento de Tecnología por brindarnos el espacio educativo para la realización de esta experiencia. Se agradece muy especialmente a los estudiantes por su compromiso y dedicación: Renata Cabrera, Marcelo Enrique Castilla, Camila Castro, Néstor Agustín Fernández, Anahí Ferrabelo, Javier Nicolás Lamicq, María Eugenia Latapie, Oriana Mangini, Lucas Mateo Massare y Lucila Panessi.

## 5. Referencias

1. *Impact of Artificial Intelligence on Education.* Mureşan, Mircea. RAIS RESEARCH ASSOCIATION FOR INTERDISCIPLINARY STUDIES.
2. UNESCO. Guia para el uso de IA generativa en educación e investigación. s.l. : Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura , 2024.
3. Pedreño Muñoz, Andrés . Inteligencia artificial como catalizador de la renovación universitaria. *La inteligencia artificial en las universidades: retos y oportunidades.* 2024, pág. 6.
4. Harvard University Information Technology. Getting started with prompts for text-based Generative AI tools. [En línea] 30 de Agosto de 2023. <https://huit.harvard.edu/news/ai-prompts>.
5. *Integración de la Inteligencia Artificial en la Higher Education: Challenges and Opportunities.* Vera, Fernando. 1, 2023, Revista electrónica Transformar, Vol. 4. ISSN 2735-6302.

Versión Preliminar

# Codificación a la carta

Carolina Longo

<sup>1</sup>Escuela "Dr. Ricardo Luti", Villa Ciudad Parque, Argentina  
[longocarolina@hotmail.com](mailto:longocarolina@hotmail.com)

## Resumen

El paradigma educativo actual se encuentra centrado en el estudiante, promoviendo el desarrollo de habilidades tales como pensamiento crítico, creatividad y colaboración, en el marco de aulas heterogéneas, diversas e inclusivas. La incorporación de la educación digital en este aspecto resulta fundamental, dado que facilita el acceso a la información, permite la construcción de conocimientos a partir del aprendizaje activo, desarrollando habilidades digitales esenciales en la sociedad actual, empoderando a los alumnos.

En este sentido, la experiencia presente, que comienza con una programación desenchufada, incorporando elementos lúdicos, se encuentra alineada con lo propuesto dentro del diseño curricular de la provincia de Córdoba, atendiendo tanto a prácticas pedagógicas de carácter innovador, como a las necesidades e intereses de los estudiantes involucrados.

**Palabras clave:** Programación desenchufada; Codificación; Lúdico; Inclusión; Didáctica.

## 1. Introducción

En un contexto cada vez más digital, la programación se ha convertido en una habilidad fundamental. Sin embargo, introducir a los niños en este apasionante mundo no siempre requiere de una computadora. La programación desenchufada, una metodología que utiliza actividades prácticas y lúdicas sin la necesidad de dispositivos electrónicos, es una excelente manera de despertar el interés por la codificación desde edades tempranas.

La presente experiencia, propone una situación de aprendizaje divertida y significativa, donde los estudiantes de nivel primario descubrirán el fascinante mundo de la programación a través de la asignación de números a letras. Al armar palabras con base en estos códigos numéricos, los niños no solo estarán desarrollando habilidades de pensamiento lógico y matemático, sino que también estarán dando sus primeros pasos en la comprensión de los algoritmos y la resolución de problemas, pilares fundamentales de la programación.

La gamificación, es decir, la incorporación de elementos propios de los juegos en el proceso de aprendizaje, será una excelente aliada para hacer de esta experiencia algo realmente significativo. A través de desafíos, recompensas y la creación de un ambiente de colaboración, se motiva a los estudiantes a explorar y experimentar con diferentes códigos. Como señalan Kafai y Resnick (1996), el "construcionismo en práctica" permite a los estudiantes diseñar, pensar y aprender en un mundo digital, y la gamificación es una herramienta clave para fomentar este tipo de aprendizaje activo.

La importancia de lo lúdico radica en que el juego es una actividad natural para los niños, que les permite aprender de manera espontánea. Al transformar entonces la codificación en un juego, se facilita la comprensión de conceptos abstractos, fomentando el desarrollo de habilidades cognitivas como la atención, la memoria y la creatividad. En palabras de Papert (1980), los niños pueden convertirse en "pensadores poderosos" a través de experiencias de aprendizaje significativas y lúdicas.

Además de ser una introducción a las Ciencias de la Computación, esta propuesta tiene una estrecha relación con la didáctica y las matemáticas. Al asignar números a letras, se establecen relaciones entre diferentes sistemas de representación, lo que fortalece el pensamiento lógico-matemático. Asimismo, el diseño de actividades lúdicas y la creación de códigos propios promueven el desarrollo de habilidades comunicativas y el trabajo en equipo, aspectos fundamentales de la didáctica contemporánea.

En definitiva, este proyecto busca que los estudiantes de nivel primario comprendan que la programación no es solo una habilidad técnica, sino una forma de pensar y resolver problemas que puede aplicarse a diversas áreas de la vida. A través de actividades divertidas y desafiantes, los niños descubren el placer de crear y experimentar, sentando las bases para un futuro lleno de posibilidades. Como indica Shute (2008), la evaluación formativa en juegos educativos puede ser una herramienta poderosa para fomentar el aprendizaje y la motivación de los estudiantes.

## 2. Descripción de la Experiencia

### 2.1. Propuesta Didáctica

Diseño Detallado de la Unidad Didáctica

Contenidos Seleccionados

Basado en el diseño curricular de la provincia de Córdoba y los lineamientos de Program.ar, se seleccionan los siguientes contenidos:

- Conceptos básicos de programación: Algoritmos simples, secuencias, instrucciones, datos.
- Codificación y decodificación: Asignación de valores numéricos a letras, creación de códigos simples y descifrado de mensajes.
- Lógica matemática: Relaciones entre números y letras, patrones y secuencias.
- Pensamiento computacional: Descomposición de problemas, identificación de patrones, diseño de algoritmos.
- Creatividad y resolución de problemas: Diseño de juegos y actividades basadas en los códigos creados.

Secuenciación de los Contenidos

1. Introducción a la Codificación:

- ¿Qué es un código? Actividades para introducir el concepto de código como un sistema de símbolos que representan información.
- Ejemplos de códigos en la vida cotidiana: códigos de barras, señales de tránsito, etc.

2. Codificación de letras con números:

- Creación de un sistema de codificación simple: asignar un número a cada letra del alfabeto.
- Codificar palabras sencillas y frases cortas.
- Decodificar mensajes utilizando el sistema creado.

3. Juegos y actividades:

- "El mensaje secreto": Los estudiantes crean mensajes codificados y los intercambian con sus compañeros.
- "El detective del código": A partir de un mensaje codificado, los estudiantes deben descubrir el sistema de codificación utilizado.
- "La carrera de codificación": Los estudiantes compiten para ser los primeros en codificar y decodificar una serie de palabras.

4. Ampliación del sistema de codificación:

- Introducción de símbolos especiales y números de dos dígitos.
- Creación de códigos más complejos con sustituciones y permutaciones.

5. Introducción a los algoritmos:

- Explicación del concepto de algoritmo como una secuencia de pasos para resolver un problema.
- Creación de algoritmos sencillos para realizar tareas como dibujar una figura o resolver un laberinto.

6. Actividad final:

- Diseño y creación de un juego de mesa o una aplicación móvil simple basado en los conceptos aprendidos.

Evaluación

La evaluación se realiza de manera formativa y sumativa, considerando los siguientes aspectos:

- Participación: Actitud en clase, interés por las actividades, trabajo en equipo.
- Comprensión de conceptos: Capacidad para explicar los conceptos de codificación, decodificación y algoritmo.
- Habilidades de resolución de problemas: Capacidad para aplicar los conocimientos adquiridos para resolver desafíos.
- Creatividad: Originalidad en la creación de códigos y juegos.

Instrumentos de evaluación:

- Observación sistemática: Durante las actividades, se observa la participación de los estudiantes, su comprensión de los conceptos y su capacidad para resolver problemas.

- Rúbrica de evaluación: Se utiliza una rúbrica para evaluar los proyectos finales, considerando aspectos como la originalidad, la complejidad del código y la claridad de las instrucciones.
- Autoevaluación y coevaluación: Los estudiantes realizan una reflexión sobre su propio aprendizaje y evalúan el trabajo de sus compañeros.

Adaptación a la diversidad:

- Diferentes niveles de dificultad: Se ofrecen actividades adaptadas a las necesidades de cada estudiante.
- Materiales diversos: Se utilizan diversos materiales manipulativos para facilitar la comprensión de los conceptos.
- Atención a las necesidades individuales: Se brinda apoyo personalizado a los estudiantes que lo requieran.

## 2.2. Contexto

La experiencia se desarrolla en quinto y sexto grado del turno mañana, de la escuela "Dr. Ricardo Luti", de la localidad de Villa Ciudad Parque, única escuela de nivel, siendo la misma pública. En ella confluyen diferentes estratos socio, económico, culturales, lo que la convierte en un crisol, en el que el paradigma social de la inclusión no resulta un dato de color menor, sino una categoría en toda regla. Ambos grados son bastante numerosos, contando con 32 y 28 estudiantes respectivamente, con una población relativamente equilibrada de varones y mujeres, de entre 9 y 11 años.

## 2.3. La Experiencia

El aula se transforma en un dinámico laboratorio de codificación, donde el aprendizaje se vuelve una experiencia tangible y divertida. Este espacio, organizado en estaciones de trabajo flexibles, invita a los estudiantes a sumergirse en el mundo de la programación de una manera práctica y colaborativa. La presencia de materiales manipulativos, como fichas y bloques, crea un entorno enriquecedor donde los conceptos abstractos de la codificación se vuelven concretos.

El desarrollo de las actividades en este aula taller se caracteriza por su dinamismo y flexibilidad. Las clases inician con juegos y actividades lúdicas que despiertan la curiosidad y preparan el terreno para la introducción de conceptos más complejos. Los estudiantes trabajan en equipos, fomentando la colaboración y el aprendizaje entre pares. A lo largo de las sesiones, se promueve la experimentación y el descubrimiento, permitiendo que los estudiantes desarrollen sus propias soluciones y resuelvan problemas de manera creativa.

La retroalimentación constante del docente es fundamental en este proceso. A través de observaciones y sugerencias, se guía a los estudiantes hacia un aprendizaje significativo y personalizado. Por ejemplo, en una sesión, los estudiantes trabajan en parejas para diseñar y codificar un laberinto, utilizando códigos de dirección para indicar el camino a seguir. Al finalizar, comparten sus creaciones con el resto de la clase, explicando el proceso de diseño y resolución.

El aula taller ofrece múltiples beneficios para el aprendizaje de la programación. Al combinar lo teórico con lo práctico, lo digital con lo analógico, y el trabajo individual con el colaborativo, se crea un ambiente que favorece el desarrollo de habilidades claves, como el pensamiento computacional, la resolución de problemas, la creatividad y la colaboración. Además, la adaptación de las actividades a diferentes estilos de aprendizaje garantiza que todos los estudiantes puedan participar de manera activa y significativa.

De esta manera, el aula se transforma en un animado restaurante donde cada plato esconde un código secreto. Los estudiantes, con sus habilidades recién adquiridas en codificación, se convierten en detectives culinarios. Equipados con sus hojas de respuestas y armados con sus conocimientos matemáticos, exploran el menú cifrado.

La emoción se palpa en el aire mientras los jugadores descifran los códigos y revelan las opciones gastronómicas ocultas. Con cada tirada de los dados o cada carta de operación, se desata una nueva aventura culinaria. La concentración se hace evidente mientras los estudiantes analizan los números, realizan cálculos mentales y buscan la correspondencia entre los códigos y los platos del menú.

La colaboración y el compañerismo se fomentan mientras los jugadores se ayudan mutuamente a resolver los enigmas. El aula se convierte en un espacio de intercambio de ideas y estrategias, donde cada estudiante aporta su propio punto de vista y conocimientos. La satisfacción de descifrar un código y descubrir un nuevo plato es compartida por todos, creando un ambiente de camaradería y celebración.

El juego no solo es una divertida forma de poner en práctica lo aprendido, sino también una oportunidad para desarrollar habilidades sociales y emocionales. Los estudiantes aprenden a trabajar en equipo, a resolver problemas de manera

colaborativa y a celebrar los logros de sus compañeros. Además, el juego fomenta la perseverancia y la capacidad de adaptación, ya que los desafíos pueden variar en dificultad.

### 3. Resultados

La experiencia de aprendizaje en programación ha arrojado resultados altamente positivos. Los estudiantes demostraron un gran entusiasmo y compromiso al desarrollar sus proyectos, lo cual se evidenció en su participación activa durante las clases y en la calidad de los trabajos finales.

Un estudio cuantitativo realizado al finalizar el curso reveló que el 95% de los estudiantes afirmó haber comprendido los conceptos básicos de programación y el 80% aseguró sentirse capaz de crear sus propios programas sencillos. Además, el 75% de los estudiantes indicó que había mejorado sus habilidades de resolución de problemas y pensamiento lógico.

Los testimonios de los estudiantes corroboran estos resultados. María, una de las participantes, comentó: "Antes de este curso, la programación me parecía algo muy complicado, pero ahora me siento más segura y creo que puedo seguir aprendiendo por mi cuenta". Por su parte, Juan expresó: "Lo que más me gustó fue trabajar en equipo y ver cómo juntos podíamos crear algo desde cero".

El análisis detallado de los proyectos desarrollados por los estudiantes también permitió identificar patrones interesantes. La mayoría de los proyectos demostraron un alto nivel de creatividad y originalidad, y muchos de ellos resolvieron problemas reales de su entorno. Esto sugiere que los estudiantes no solo adquirieron conocimientos técnicos, sino que también desarrollaron habilidades de pensamiento crítico y resolución de problemas.

Estos resultados positivos pueden atribuirse a diversos factores, como la metodología de enseñanza empleada, que hizo que el aprendizaje fuera divertido y motivador, y el apoyo brindado por los docentes. Además, el uso de herramientas y plataformas de programación intuitivas facilitó la comprensión de los conceptos y el desarrollo de los proyectos.

En conclusión, la experiencia de aprendizaje en programación ha sido un éxito rotundo. Los estudiantes han adquirido conocimientos sólidos en programación, han desarrollado habilidades valiosas para el futuro y han demostrado un gran potencial para seguir aprendiendo y creciendo en este campo.

### 4. Conclusiones

La programación desenchufada, al combinarse con la gamificación, ha revelado su potencial como una metodología de enseñanza altamente efectiva. Al introducir los conceptos de programación a través de actividades prácticas y juegos, los estudiantes han demostrado un mayor engagement y motivación. Esta metodología no solo facilita la comprensión de conceptos abstractos, sino que también fomenta el desarrollo de habilidades transversales como el pensamiento crítico, la resolución de problemas y la creatividad. La programación desenchufada, al desvincular el aprendizaje de la programación de la computadora, permite a los estudiantes centrarse en los fundamentos y construir una base sólida antes de adentrarse en entornos de programación más complejos.

En conclusión, la experiencia ha puesto de manifiesto la importancia de integrar la programación desenchufada y la gamificación en los procesos de enseñanza. Al transformar el aprendizaje de la programación en una experiencia lúdica y significativa, se ha logrado no solo mejorar el rendimiento académico de los estudiantes, sino también fomentar una cultura de innovación y curiosidad. Esta metodología pedagógica tiene el potencial de democratizar el acceso a la programación y de preparar a las nuevas generaciones para los desafíos del mundo actual, ya que, al hacer que la programación sea accesible y divertida, estamos empoderando a los estudiantes para que sean los creadores del futuro.

## Referencias

- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- Kafai, Y. B., & Resnick, M. (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world*. Lawrence Erlbaum Associates.
- Shute, V. J. (2008). Focus on formative computer game assessment. *Computers & Education*, 51(1), 1-13.

Versión Preliminar

# **“Cultivando el Futuro: Diseño y Aplicación de un Riego Automático en el Aula”**

Tomás Koch<sup>1</sup> y Yanina Swarynski<sup>2</sup>

<sup>1</sup> Escuela Yeshurun Torá, CABA, Argentina

<sup>2</sup> Escuela Yeshurun Torá, CABA, Argentina

[tomaskoch@yeshuruntora.edu.ar](mailto:tomaskoch@yeshuruntora.edu.ar), [yaninaswarynski@yeshuruntora.edu.ar](mailto:yaninaswarynski@yeshuruntora.edu.ar)

## **Resumen**

Esta experiencia se implementó en un 2do año de Nivel Secundario, en la materia Tecnología. Los docentes desarrollaron un proyecto pedagógico respetuoso del enfoque que propone el “Aprendizaje Basado en Problemas”. A través de una situación problemática real surgida en la escuela, se abordó un desafío pedagógico que implicó que los estudiantes debían diseñar un prototipo de regado automático, favoreciendo un tipo de trabajo grupal ligado a la “Cultura Maker”, en el sentido de crear, explorar, experimentar. El propósito docente estuvo ligado al diseño de una propuesta donde poner en práctica los contenidos teóricos abordados en la materia, a través de un desafío grupal, donde el trabajo en equipos, la colaboración, la formulación de hipótesis, la inventiva y la creatividad, fueran ejes centrales.

**Palabras clave:** Aprendizaje Basado en Problemas, Cultura Maker, Creatividad, Nivel Medio, Experiencia Docente.

## **1. Introducción**

En el área de tecnologías, se pretende abordar la problemática ambiental actual a través de un enfoque sustentable. Es decir, se busca ampliar la mirada acerca de la función que cumple el diseño o bien la creación técnica de lo artificial que incluye tanto artefactos como procesos, organizaciones, servicios, programas y métodos. Se intentará indagar acerca de cómo la tecnología puede ser pensada en términos no solo de eficiencia y eficacia en relación a los procesos, sino también ponerse al servicio de un bien social. Tener un impacto positivo en las sociedades, donde la dimensión ética forma parte desde la concepción del artefacto a diseñar. En este sentido, y situando el aporte que Israel vuela al área de ciencias y tecnologías en el centro de atención, se llevarán adelante estudios de casos en donde se vea directamente esa injerencia que tiene el país en dicho sector.

Greeneye Technologies, una empresa que desarrolla un sistema que usa inteligencia artificial para aplicar herbicidas solo donde es necesario, reduciendo el uso de químicos que dañan innecesariamente la tierra, contribuyendo así a una forma de agricultura regenerativa, es, por ejemplo, uno de los casos en cuestión. N-Drip, un sistema de riego automatizado que reduce considerablemente las emisiones de dióxido de carbono o metano, es otro caso aportado al estudio de este proyecto. O sea, se busca trazar una relación entre el desarrollo tecnológico de un país y la exportación que de esa tecnología hacen, con el diseño de soluciones largoplacistas a la problemática ambiental corriente.

Teniendo en cuenta los modelos que estas empresas proponen, ofreceremos una situación problemática a los estudiantes en la que se buscará diseñar una máquina de riego automático para luego construirla, posteriormente programarla y finalmente utilizarla y realizar un registro semanal durante un período de tres meses aproximadamente de modo que se pueda ir llevando a cabo un monitoreo de la humedad de la tierra en contacto con semillas que se pretende hacer crecer. En resumen, no solo se buscará mantener húmeda la tierra sino también hacer crecer una planta.

### Situación Problemática:

La escuela decidió armar una pequeña huerta, con algunos plantines de hierbas aromáticas para ser usado en el comedor. Durante el mes de octubre, debido a asuetos religiosos, varios días no habrán clases y por lo tanto, los cuidados de riego que las plantas necesitan no podrán ser atendidos.

Pregunta Problematizadora: ¿Cómo podemos resolver con ayuda de la tecnología las tareas de riego que la huerta precisa de forma automática?

## 2. Descripción de la Experiencia

### 2.1. Propuesta Didáctica

Objetivos

- Identificar a los productos tecnológicos como el resultado de procesos de diseño que surgen en respuesta a una compleja interacción entre intenciones, intereses, necesidades, restricciones y condicionamientos técnico-económico-sociales.
- Conocer los aspectos que caracterizan a las lógicas del pensamiento y el quehacer técnico, analizando procesos de intervención humana sobre el medio que equilibran criterios de eficacia, eficiencia y sustentabilidad social y ambiental.
- Desarrollar capacidades para identificar, analizar y resolver problemas que involucren operaciones sobre materiales, energía o información, proponiendo y seleccionando alternativas, planificando soluciones mediante estrategias y lógicas algorítmicas y heurísticas y evaluando los resultados obtenidos (\*).
- Desarrollar una mirada crítica y científico-metodológica que le permita a los alumnos incorporar nuevos hábitos de estudio y estrategias para el incremento de un pensamiento racional.

Contenidos:

- El concepto de diseño y su relación con la creación e innovación tecnológica: El diseño como creación de lo artificial. El diseño aplicado a la creación de mediadores entre el cuerpo humano y el entorno. El diseño aplicado a la mecanización y motorización de las tareas. El diseño aplicado a la automatización de las tareas y a la creación de sistemas de información (pensar > bocetar > diseñar > construir > programar > utilizar > registrar).
- La automatización.
- Tipos de sistemas automáticos: El control a lazo abierto por tiempo y por sensores. Exploración de diferentes tipos de actuadores, controladores y sensores. Entradas analógicas y digitales.
- La automatización como proceso de tecnicificación: La delegación de los procesos sobre la información.
- Introducción a la Robótica: la robótica en la vida cotidiana. Formas de programación.

Possibles articulaciones: Biología, Formación Ético Ciudadana, Historia, Geografía.

### 2.2. Contexto

La experiencia se desarrolla en la Escuela Secundaria Yeshurun Torá, en un 2do año A (24 estudiantes) y 2do año B (27 estudiantes). La población que asiste es de clase media, media alta.

### 2.3. La Experiencia

- Actividades:
  - Primer Encuentro: Se destinará una primera clase teórico - exploratoria presentando los dos casos de empresas israelíes especialistas en riego automático.

¿Qué es un sistema de control de lazo cerrado?

El control en lazo cerrado aparece ante la necesidad de corregir las desviaciones, de la salida frente a la referencia, de forma automática. Esto introduce un nuevo elemento en el sistema, un automatismo o procesador que se encargará de evaluar los valores del sensor y actuar en consecuencia.

Se analizan algunos ejemplos: puerta automática, aire acondicionado automático.

- Segundo Encuentro: En un segundo encuentro, se presentará al grupo total la situación problemática:

La escuela decidió armar una pequeña huerta, con algunos plantines de hierbas aromáticas para ser usado en el comedor. Durante el mes de octubre, debido a asuetos religiosos, varios días no habrán clases y por lo tanto, los cuidados de riego que las plantas necesitan no podrán ser atendidos.

Pregunta Problematizadora: ¿Cómo podemos resolver con ayuda de la tecnología las tareas de riego que la huerta precisa de forma automática?

Con el fin de pensar distintas hipótesis para responder la pregunta, se dividirá al grupo en pequeños equipos de trabajo compuestos por 2 o 3 integrantes.

Se les ofrecerán los siguientes materiales para el diseño de la máquina:

maceta  
tierra  
semillas  
sorbete  
palito de helado  
cinta bifaz  
cinta  
banditas elásticas  
placa microbit  
servo motor  
cables conectores  
una computadora



La propuesta de trabajo es de exploración de los recursos en un primer momento y luego de diseño de la máquina de riego.

- Tercer encuentro: presentaremos los diseños propuestos por los diferentes grupos. Hablaremos entre todos sobre distintos tipos de sensores, actuadores y controladores. Indagaremos sobre la programación de las placas Microbit.

¿Qué es un sensor? Tipos de sensores

### **Los sensores:**

Dispositivos que se utilizan para tomar datos de la realidad, transformando una magnitud física o química, en una magnitud eléctrica (generalmente), como el voltaje.

Los sensores pueden ser por ejemplo: de temperatura, intensidad lumínica, aceleración, inclinación, calor, presión, fuerza, torsión, humedad, etc.

¿Qué es un actuador? Tipos de actuadores

### **Los actuadores:**

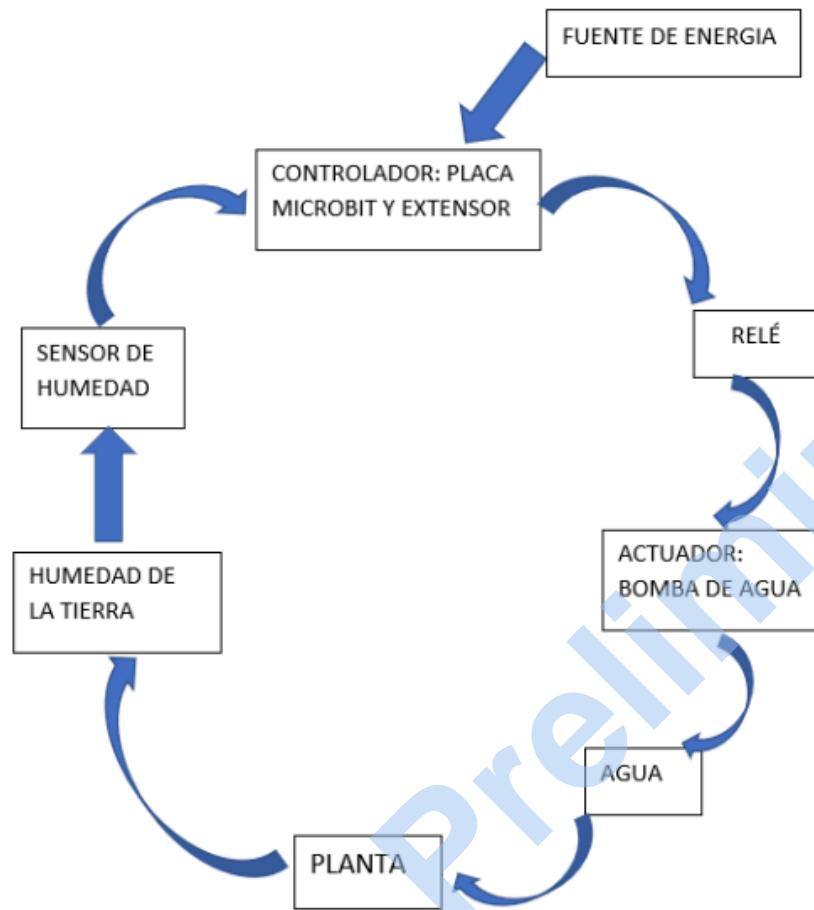
Dispositivo o herramienta que genera una acción cuando recibe la orden de un controlador.

Los actuadores pueden funcionar a partir de energía eléctrica, líquidos, gases o mecanismos. Pueden ser por ejemplo, una válvula, un pistón, un motor, un electroimán, etc.

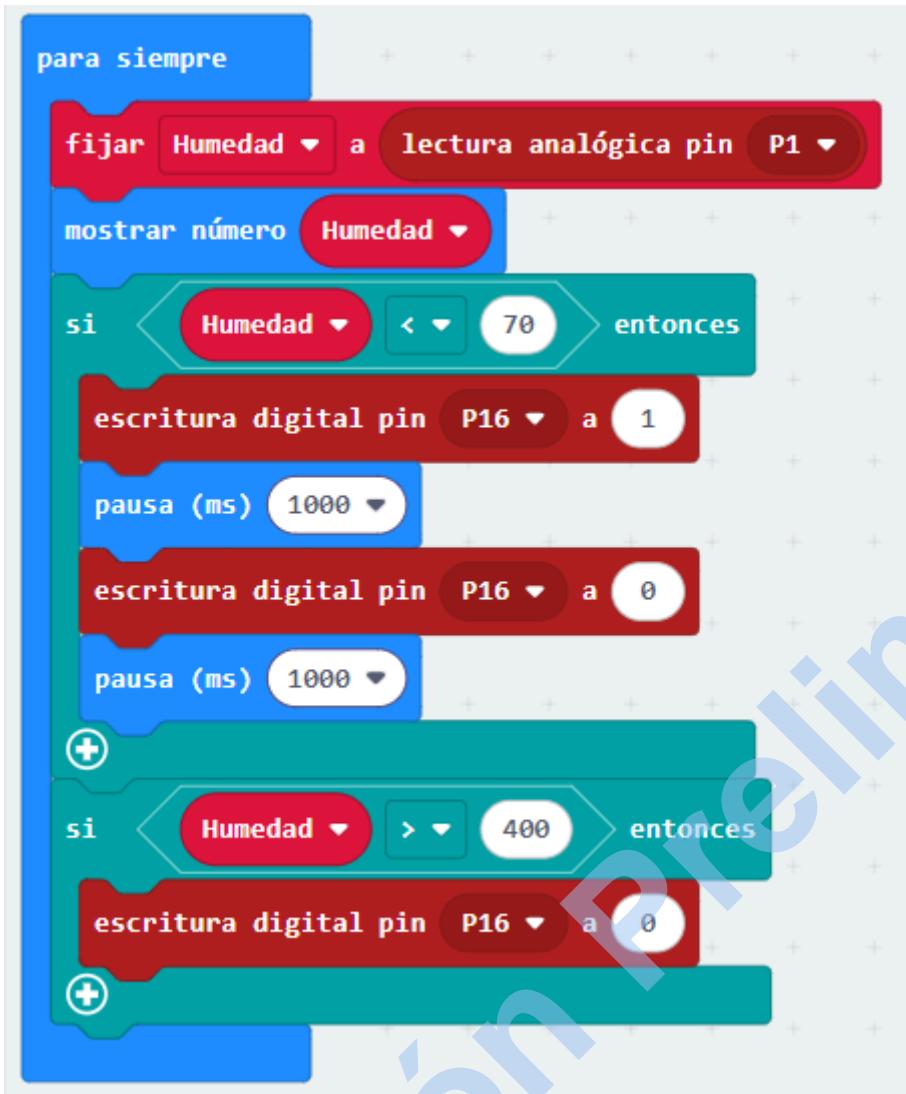
Analizamos el siguiente esquema:



Entre todos graficamos el diagrama para un sistema de riego automático:



Programamos la automatización:

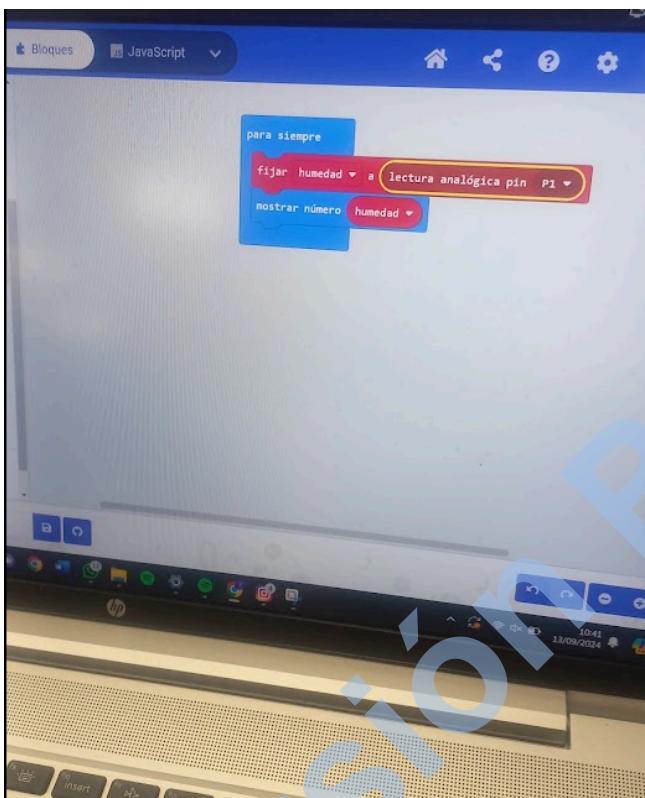
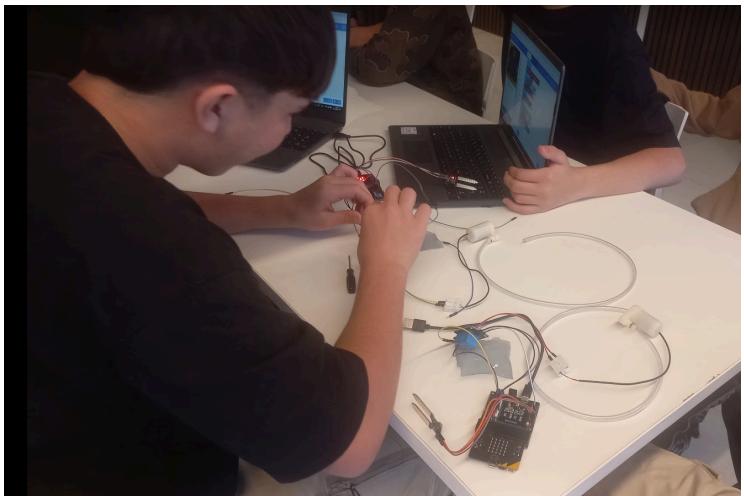


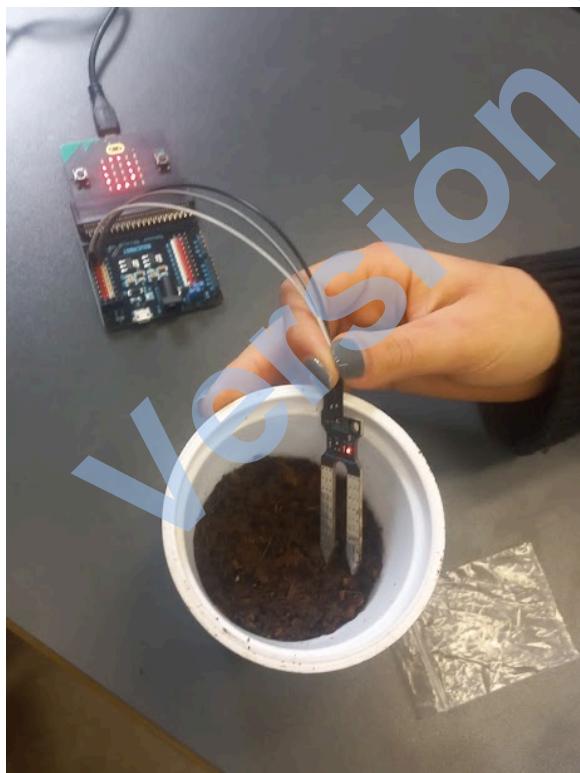
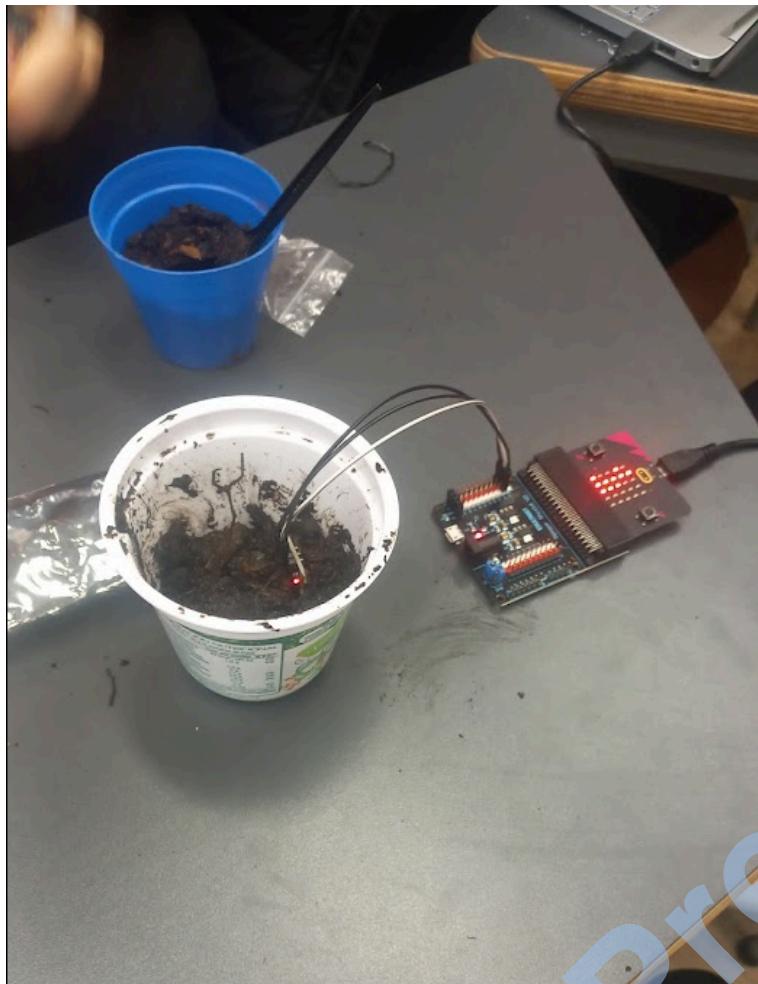
Exploramos con los estudiantes:

- El concepto de “variable”. La necesidad de crear una variable a la que llamamos “Humedad” para poder almacenar allí los datos que arroje el sensor de humedad.
- Reflexionamos sobre el bloque “lectura analógica pin P1” que hace referencia al sensor que está conectado a la placa. Conversamos sobre la diferencia entre entradas analógicas y digitales. Pensamos juntos por qué en este caso el sensor de humedad es una entrada analógica al sistema.
- Conversamos sobre el uso de los condicionales en la programación ¿Qué es una condición? ¿Cuáles son las condiciones que necesitamos para controlar el funcionamiento de la bomba de agua? Para ello, tomamos los valores que el sensor de humedad nos arrojó de una maceta con tierra seca y una maceta con tierra húmeda. Esos dos valores los usamos para crear las condiciones en las que la bomba se encenderá y se apagará.
- Analizamos con los estudiantes el bloque “escritura digital p16 a (0/1): encendido o apagado.

Con la aprobación de los docentes, podrán pasar al paso de construcción de la máquina.

- Cuarto encuentro: Construcción y testeо de la máquina.





### **3. Resultados**

Los resultados fueron satisfactorios. Los estudiantes lograron consolidar habilidades de colaboración desarrolladas en los pequeños grupos de trabajo. Surgieron dificultades en relación al armado de la máquina de riego, con aciertos y desaciertos que pudieron ir sorteando con la ayuda de los docentes, movimientos ligados al ensayo/error/revisión esperados en una propuesta que tenía como eje la experimentación y la formulación de hipótesis. Los tiempos pautados en la planificación se alargaron, teniendo que sumar dos clases al proyecto, para dar tiempo a los procesos de construcción respetuosos de los tiempos que requirió cada equipo. Algunos grupos arribaron antes que otros a los resultados. En estos casos, esos mismos equipos se sumaron a colaborar con los que aún estaban trabajando.

### **4. Conclusiones**

Si bien el proyecto requirió un tiempo mayor al pautado desde el comienzo de la planificación y una apropiación del encuadre de trabajo por parte de los estudiantes que también requirió tiempo (más cercano al espacio de taller que al del aula tradicional), la experiencia generó un compromiso de los estudiantes que se vieron motivados por el *défido* y el espacio para poner en juego saberes prácticos, un trabajo en equipo genuino y la capacidad de inventiva y creatividad propias de cada uno.

Fue interesante también el rol docente asumido durante el proyecto, ligado a un facilitador de aprendizaje que fue acompañando la propuesta que cada grupo llevó a cabo.

### **Referencias**

- Educación tecnológica : del control automático a la robótica : primer año. - 1a edición para el profesor - Ciudad Autónoma de Buenos Aires : Ministerio de Educación del Gobierno de la Ciudad Autónoma de Buenos Aires. Dirección General de Planeamiento Educativo, 2018.
- Gobierno de la Ciudad de Buenos Aires. Ministerio de Educación. Dirección General de Planeamiento e Innovación Educativa. Gerencia Operativa de Currículum Diseño curricular nueva escuela secundaria de la Ciudad de Buenos Aires, ciclo básico, disponible en <https://buenosaires.gob.ar/sites/default/files/media/document/2017/10/26/ad4a5c873f97638ecdfa20ccb54bf6ddb7551cf.pdf>

Objetivos de Desarrollo Sostenible. Naciones Unidas. Disponible en  
<https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>

# Integración de Herramientas de Desarrollo de Software y de Aprendizaje Automático en un Trabajo Integrador Final de Programación

Diana C. Vertiz del Valle<sup>1,2</sup>, Santiago Salinas Sosa<sup>1,2</sup>, Jordán F. Insfrán<sup>1,2</sup> y Javier E. Diaz Zamboni<sup>1,2</sup>

<sup>1</sup>Laboratorio de Informática y Computación Aplicada, FIUNER, Oro Verde, Argentina

<sup>2</sup>Equipo de cátedra Programación Avanzada, FIUNER, Oro Verde, Argentina

[diana.vertiz@uner.edu.ar](mailto:diana.vertiz@uner.edu.ar), [santiago.salinas@uner.edu.ar](mailto:santiago.salinas@uner.edu.ar), [jordan.insfran@uner.edu.ar](mailto:jordan.insfran@uner.edu.ar), [javier.diaz@uner.edu.ar](mailto:javier.diaz@uner.edu.ar)

## Resumen

En este trabajo describiremos una experiencia didáctica que consiste en la realización de un trabajo práctico integrador en el que las y los estudiantes diseñan y desarrollan un sistema de gestión de reclamos. Dada la complejidad del problema y la solución, el trabajo se realiza en grupos de dos personas, quienes deben integrar diversas herramientas para desarrollar el sistema, entre ellas: un *microframework* para la interfaz web, un sistema de control de versiones, entornos virtuales y un clasificador de aprendizaje automático. La experiencia está pensada para contribuir en las competencias genéricas de egreso de estudiantes de segundo año de la FIUNER. Se busca mejorar el aprendizaje en diseño y desarrollo de programas, aportando al desarrollo de habilidades técnicas en el uso de herramientas y recursos, así como desarrollar y fortalecer habilidades blandas. Los resultados de la experiencia revelan un progreso significativo de los estudiantes en la adquisición, comprensión y aplicación de conceptos, la colaboración en equipo y la gestión del trabajo. Sin embargo, también se evidenciaron algunas dificultades durante su implementación, lo que también destaca aspectos importantes a mejorar en la enseñanza de la asignatura.

**Palabras clave:** Herramientas de desarrollo de software, Aprendizaje automático, Programación orientada a objetos, Habilidades técnicas, Habilidades blandas.

## 1. Introducción

La enseñanza en la programación enfrenta desafíos significativos para reducir la brecha entre lo aprendido en el aula y las aplicaciones prácticas del mundo real. A menudo, los estudiantes adquieren conocimientos teórico-prácticos sin oportunidades suficientes para aplicarlos en contextos reales. Esto tiene como consecuencia una falta de motivación, dificultades en la comprensión profunda de los conceptos y habilidades limitadas para resolver problemas prácticos.

En el contexto del desarrollo del software moderno, los egresados de las carreras de Bioingeniería, Licenciatura en Bioinformática e Ingeniería en Transporte de la FIUNER es fundamental que adquieran habilidades técnicas, como el conocimiento sólido de al menos un lenguaje de programación de alto nivel, conocimientos de diseño de software y experiencia en el uso de herramientas diversas como el software de control de versiones, entornos virtuales, entornos de desarrollo integrados (IDE) y *frameworks*. Además, la integración de bibliotecas, APIs y componentes externos, incluyendo herramientas de inteligencia artificial y aprendizaje automático, se ha vuelto esencial en sus ámbitos profesionales. Sin embargo, las habilidades técnicas por sí solas, no son suficientes, también es fundamental que el paso por la universidad les aporte al desarrollo de habilidades blandas como la capacidad de comunicación clara y efectiva, la habilidad para trabajar en equipo y la capacidad de autogestión y aprendizaje autónomo. Es fundamental que las y los estudiantes integren de manera inteligente el uso de estas herramientas y recursos, dentro de un entorno didáctico que fomente el desarrollo tanto de habilidades técnicas como habilidades blandas, ya que ambas son indispensables en el entorno tecnológico actual.

En este trabajo se presenta una experiencia didáctica diseñada para introducir a los estudiantes en el uso de herramientas de desarrollo de software y de aprendizaje automático en una metodología de trabajo colaborativo. El enfoque

metodológico se centra en fomentar, motivar y habilitar que las y los estudiantes adquieran, profundicen y apliquen sus conocimientos en diseño y desarrollo de software para programar sus propios módulos, integrar módulos externos ya desarrollados, utilizando herramientas profesionales de desarrollo de software, en un marco de trabajo en equipo. Este enfoque no solo mejora sus habilidades técnicas, sino que también fortalece su capacidad de adaptación a nuevas tecnologías y facilita el desarrollo de habilidades blandas, aportando a las competencias genéricas de egreso.

## 2. Descripción de la Experiencia

La experiencia didáctica consiste en la realización de un trabajo práctico integrador final por parte de las y los estudiantes, que consiste en un proyecto de diseño y desarrollo de un sistema de gestión de reclamos para la Facultad de Ingeniería de la UNER. Para ello, en grupos de dos estudiantes deben diseñar y desarrollar el sistema, incluyendo una interfaz de usuario web por la cual los usuarios puedan realizar reclamos, que serán atendidos por cada oficina interveniente. Por otra parte, a través de la interfaz de usuario web, el sistema permite a determinados usuarios el tratamiento y cambio de estado de los reclamos desde cada oficina.

Un aspecto importante en este trabajo integrador es que los estudiantes deben incorporar al producto de software un modelo de aprendizaje automático. Para ello, les proporcionamos un modelo de clasificador entrenado para que lo integren en su aplicación. Este modelo se encarga de clasificar los reclamos en función de su contenido al área u oficina que atiende la queja, simplificando la tarea de programación. Por ejemplo, la queja “No hay agua en el dispensador” se asigna a la categoría “Maestranza”, otro caso puede ser “El sistema SIU Guaraní se cayó” se asigna a la categoría “Soporte informático”, y así sucesivamente con otros reclamos que puedan surgir. El modelo fue preparado y entrenado por el equipo docente para algunas categorías y se encuentra almacenado en un archivo *pickle* como un módulo externo y disponible en un [repositorio en GitHub](#) [1]. El repositorio incluye documentación y ejemplos de uso para que los estudiantes puedan seguir las instrucciones e integren el clasificador en sus proyectos de manera autónoma, tal como se haría en un entorno profesional.

El componente de aprendizaje automático es sumamente útil e interesante para mostrar a las y los estudiantes cómo tecnologías avanzadas pueden simplificar y agregar valor a sus soluciones de software. Además la integración del clasificador permite practicar la serialización y deserialización de modelos, y comprender cómo estas técnicas son utilizadas.

### 2.1. Propuesta Didáctica

**Duración:** 8 semanas

#### Objetivo general de la experiencia:

Desarrollar habilidades clave en el diseño y desarrollo de software, incluyendo la integración de herramientas profesionales, el uso de herramientas de aprendizaje automático y la colaboración efectiva en equipos de trabajo, para aportar a las competencias genéricas de egreso.

#### Habilidades que se quieren desarrollar:

En la siguiente tabla, se muestran las habilidades técnicas y las habilidades blandas que buscamos desarrollar:

Habilidades técnicas	Habilidades blandas
<b>Programación y diseño:</b> Conocimientos sólidos de un lenguaje de programación y diseño de software.	<b>Trabajo en equipo y colaboración:</b> Capacidad para la resolución de conflictos en el código al trabajar en un repositorio compartido.
<b>Uso de herramientas profesionales:</b> Conocimiento sobre control de versiones, uso de entornos virtuales, IDEs, pruebas unitarias y frameworks.	<b>Comunicación clara y efectiva:</b> Capacidad para expresar ideas y soluciones que sean fácilmente comprensibles para otros.
<b>Adaptación a nuevas tecnologías:</b> Capacidad para solucionar problemas que requieran la integración de herramientas externas: bibliotecas, APIs, herramientas de inteligencia artificial y aprendizaje automático, etc.	<b>Aprendizaje autónomo:</b> Capacidad para autogestionar su aprendizaje mediante el uso de recursos y documentación.  <b>Resolución de problemas:</b> Capacidad para resolver problemas abiertos aplicando Diseño y Programación Orientada a Objetos (POO)

## Contenidos:

Los contenidos cubiertos en la experiencia incluyen:

- Análisis y diseño orientado a objetos: aplicación de los principios del diseño orientado a objetos para estructurar e implementar proyectos usando diagramas UML para modelar relaciones entre clases.
- Desarrollo de pruebas unitarias: escritura y ejecución de pruebas unitarias utilizando el framework **unittest**.
- Polimorfismo: identificación desde el diseño de los casos donde se puede aplicar polimorfismo.

## Herramientas utilizadas:

Las herramientas utilizadas por los estudiantes durante la experiencia son las siguientes:

- Lenguaje de programación Python 3.
- IDE de desarrollo VSCode para escritura y depuración de código.
- Plantilla con estructura de carpetas para proyectos de Python: ayuda a mantener el código organizado.
- Git y GitHub para control de versiones y trabajo colaborativo en un repositorio compartido.
- Entornos virtuales utilizando el módulo **venv** para gestionar las dependencias de un proyecto.
- El *microframework* **Flask** para implementación de interfaces web.
- Módulo **coverage** utilizado para medir la cobertura de código de las pruebas unitarias.

## Metodología:

Adoptamos una metodología de aprendizaje basado en proyectos, la cual se centra en el desarrollo de habilidades y conocimientos aplicados a problemas del mundo real. Este enfoque involucra a los estudiantes en la solución de proyectos complejos que les permiten trabajar de manera autónoma para construir su propio aprendizaje obteniendo resultados tangibles generados por sí mismos [2]. Los estudiantes trabajan en un proyecto integrador donde aplican los conocimientos adquiridos y las herramientas que han utilizado hasta el momento. El desarrollo del proyecto se refuerza con las clases presenciales de teoría-coloquio y de práctica semanales.

## Actividades:

Las actividades por semana se detallan en la siguiente tabla:

Actividades de los Docentes	Actividades de los alumnos
<b>Semana 1: Presentación del proyecto y análisis de requerimientos</b>	
<ul style="list-style-type: none"><li>- Presentar el enunciado del proyecto con sus requerimientos funcionales y no funcionales.</li><li>- Guiar una discusión en clase sobre los posibles enfoques para la implementación de funcionalidades clave.</li></ul>	<ul style="list-style-type: none"><li>- Participar activamente en la discusión sobre propuestas de implementación.</li><li>- Comenzar a esbozar un posible diseño para abordar el proyecto, identificando clases y relaciones entre las mismas.</li></ul>
<b>Semana 2: Integración de bases de datos con SQLAlchemy</b>	
<ul style="list-style-type: none"><li>- Introducir a los estudiantes en el uso de SQLAlchemy para manejar bases de datos en sus proyectos.</li><li>- Proporcionar un ejemplo introductorio que cubra la creación de la base de datos, almacenamiento y recuperación de datos.</li></ul>	<ul style="list-style-type: none"><li>- Continuar con el diseño del proyecto, incorporando la creación y configuración de la base de datos.</li><li>- Trabajar en grupo para aplicar el ejemplo proporcionado por el docente adaptándolo al contexto del proyecto.</li></ul>
<b>Semana 3-4: Profundización en SQLAlchemy y configuración de un sistema de login de usuarios</b>	
<ul style="list-style-type: none"><li>- Proporcionar un ejemplo de implementación de un sistema de login de usuarios integrado con base de datos.</li><li>- Explicar cómo manejar relaciones entre tablas (uno a muchos y muchos a muchos) en SQLAlchemy y cómo implementarlas.</li><li>- Incentivar el uso de pruebas unitarias durante todo el desarrollo del proyecto.</li><li>- Realizar la primera revisión asíncrona del proyecto enfocada en los diagramas UML y prototipos de clases</li></ul>	<ul style="list-style-type: none"><li>- Continuar con el diseño y desarrollo del proyecto, incorporando el sistema de login de usuarios y relaciones entre tablas para finalizar el diseño de la base de datos.</li><li>- Trabajar en grupo para aplicar el ejemplo proporcionado por el docente adaptándolo al contexto del proyecto.</li></ul>

<p>subidos por los estudiantes en GitHub. Se proporciona retroalimentación a través de <i>issues</i>.</p>	
<b>Semana 5: Integración del clasificador como módulo externo</b>	
<ul style="list-style-type: none"> <li>- Presentar el clasificador entrenado como un proyecto almacenado en un repositorio en GitHub con su documentación y ejemplo básico para su integración.</li> <li>- Explicar el concepto básico de aprendizaje automático y cómo se aplica en el contexto del proyecto para clasificar los reclamos.</li> <li>- Introducir el concepto de serialización en Python y cómo se utiliza para guardar modelos entrenados en archivos <b>pickle</b>.</li> </ul>	<ul style="list-style-type: none"> <li>- Continuar con el desarrollo del proyecto, seguir la documentación y las instrucciones proporcionadas para integrar el clasificador en la aplicación.</li> <li>- Realizar pruebas para verificar que el clasificador funcione correctamente con los datos almacenados en la base de datos.</li> </ul>
<b>Semana 6: Desafío algorítmico</b>	
<ul style="list-style-type: none"> <li>- Presentar un desafío algorítmico que debe ser implementado en el proyecto. Explicar los pasos claves del mismo.</li> <li>- Proporcionar a los estudiantes un <b>anexo</b> a la guía de trabajo práctico integrador que detalle los pasos involucrados en la implementación del algoritmo.</li> <li>- Incentivar el uso de pruebas unitarias para verificar el funcionamiento del mismo.</li> <li>- Realizar la segunda revisión asíncrona enfocada en los avances del proyecto y la implementación de las funcionalidades.</li> </ul>	<ul style="list-style-type: none"> <li>- Continuar con el desarrollo del proyecto, utilizar el anexo proporcionado para desarrollar el algoritmo en su aplicación.</li> <li>- Realizar pruebas unitarias para verificar que el algoritmo funcione correctamente.</li> </ul>
<b>Semana 7-8: Finalización del proyecto</b>	
<ul style="list-style-type: none"> <li>- Proporcionar asistencia en la finalización del proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>- Completar el desarrollo de la aplicación, asegurando que todas las funcionalidades están correctamente implementadas.</li> <li>- Finalizar la documentación y preparar el proyecto para su entrega.</li> </ul>

### Evaluación:

- Se realiza un seguimiento regular de los grupos durante las actividades prácticas, proporcionando retroalimentación continua para guiar y facilitar el avance del proyecto.
- Se utiliza una planilla de seguimiento en la que el docente registra el cumplimiento de cada uno de los trabajos prácticos realizados durante el cursado de la asignatura.
- Se realiza una evaluación del proyecto integrador en grupo basada en criterios que incluyen: estructura del proyecto, análisis y diseño orientado a objetos, implementación de funcionalidades, documentación del código, integración de herramientas, e implementación de pruebas unitarias.
- Además de la evaluación grupal, se ofrece una instancia final opcional para la promoción. En la misma, los estudiantes son evaluados de manera individual con los mismos criterios de la evaluación grupal, pero con un enfoque adicional en los conceptos teóricos de la asignatura.

## 2.2. Contexto

La experiencia se lleva a cabo en la asignatura de Programación Avanzada con estudiantes de segundo año de diversas carreras de la Facultad de Ingeniería de la Universidad Nacional de Entre Ríos (FI-UNER). Programación Avanzada es de cursado cuatrimestral, es una materia obligatoria para las carreras de Licenciatura en Bioinformática y Bioingeniería y optativa para Ingeniería en Transporte. La carga horaria semanal de la asignatura es de 8 hs, repartidas en 3 hs dedicadas a una clase de teoría-colloquio, y 5 hs para la clase de práctica.

Se tienen tres trabajos prácticos de entrega obligatoria y cuya aprobación es condición necesaria para la promoción de la materia, en conjunto con la aprobación de dos evaluaciones parciales realizadas con cuestionarios autoevaluativos de

carácter teórico-práctico. El desarrollo de los trabajos prácticos se da a lo largo del cuatrimestre, fundamentalmente durante las clases prácticas, y deben realizarse en grupos de dos estudiantes. El tercer trabajo práctico es el más extenso y el que ocupa la mayor parte del tiempo del cuatrimestre y consiste en el trabajo integrador descrito previamente. Los alumnos tienen 8 semanas para completar el trabajo práctico integrador. Al igual que en los trabajos prácticos previos, los docentes realizamos un seguimiento durante las clases prácticas y se ofrecen horarios de consulta semanales. También se llevan a cabo revisiones asíncronas a través de GitHub, proporcionando retroalimentación mediante el uso de *issues*.

Se asume que los estudiantes han construido ciertos saberes previos antes de participar en la experiencia. Por un lado cuentan con conocimientos en programación básica estructurada con lenguaje Python, cuya experiencia la obtienen habiendo cursado la asignatura Fundamentos de Programación. Además, al momento de comenzar el trabajo integrador, las y los estudiantes han incorporado conceptos de diseño y programación orientada a objetos en nuestra asignatura, en base a las actividades teórico-prácticas previas.

### 2.3. La Experiencia

El desarrollo de la experiencia descrita se conduce de manera planificada, siguiendo las actividades detalladas en la propuesta didáctica, adaptándose a las necesidades y avances de los distintos grupos. A lo largo de las 8 semanas del trabajo integrador, las y los estudiantes son guiados en la aplicación de sus conocimientos técnicos y el desarrollo de habilidades blandas.

Desde el inicio del cuatrimestre, se establece una estructura de carpetas para los proyectos, promoviendo el trabajo en equipo y la colaboración mediante el uso de Git y GitHub para el control de versiones. Esta estructura permite trabajar de manera eficiente, facilitando la resolución de conflictos y la integración de diferentes partes del proyecto. La instrucción específica sobre estas herramientas al inicio del cuatrimestre, en las que se simulan posibles escenarios colaborativos, son esenciales para que adquieran confianza en el uso de Git y GitHub. Asimismo, se les instruye en la creación y uso de entornos virtuales para la gestión de dependencias.

Como mencionamos previamente en el contexto, las y los estudiantes antes de comenzar el trabajo integrador final, completan dos trabajos prácticos. Además de cubrir los contenidos de la asignatura (enfocados en la programación y diseño orientado a objetos), en estos trabajos se consolida el uso de las herramientas y el trabajo colaborativo. Para las herramientas como Flask, Git, GitHub y entornos virtuales, se proporcionan guías de seguimiento y repaso de los contenidos vistos en clase. Se les instruye en la escritura y ejecución de pruebas unitarias para verificar que cada componente de su código funcione correctamente de manera aislada. Además, se fomenta el uso de la documentación oficial de Flask y otras herramientas para promover la autonomía en el aprendizaje.

Como herramienta de evaluación y monitoreo del progreso de los estudiantes, se utiliza una planilla de seguimiento en la que el equipo docente registra los criterios necesarios que se deben cumplir para la aprobación de cada trabajo práctico. Esta planilla es visible a los estudiantes, quienes son responsables de revisar su evolución y verificar que el docente haya completado correctamente las evaluaciones. Esto incentiva la autogestión y la responsabilidad en el seguimiento de su propio progreso.

Durante las primeras semanas de la experiencia, los estudiantes se centran en el análisis y diseño orientado a objetos para estructurar el proyecto usando diagramas UML, posteriormente integran el diseño de la base de datos a este diagrama. Esta actividad inicial del proyecto, además de fortalecer sus habilidades técnicas, promueve el trabajo en equipo y la resolución de problemas en grupo permitiéndoles compartir ideas y estrategias en el diseño. Para esta primera etapa, se realiza una revisión asíncrona a través de GitHub para ofrecer retroalimentación, asegurando que el diseño del proyecto sea adecuado antes de iniciar la codificación.

La incorporación del clasificador entrenado como módulo externo, representa un desafío adicional para las y los estudiantes, que les permite familiarizarse con tecnologías avanzadas como el aprendizaje automático y conceptos de serialización. Este componente integrado en sus proyectos, siguiendo la documentación y ejemplos del repositorio, refuerza las habilidades de autogestión y aprendizaje autónomo. Además, la aplicación seleccionada demuestra cómo las tecnologías avanzadas pueden simplificar partes del software muy complejas desde el punto de vista algorítmico.

El siguiente componente que enfrentan los estudiantes en el desarrollo del proyecto, es el desafío algorítmico. Este desafío junto con la guía proporcionada, no sólo evalúa sus capacidades lógicas de programación, sino que fomenta su capacidad de resolución de problemas al desglosar un algoritmo complejo en pasos simples. En esta etapa se realiza la segunda revisión asíncrona a través de GitHub para hacer un seguimiento del código y guiar en la implementación de las funcionalidades.

Durante el tiempo de codificación del proyecto, se incentiva a los estudiantes a incorporar pruebas unitarias, para verificar el funcionamiento de los módulos implementados. Este enfoque busca que los estudiantes adquieran un entendimiento de la importancia de las pruebas para verificar la correcta funcionalidad del código y detectar errores de forma temprana, y que además adquieran habilidades prácticas en la automatización de ejecución de pruebas, en este caso haciendo uso del módulo *coverage*. Se expone la utilidad que posee este módulo para medir la cobertura del código garantizando que las pruebas abarquen, como mínimo, una determinada porción del código, pero comprendiendo las limitaciones que esto puede presentar.

La etapa de evaluación grupal se centra en asegurar que se cumplan tanto los requerimientos funcionales como los no funcionales del proyecto, siguiendo los criterios registrados en la planilla de seguimiento. Se espera que las y los estudiantes expresen y justifiquen de manera clara las decisiones de diseño adoptadas en su solución. En la evaluación final para la promoción, la o el estudiante debe primero explicar cuál es el problema que se planteó y cómo su aplicación busca resolverlo. Posteriormente, se evalúa que el estudiante demuestre una comprensión profunda del código implementado, aplicando los conceptos teóricos de la asignatura. La claridad de sus respuestas al explicar los conceptos aprendidos es un aspecto importante de la evaluación, asegurando que el estudiante no solo posea habilidades técnicas, sino también la capacidad para transmitir sus ideas de manera efectiva.

### 3. Resultados

La experiencia didáctica reveló aspectos significativos relacionados con el desarrollo de habilidades técnicas y blandas en los estudiantes. Se analizaron el progreso en la comprensión de los nuevos conceptos dados en la materia, la efectividad en la colaboración en equipo, el uso de herramientas y recursos proporcionados y la calidad del producto final.

El análisis y diseño orientado a objetos tuvo gran relevancia en el desarrollo del proyecto. A lo largo de la etapa de diseño, las modificaciones realizadas ayudaron a la comprensión de conceptos clave como la abstracción, modularidad, encapsulamiento y polimorfismo. El trabajo en grupo, repartiendo tareas y responsabilidades, permitió a los estudiantes comprender las ventajas de aplicar estos conceptos en su diseño al hacerlos más sencillos de actualizar y mantener.

El uso de Git y GitHub como herramientas para el trabajo colaborativo, demostró ser sumamente beneficioso para los estudiantes en el desarrollo de habilidades blandas. Al trabajar en un repositorio compartido, los estudiantes tuvieron que coordinarse y cooperar para que el proyecto avance, dividiendo las tareas de manera eficiente y gestionando el tiempo para cumplir con el plazo establecido. Además, la resolución de conflictos en el código fortaleció sus habilidades de comunicación y toma de decisiones, al tener que discutir los cambios y elegir entre las soluciones posibles.

El uso de GitHub por parte del equipo docente, facilitó la corrección fuera del aula ofreciendo una retroalimentación a través de *issues*. Esto permitió a los estudiantes aplicar las sugerencias de mejoras además de aprender a rastrear y solucionar errores para obtener un código funcional y corregido.

Una de las principales dificultades para los estudiantes fue la incorporación del clasificador entrenado al sistema de gestión de reclamos. Desde el equipo de cátedra esperábamos que los estudiantes implementaran algún grado de encapsulamiento, quizás mediante la creación de una clase propia contenedora de la instancia del clasificador, aplicando conceptos de agregación o incluso explorando alguna herencia. Sin embargo, observamos que estas expectativas no siempre se cumplieron. En muchos casos, la implementación no reflejó el nivel de abstracción y aplicación de POO que esperábamos. Esto podría deberse a que quizás este grado de manejo de abstracción y uso de la POO es a veces complejo de ver a simple vista para las y los estudiantes, ya que implicaba no solo entender conceptos complejos como la herencia y la agregación, sino también aplicarlos de manera práctica en un proyecto realista. No obstante, este proceso fue fundamental para que los estudiantes desarrollaran la habilidad de adaptación a nuevas tecnologías, ya que les exigió integrar herramientas externas complejas y resolver problemas técnicos asociados a esta integración.

A pesar de esta dificultad, se observó un progreso significativo en la capacidad de trabajo en equipo y colaboración. Las y los estudiantes demostraron una gran creatividad al abordar el problema del clasificador de reclamos, explorando diferentes estrategias para su integración efectiva. Esta flexibilidad y disposición les permitió encontrar soluciones innovadoras que reflejan un entendimiento profundo de la POO y una disposición creciente para aplicar conceptos avanzados de manera práctica. La opción de utilizar el clasificador proporcionado por la cátedra, aunque no obligatoria, resultó en una mejora notable en la eficiencia de los tiempos de desarrollo, demostrando la importancia de ofrecer herramientas que faciliten la producción sin limitar la autonomía de los estudiantes.

En la implementación del desafío algorítmico, al igual que en la integración del clasificador, esperábamos que los estudiantes aplicaran conceptos clave de POO, como la creación de clases que encapsulan comportamientos específicos y la utilización de composición para combinar funcionalidades. Sin embargo, estas expectativas no se cumplieron en la mayoría de los casos, evidenciando nuevamente que determinados conceptos relacionados a la POO, como la abstracción y composición, requieren desarrollar una profundidad conceptual y un nivel de competencia técnica apropiados para poder aplicarlos de manera más efectiva. Por otro lado, la implementación del algoritmo, siguiendo el anexo proporcionado, demostró la capacidad de los estudiantes para analizar y resolver problemas, aplicar pensamiento lógico y emplear sus habilidades técnicas de programación para traducir los pasos en código. Este anexo ofreció una herramienta para trabajar de manera más autónoma, sirviendo como referencia de la secuencia de pasos del algoritmo en caso de que la solución no produjera los resultados esperados. Además, fomentó la autoevaluación, permitiéndoles comparar su solución con los pasos del anexo y así identificar las secciones a mejorar.

Para superar las dificultades mencionadas en la aplicación de los conceptos de POO durante la implementación, fue necesario guiar a los alumnos en estos pasos, ayudándolos a comprender las ventajas de aplicar dichos conceptos en sus diseños. En este contexto, las revisiones asíncronas a través de GitHub fueron muy beneficiosas, permitiendo detectar y corregir estas deficiencias en los diseños de manera temprana.

El análisis del uso de pruebas unitarias reveló que, aunque incentivamos a los estudiantes a incorporar pruebas desde el inicio de la codificación, muchos pospusieron esta tarea hasta las etapas finales del desarrollo. Esto en muchos casos, resultó en la acumulación de errores que no se detectaron hasta fases avanzadas del proyecto, lo que dificultó la identificación y corrección de la fuente del problema. Una posible razón para que esto suceda podría ser la percepción de los estudiantes de que escribir pruebas es una tarea adicional que consume tiempo y esfuerzo, además de la presión por cumplir con los requerimientos dentro de un plazo limitado, lo que los lleva a priorizar la escritura del código sobre la implementación de pruebas unitarias. Sin embargo, haber enfrentado este problema resultó ser sumamente valioso para el aprendizaje, porque permite reforzar la importancia de las pruebas. Los problemas encontrados destacaron la necesidad de detectar errores de manera temprana, validando la recomendación de integrar las pruebas desde el inicio del proceso de codificación.

Por otra parte, el uso del módulo *coverage* para medir la cobertura de código mostró resultados diversos. Si bien algunos grupos se esforzaron por alcanzar altos porcentajes de cobertura, a menudo lo hicieron sin considerar la calidad de las pruebas. Esto demostró la necesidad de enfatizar que una alta cobertura no garantiza la calidad en el diseño de las pruebas unitarias.

La etapa de evaluación grupal demostró una correcta integración de las habilidades adquiridas a lo largo del cursado, incluyendo la capacidad de expresar y justificar decisiones de diseño, la habilidad en el uso de herramientas de desarrollo y control de versiones y capacidad de colaboración y comunicación dentro del grupo. Esto se reflejó en la forma en que los estudiantes justificaron las decisiones de diseño adoptadas en sus soluciones, en la correcta integración de herramientas diversas y en el cumplimiento de los requerimientos del proyecto. Sin embargo, una limitación importante detectada en esta etapa, es la falta de actualización de la documentación, lo que ocasiona que el diagrama UML no corresponda con el código final o que refleje parcialmente la estructura real del proyecto.

En la defensa individual para promoción, se observó en algunos casos dificultades para explicar claramente el problema planteado, esto parece estar relacionado con una falta de preparación o ensayo previo para esta etapa. A pesar de esto, también se notó una mejora en la forma en que los estudiantes se expresaron, utilizando términos y expresiones propias de la asignatura con mayor precisión y confianza. Esto reflejó un claro avance en su capacidad para internalizar y aplicar el lenguaje técnico específico del ámbito de la programación.

Como docentes, consideramos que uno de los resultados más valiosos de esta experiencia didáctica es el repositorio de código que cada equipo mantiene a lo largo de la asignatura. Para mostrar estos resultados, elaboramos un repositorio en GitHub [3] “[JADiCC-2024-Experiencia-Docente-Resultados](#)” que reúne una selección de algunos de los trabajos realizados por las y los estudiantes. En general, siguen una misma lógica de organización, poseen documentación elaborada por las y los estudiantes y permiten evidenciar el nivel de separación de tareas y responsabilidades en cada actualización del repositorio.

## 4. Conclusiones

La experiencia didáctica descrita en este trabajo permite identificar logros significativos en la formación de estudiantes de segundo año en las carreras de grado de la FIUNER. Uno de los principales logros es el aporte al perfeccionamiento de la capacidad de las y los estudiantes para integrar diversas herramientas de desarrollo de software en un proyecto realista, fortaleciendo tanto sus habilidades técnicas como blandas. La implementación de tecnologías como Git, GitHub, Flask y un clasificador de aprendizaje automático proporciona una mejor comprensión de cómo aplicar sus conocimientos y los conceptos de la asignatura en un caso práctico.

Sin embargo, también se identifican retos importantes, como la dificultad para aplicar los conceptos más avanzados de POO, la tendencia a posponer la implementación de pruebas unitarias hasta las etapas finales del desarrollo y la falta de actualización de la documentación del proyecto. Aunque la primera revisión asíncrona a través de GitHub permite la corrección de los diseños de manera temprana, se considera necesario reforzar la aplicación práctica de los conceptos clave de POO desde etapas más tempranas del cursado. En la segunda revisión asíncrona a través de GitHub se podría incluir el seguimiento de la documentación para fomentar la práctica de mantenerla actualizada en paralelo con la codificación. Respecto a la tendencia a posponer la implementación de pruebas unitarias, sería beneficioso incluir pruebas unitarias en la mayoría de los ejemplos de código que se enseñan en clase reflexionando sobre la calidad de las mismas, no sólo en términos de cobertura de código. De esta forma, los estudiantes verían las pruebas como una parte natural del desarrollo, no como una tarea adicional.

La integración del clasificador representa un componente desafiante que resulta ser una oportunidad para que las y los estudiantes se familiaricen con tecnologías avanzadas y aprendan a adaptar módulos externos a sus proyectos. Como propuesta de mejora, se podría considerar la incorporación de otras formas tecnológicas de integrar el clasificador, como a través de un servicio web o una API, lo que ofrecería a los estudiantes una visión más amplia de cómo integrar y utilizar tecnologías externas.

El repositorio grupal en GitHub es una herramienta beneficiosa para la organización y seguimiento del trabajo en equipo. Este repositorio facilita la colaboración y gestión del proyecto y actúa como recurso de documentación y evidencia del progreso de los estudiantes. La organización y estructura resultante del repositorio, refleja la habilidad de las y los estudiantes para trabajar de forma colaborativa.

Las instancias de evaluación confirman un avance significativo en el desarrollo de habilidades técnicas y blandas en las y los estudiantes, incluyendo la colaboración y comunicación dentro del equipo. Además, la mejora en la precisión y confianza al utilizar el lenguaje técnico de la asignatura refleja el progreso de los estudiantes a lo largo del cursado y el impacto positivo de la experiencia didáctica en la formación de los estudiantes.

Consideramos que esta experiencia cumple el objetivo propuesto de mejorar el aprendizaje de la asignatura Programación Avanzada, aportando al desarrollo de habilidades técnicas y blandas proporcionando además aspectos clave sobre los cuales implementar mejoras en la enseñanza de la asignatura.

## Referencias

- [1] FIUNER-LICA, «Clasificador de Reclamos». <https://github.com/FIUNER-LICA/Clasificador-de-Reclamos>.
- [2] Dirección de Investigación y Desarrollo Educativo Vicerrectoría Académica, I. T., «El método de proyectos como técnica didáctica». Monterrey: Tecológico de Monterrey de 2008. [En línea]. Disponible en: [https://campus.ingenieria.uner.edu.ar/pluginfile.php/24561/mod\\_folder/content/0/Monterrey\\_El\\_metodo\\_de\\_proyectos\\_como\\_tecnica\\_didactica%20%282%29.pdf](https://campus.ingenieria.uner.edu.ar/pluginfile.php/24561/mod_folder/content/0/Monterrey_El_metodo_de_proyectos_como_tecnica_didactica%20%282%29.pdf)
- [3] FIUNER-LICA, «JADiCC-2024-Experiencia-Docente-Resultados». <https://github.com/FIUNER-LICA/JADiCC-2024-Experiencia-Docente-Resultados>.

# Aprender a aprender, aprendiendo: Experiencias para y con otros

Mariana Frutos<sup>1</sup>, Fabio Andres Zorzan<sup>1</sup>, Sandra Edith Angeli<sup>1</sup>, Guillermo Fraschetti<sup>1</sup>

<sup>1</sup> Universidad Nacional de Río Cuarto, Río Cuarto, Córdoba, Argentina  
[mfrutos@dc.exa.unrd.edu.ar](mailto:mfrutos@dc.exa.unrd.edu.ar) - [fzorzan@dc.exa.unrd.edu.ar](mailto:fzorzan@dc.exa.unrd.edu.ar) - [sangeli@exa.unrc.edu.ar](mailto:sangeli@exa.unrc.edu.ar) - [fraschetti@dc.exa.unrd.edu.ar](mailto:fraschetti@dc.exa.unrd.edu.ar)

## Resumen

En este trabajo presentamos una experiencia entre Prácticas Socio Comunitarias (PSC) y la materia Bases de Datos de la UNRC que tienen como objetivo la tarea de aprender a aprender, aprendiendo.

Es de fundamental importancia el trabajo conjunto y lo proponemos e implementamos entre estudiantes del último año de la carrera de Analista en Computación y estudiantes de los Centros Educativos de Nivel Medio para Adultos (CENMA), para que gradualmente se pudieran ir logrando la adquisición de conocimientos en este caso respecto del diseño de Bases de Datos y su aplicación en la vida cotidiana.

Es valioso destacar la diferencia entre lo que una persona puede hacer individualmente y lo que puede llegar a hacer con otros, en un contexto que lo permita. Conceptos como el de compromiso y diferentes estrategias tanto para la enseñanza como para el aprendizaje pueden ayudar a pensar en los caminos a seguir, desde las aulas, más allá de ellas, con otros, en función de generar autonomía creciente y en pos de contribuir a la innovación educativa, tan ansiada y necesaria en estas épocas. Aunque puede resultar un tanto ambiciosos estos objetivos, consideramos que este tipo de experiencias pueden constituirse en un puntapié inicial.

**Palabras clave:** Bases de Datos, PSC, CENMA, Innovación Educativa

## 1. Introducción

La Universidad Pública tiene la obligación de formar profesionales altamente capacitados promoviendo el desarrollo de una conciencia ciudadana basada en valores como la solidaridad, el respeto y la inclusión entre otros.

La Universidad Nacional de Río Cuarto aprobó, mediante Resolución 322/2009 del Consejo Superior [10], la incorporación de prácticas socio-comunitarias en todas las carreras de la institución. Las mismas pueden incluirse en las currículas como módulos dentro de las asignaturas, seminarios, talleres o espacios de prácticas profesionales, según consideren los docentes a cargo.

Esta iniciativa dota de sentido y compromiso social a la Universidad, a la vez que promueve la formación socio-crítica y ciudadana de los futuros egresados. Los estudiantes, simultáneamente, aprenden los contenidos de las asignaturas de sus respectivos campos profesionales y científicos, y realizan este tipo de proyectos que contribuyen a la comprensión y resolución de problemas sociales, medio - ambientales y/o económicos.

Además, en el diseño, desarrollo y evaluación de estos proyectos, convergen diversos actores o instituciones: UNRC, centros educativos y de la salud, colegios profesionales, vecinales barriales, organismos públicos, Estado, medios de comunicación, entre otros, constituyendo verdaderas comunidades de aprendizaje.

Cabe destacar que la UNRC fue la primera Universidad Nacional que institucionaliza esta experiencia, lo que fue reconocido con el Primer Premio Presidencial “Prácticas Educativas Solidarias en Educación Superior 2010”. Le han seguido en esta iniciativa, con diversas modalidades, otras casas de altos estudios como la Universidad Nacional de Buenos Aires, Universidad Nacional de Mar del Plata, Universidad Nacional General Sarmiento, Universidad Nacional de Avellaneda, Universidad Nacional de Rosario.

Nuestra propuesta tiene entre sus objetivos principales los de ofrecer a la Universidad como una posibilidad cierta de desarrollo social y económico mediante las PSC en las cuales participamos. Estas prácticas involucran a los estudiantes de tercer año que forman parte de la asignatura Bases de Datos perteneciente al Departamento de Computación de la Facultad de Ciencias Exactas, Físico-Químicas y Naturales de la Universidad Nacional de Río Cuarto. Se trata de una experiencia que los y nos compromete con esas realidades al participar tanto en la planificación como en el desarrollo de la propuesta.

## 2. Descripción de la Experiencia

Las PSC son concebidas, en la misma normativa institucional que las crea, como comunidades de aprendizaje, donde estudiantes, docentes y actores de la comunidad aprenden juntos participando en una práctica común. Estas PSC pueden contribuir a reducir la exclusión social y promover la justicia y la equidad a través de la docencia, la investigación y la extensión universitaria.

El aprendizaje, en este contexto, sólo puede entenderse desde una visión social [1] [2]. La palabra *social* del latín *socialis*, significa perteneciente a una comunidad de personas. Lo social se asocia con la palabra *socius* (socio), la cual se refiere a un compañero (persona o institución), alguien con quien se establece una relación de ayuda o cooperación mutua. De este modo, se diría, lo que llamamos y se practica como servicio social tendría que ver fundamentalmente con las relaciones y modos de convivencia que se establecen entre entidades físicas o morales para generar un beneficio y crecimiento mutuo [3].

Las prácticas sociales educativas constituyen una estrategia pedagógica que, desde la función de docencia universitaria y a través de la articulación de las funciones sustantivas de la universidad -enseñanza, investigación y extensión- procura un espacio específico de formación integral, promoviendo que el compromiso social universitario no se realice escindido del acto educativo y del proyecto pedagógico de la universidad [4].

Se trata de un aprendizaje colaborativo que, desde una perspectiva sociocultural, se concibe como un proceso interactivo, dialógico y dialéctico donde las personas, trabajando en función de una meta común, pueden tener como resultado un producto más enriquecido y acabado que si se trabajara individualmente. Se construye un conocimiento compartido y socialmente distribuido. Supone responsabilidad individual dentro del grupo e interdependencia en un proyecto común. Es también un aprendizaje experiencial, situado, en contexto, producto de la acción y participación en prácticas con la comunidad.

### 2.1. Propuesta Didáctica

La propuesta consistió en la realización de una serie de encuentros presenciales en la Institución Educativa, en este caso el CENMA, en los cuales se desarrollaron clases con modalidad taller que tuvieron como objetivo fortalecer la formación en las ciencias de la computación.

#### ¿Cómo pensamos, planificamos y desarrollamos esta experiencia?

Uno de los objetivos de nuestro trabajo es que los estudiantes avanzados de la carrera puedan vivenciar una experiencia de docencia a la vez que puedan afianzar los conocimientos disciplinares específicos que van adquiriendo a lo largo del cursado de las materias; vincularse con pares con otra realidad social e incentivar a los estudiantes de la institución con la cual trabajamos a formar parte de la UNRC. Este “formar parte” adquiere sentido si se logra que comiencen a pensar en el hecho de que realizar trayectos educativos en esta Universidad podría ser factible.

El servicio social es una actividad que hace crecer al ser humano por medio de la donación a los demás; es por esto que el principal objetivo del equipo docente es motivar a los estudiantes a que sean parte de los proyectos. Algunos de los problemas que presenta hoy nuestra sociedad incluyen el individualismo y la falta de solidaridad, lo que está impactando negativamente a nuestro país y el mundo. Con esta práctica se intenta sumar un granito de arena para poder revertir tal situación. En síntesis, más allá de los objetivos planteados con los estudiantes de las carreras del Departamento de Computación, el mayor logro que se persigue es un compromiso social y profesional que adquirirán nuestros estudiantes avanzados en el marco de la práctica.

Desarrollar vocaciones en el ámbito de las ciencias de la computación, en estudiantes secundarios y puntualmente en este caso, en adultos, es un proceso que puede ser muy enriquecedor y las PSC son una estrategia y propuesta que consideramos ayudan a despertar el interés en el campo.

Para la selección de los contenidos, su secuenciación y evaluación, se tuvieron en cuenta diferentes criterios, como son: los destinatarios, la profundidad y complejidad de los mismos que se podía abordar, la coherencia y cohesión entre los mismos. La secuenciación fue trabajada desde lo más simple a lo más complejo. Aquí volvemos a recalcar que se trata de una familiarización e iniciación al diseño de Bases de Datos [5] [6] [7].

Respecto de las cuestiones que tienen que ver con lo didáctico, la enseñanza se dió de manera gradual y contemplando los ritmos de aprendizaje de los estudiantes al momento del desarrollo. También respecto de los tipos de tareas educativas propuestas, dado que algunas eran de resolución individual mientras que otras lo eran de manera grupal. Por otra parte, el hecho de plantear situaciones problemáticas cotidianas fue otro de los aspectos didácticos que tuvimos en cuenta para lograr procesos de aprendizaje significativos, al favorecer la comprensión de en qué situaciones concretas los estudiantes podrían aplicar los conocimientos adquiridos.

### **Actividades de articulación entre los proyectos que integran la red que se planificaron realizar.**

Cronograma de las mismas: Se previó una primera instancia de familiarización para que todos quienes participamos pudiéramos comenzar a apropiarnos del lugar, tanto alumnos y docentes de la materia, a través de meriendas compartidas y dinámicas de interacción con los alumnos de los CENMA.

Se revisaron las necesidades propias de cada Centro, se plantearon algunas propuestas y se emprendió el trazado de planes de acción, algunas intervenciones y la realización de talleres, tras una especie de diagnóstico de cada situación (cuánta experiencia tienen en relación a la temática, fortalezas, dificultades al momento de aprender, etc.)

Conjuntamente se trabajaron los conceptos de las materias que tienen vinculación con la tarea a desarrollarse en los Centros.

De esta manera, se fue construyendo la instancia compartida (PSC) a través de encuentros a manera de talleres conjuntos en torno a la configuración de una Base de Datos, su utilidad, cómo se va aprendiendo ese contenido, qué podemos ir haciendo para comprenderlo, etc., si pueden considerarse de utilidad para el presente y futuro, qué se va sintiendo, experimentando [8] [9] [10]. Sabemos que el trabajo debe ser continuo.

Más allá de los criterios de evaluación académicos de la materia involucrada en la PSC, también fué criterio de evaluación del proyecto muy importante el reconocimiento del aprendizaje- servicio como forma responsable y reflexiva de construir el conocimiento. La evaluación no se centra sólo en el aprendizaje de los contenidos teóricos sino en el uso de ellos para resolver o abordar las situaciones de la práctica.

## **2.2. Contexto**

El CENMA 24 y la vinculación de la PSC se realizó con la asignatura EVSO (Espacio de Vinculación con el Sector de la Orientación) de 3º año con orientación en Economía y Administración. La modalidad de Educación de Jóvenes y Adultos de esta Institución, está compuesta por un grupo de estudiantes heterogéneo, de diferentes edades (desde 18 años hasta más de 50 años), en el último año de esta modalidad (tercer año) nos encontramos con estudiantes que iniciaron sus estudios desde primer año en la institución y otros que se han sumado a la misma a partir de segundo o tercer año, por lo que había estudiantes por lo general los más jóvenes que proceden de la media común que son repitentes o han abandonado no hace mucho tiempo, y los más grandes que han retomado sus estudios después de un tiempo más prolongado. En este último caso, nos encontramos con madres y padres de familias algunos de ellos desempleados o en precariedad laboral, y otros con empleos formales. Podemos decir que todos llegan a esta modalidad con un objetivo en común que es el de concluir sus estudios secundarios y algunos de ellos con la intención de continuar estudios superiores.

La fortaleza de los recursos humanos, un clima institucional positivo, participativo y colaborativo, los intercambios con la comunidad, la disposición y administración de los recursos, etc., son elementos que nos permiten lograr resultados positivos y trabajar por aquellos que hay que mejorar, entendiendo que es un proceso de construcción continua que compromete a todos los actores institucionales. En consecuencia nuestras acciones hicieron hincapié en la implementación de nuevos formatos y acuerdos didácticos para fortalecer las prioridades educativas, la consolidación del trabajo en red y en equipo en procura de la inclusión y la regularidad en las trayectorias de los estudiantes.

## **2.3 La Experiencia**

La PSC se desarrolló por medio de distintos encuentros presenciales entre el CENMA, estudiantes y docentes de la materia Bases de Datos.

El aprendizaje en las PSC está atravesado por emociones: miedo, alegría, satisfacción, extrañeza, incertidumbre, marcando un antes y un después en la vida de los estudiantes participantes. O por lo menos esto es lo que intentamos, que los estudiantes del CENMA, no vean a la Universidad como algo lejano e inalcanzable sino todo lo contrario.

Uno de los desafíos de la PSC es que la secuencia de contenidos y actividades que se llevaron a cabo corresponden a las necesidades de intervención y no solo la lógica curricular de la disciplina. La propuesta curricular, en tanto, se organizó en función de las demandas del territorio y “sigue el pulso” de este, lo cual adquiere necesaria flexibilidad y gradualidad.

La planificación es abierta a diversos escenarios posibles y se ajustó durante su desarrollo.

## **2.4 Objetivos generales de esta Práctica Socio Comunitaria**

De aprendizaje de los alumnos y de articulación con la comunidad:

- Profundizar el acercamiento a ámbitos específicos de aplicación propios de la computación.
- Facilitar una revisión de las estrategias de aprendizaje que se despliegan en cada situación taller dictado por nuestros estudiantes, en pos de favorecer su identificación, mejora y manejo cognitivo y afectivo de la situación.
- Favorecer espacios de reflexión en torno a considerar la posibilidad de que los estudiantes del CENMA piensen en continuar con estudios universitarios o de educación superior.
- Involucrar a nuestros estudiantes en ámbitos socioeducativos donde en un futuro pueden desempeñarse como profesionales.
- Propender a que los estudiantes conozcan y trabajen activamente en realidades diversas en función de dar respuestas a necesidades reales de entornos sociales.

De los estudiantes del CENMA:

- Que se puedan familiarizar (en una etapa introductoria) con el Diseño de Bases de Datos.
- Que puedan valorar los resultados de un trabajo colaborativo y de interacción con docentes de la Universidad y con estudiantes avanzados de la Carrera de Computación.
- Que les permita reflexionar en la posibilidad de aplicar en los conocimientos disciplinares adquiridos para resolver alguna situación cotidiana, por ejemplo.

### 3. Resultados

Uno de los objetivos de esta PSC fue identificar los cambios o efectos de las mismas en los aprendizajes de los estudiantes. Las PSC habilitan, según los estudiantes que participaron, otros modos de aprender, diferentes a las formas de aprendizaje en el aula. Se trata de aprendizajes experienciales, compartidos, con sentido y que, por lo tanto, motivan a los estudiantes.

Los estudiantes comprenden el por qué y para qué de lo que aprenden. La experiencia vivida es el punto de partida y de llegada del aprendizaje; se construye desde la experiencia y la transforma.

El aprendizaje que acontece en las PSC es un aprendizaje colaborativo. El conocimiento se construye dentro de sistemas de actividad social involucrando a múltiples personas. Se aprende participando en comunidades de prácticas. Desde esta perspectiva social del aprendizaje, los actores sociales comparten un objetivo común, trabajan juntos, interactúan, y así amplían su conocimiento y su experiencia.

En síntesis, cuando el aula es también el territorio se producen nuevas formas de aprender que demandan nuevas formas de evaluar. Es un aprendizaje resultado de la participación social y de la acción en entornos socio-comunitarios donde los estudiantes se enfrentan a problemas de la vida reales.

Para los docentes, las PSC han significado cambios en sus actuaciones docentes en diversos sentidos: movilizando procesos reflexivos; modificando las concepciones de docencia; integrando o no, según los casos la docencia, la extensión y la investigación y transformando los modos de vinculación con los estudiantes.

Los estudiantes del CENMA:

- Adquirieron conocimiento sobre modelización y diseño de bases de datos con su respectiva aplicación práctica, especialmente usando el modelo Entidad-Relación.



*Figura 1.* Alumnos de la Materia y CENMA

- Adquirieron conocimiento y experiencia en la implementación de base de datos relacionales sobre un motor de base de datos actual como lo es MySQL.



Figura 2. Alumnos del CENMA

Los estudiantes de la materia Base de Datos:

- Afianzaron conocimientos y aplicaron por medio de la docencia temas sobre la modelización y diseño de bases de datos, especialmente usando los modelos Entidad-Relación y Relacional.
- Afianzaron conocimientos y adquirieron una mayor experiencia en la implementación de base de datos relacionales sobre motores de base de datos actuales.

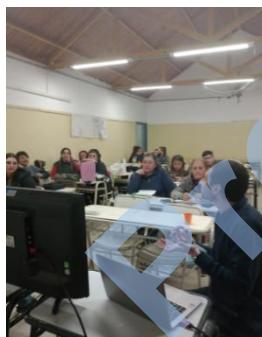


Figura 3. Alumnos de la Materia desarrollando el Taller

#### 4. Conclusiones

Uno de los grandes problemas que presenta hoy nuestra sociedad y el mundo incluyen el individualismo y la falta de solidaridad, generando un impacto totalmente negativo. Con esta PSC se intentó mediante la misma revertir tal situación. En la sección 2.4 se plantearon una serie de objetivos vinculados al aprendizaje y a la articulación entre los estudiantes de ambas instituciones intervenientes y con la comunidad, los cuales demostraron a través de las evidencias de prácticas, pruebas orales, entre otras actividades realizadas que fueron ampliamente alcanzados.

Pero, más allá de los objetivos planteados en la sección antes mencionada, con los alumnos de la materia Bases de Datos de las carreras del Departamento de Computación, el mayor logro que se consiguió es el compromiso social y profesional que adquirieron en el marco de la práctica los alumnos. Siendo el objetivo fundamental que el alumno promulgue la universidad inclusiva y que los estudiantes de la institución sepan que las puertas de nuestra casa de altos estudios están abiertas para ellos.

Muchos de ellos a través de las distintas clases dictadas veían a la Universidad como algo lejano, imposible. Se les informó en la visita a la Universidad la posibilidad de pedir becas, lo cual generó que muchos de ellos se pudieran ver como futuros alumnos de nuestra casa de altos estudios, algo que antes no estaba en sus planes debido a la situación social en la que viven.

La participación de los actores intervenientes fue muy positiva. Tanto el compromiso del equipo docente como de los alumnos y del personal de la Institución y sus estudiantes que nos recibieron en cada visita con mucho entusiasmo y predisposición.

Dejamos plasmadas aquí las valoraciones de uno de los dos estudiantes avanzados de la carrera que participaron de la experiencia:

**Pregunta 1:** ¿Cuál es tu valoración personal de la experiencia desarrollada en el marco de PSC? Te pedimos que reflexiones y detalles por favor:

\* Aspectos que consideras positivos en general y también aquellos aspectos para tu relación profesional desde el punto de vista disciplinar y muy especialmente desde la didáctica (¿cómo se enseña?) para la enseñanza del diseño de las bases de datos.

\* Dificultades encontradas durante todo el proceso (esto es, planificación de las clases e implementación de las mismas en la institución educativa).

*“Mi valoración de la experiencia es muy positiva. Me permitió no solo aplicar mis conocimientos en el diseño y enseñanza de bases de datos, sino también reflexionar sobre la didáctica y cómo abordar los temas de manera accesible para un alumnado ajeno al tema.”*

*Fue enriquecedor diseñar una clase que transformara un contenido técnico en algo comprensible y relevante para los estudiantes. Nos esforzamos en utilizar un lenguaje claro y analogías cotidianas para asegurar que los alumnos pudieran captar una noción general y significativa del tema. La colaboración con el grupo de trabajo fue sumamente importante.*

*Uno de los principales desafíos fue la coordinación dentro del grupo, especialmente para encontrar tiempos comunes que nos permitieran planificar y desarrollar las clases de manera efectiva. Afortunadamente, la disposición de todos los miembros facilitó superar estos obstáculos.*

*Otro desafío se presentó en la implementación de la clase, ya que optamos por un formato de presentación con diapositivas, el cual no es común en la institución educativa donde se llevó a cabo la práctica. La logística para conseguir y configurar el equipo necesario, como el proyector, tomó más tiempo del esperado, lo que resultó en perder tiempo de clases.*

*Finalmente, noté cierta dificultad para lograr una participación activa de todos los estudiantes. Si bien la mayoría mostró interés y entusiasmo, hubo un pequeño grupo al que resultó más difícil involucrar plenamente en la clase.”*

**Pregunta 2:** ¿Consideras que experiencias de este tipo podrían contribuir al desarrollo de vocaciones docentes? (Si-No  
¿Por qué?)

*“Sí, considero que experiencias de este tipo son importantes para el desarrollo de la vocación docente. Estas instancias brindan un espacio donde los alumnos pueden ejercer la enseñanza de manera práctica, participando en todo el proceso desde la preparación de la clase hasta su ejecución. Esta experiencia directa y práctica es clave para descubrir y desarrollar las habilidades para la docencia.”*

**Pregunta 3:** ¿Desde el punto de vista del proceso de enseñanza y de aprendizaje, cómo valorarías los conocimientos previos del grupo y los alcanzados luego del desarrollo de la PSC?

*“Considero que el grupo con el que trabajamos tenía un conocimiento muy básico sobre el tema de bases de datos. Aunque algunos habían escuchado el término, no comprendían completamente su significado o su aplicación. Al finalizar la experiencia, logramos que los estudiantes adoptaran un concepto más claro y preciso de lo que son las bases de datos. Además, pudieron entender cómo estas se integran en su vida cotidiana, lo que hizo que el concepto les resultara más cercano y relevante, viéndolo como un recurso accesible y presente en su entorno.*

*Es importante destacar que también logramos acercar un concepto clave de las ciencias de la computación a los estudiantes, ampliando su percepción y entendimiento de la disciplina.”*

A través de esta experiencia, podemos observar que la PSC permitió a los estudiantes descubrir, por un lado, el sentido social de la profesión para la que se preparan y, por el otro, vislumbrar otros espacios de trabajo no previstos en los perfiles profesionales hegemónicos.

Los docentes venimos trabajando con la vinculación de Prácticas Socio Comunitarias y la materia desde hace varios años. La valoración que como equipo docente hacemos de esta vivencia es sumamente positiva, sentimos que podemos “devolver” a la comunidad lo que la Universidad nos dio, y que fue una instancia en la que aprendimos muchísimo de los estudiantes que participaron de esta experiencia y de nuestros propios estudiantes avanzados de la carrera.

## Referencias

[1] M. Rafaghelli, “Las Prácticas de Extensión de Educación Experiencial como oportunidad para integrar docencia y extensión” Revista Nro. 6 +E Integración docencia y extensión 6.Ene-Dic, 2016.

- [2] N. CecchiI, J. Lakpnich, D. Perez,, A. Rotstein, “El compromiso social de la Universidad Latinoamericana del Siglo XXI. Entre el debate y la acción”, . IEC. CONADU, 2019
- [3] A. Lechuga, M. Valdez Borroel, “El servicio social como recurso didáctico para intervenir la realidad social”, Zincografia vol. 2 nro. 4, 2018
- [4] F. Erreguena, “Las prácticas sociales educativas en la universidad pública: una propuesta de definición y esbozo de coordenadas teóricas y metodológicas”, 2020
- [5] Silberschatz, Korth, Sudarshan. McGraw Hill Company, *Database System Concepts*, 7th, Edition, 2019.
- [6] Silberschatz, Korth, Sudarshan. McGraw Hill Company, *Fundamentos de Base de Datos*, 5th Edition, 2006.
- [7] Silberschatz, Korth, Sudarshan. McGraw Hill Company, *Data Base Systems Concepts*, 4th Edition, 2002.
- [6] Ullman. Benjamin Comings, *Principles of Database Systems*, 1982.
- [7] Elmasri, Navathe. Addison Wesley, Fundamentals of Database Systems, 1994.
- [8] Addison Wesley, *An introduction to Database Systems*, Vol 1 C.J. Date.
- [9] Addison Wesley, *An Introduction to Database Systems*, Vol 2 C.J. Date.
- [10] Resolución del Consejo Superior de la UNRC N° 322 <http://www.unrc.edu.ar>

# **Enfoque Didáctico para la Integración Transversal de Análisis y Diseño de Sistemas e Ingeniería de Software en la Carrera de Analista**

Ariel Arsaute<sup>1</sup>, Marcelo Uva<sup>1</sup>, Daniela Solivellas<sup>1</sup>, Franco Brusatti<sup>1</sup>, Marcela Daniele<sup>1</sup>

<sup>1</sup>Universidad Nacional de Río Cuarto, Río Cuarto, Córdoba, Argentina  
[{arsaute, uva, dsolivellas, fbrusatti, marcela}@dc.exa.unrd.edu.ar](mailto:{arsaute, uva, dsolivellas, fbrusatti, marcela}@dc.exa.unrd.edu.ar)

## **Resumen**

En este trabajo se presenta una propuesta didáctica con modalidad taller, que funciona como eje integrador para el desarrollo de las asignaturas “Análisis y Diseño de Sistemas” e “Ingeniería de Software”, correspondientes al tercer año de las carreras de Computación de la Facultad de Ciencias Exactas, Físico-Químicas y Naturales de la Universidad Nacional de Río Cuarto. Este trabajo ya fue presentado en una edición anterior, pero ha sido actualizado para la nueva edición del JADiCC 2024.

El Taller AyDS-IS se destaca por su enfoque en la transversalidad de contenidos, permitiendo una integración efectiva con otras materias del plan de estudios. Esta propuesta facilita la articulación de los conocimientos adquiridos en diversas asignaturas, promoviendo un aprendizaje holístico y aplicado mediante el desarrollo y gestión de un proyecto de software basado en el trabajo colaborativo y sostenido durante todo el año lectivo.

Dentro del marco del Taller AyDS-IS, se aborda la solución genérica a un problema mediante el desarrollo de un producto de software. Este proyecto permite a los estudiantes aplicar metodologías ágiles y diversas tecnologías actuales. Los resultados han sido muy positivos, logrando que los estudiantes adquieran una experiencia quasi-real y culminen el Taller AyDS-IS con aprendizajes significativos. Esto les permite abordar con mayor agilidad sus trabajos finales, potenciar su graduación e inserción laboral, y fortalecer la relación entre la Universidad y el medio socio-productivo.

**Palabras clave:** Ingeniería de software; Análisis y diseño de sistemas; Desarrollo de software.

## **1. Introducción**

La propuesta didáctica presentada en esta experiencia áulica tiene como objetivo principal fortalecer la integración curricular y los enfoques interdisciplinarios en la formación de los estudiantes de las carreras de Computación de la Universidad Nacional de Río Cuarto (UNRC). El Taller AyDS-IS, desarrollado en las asignaturas cuatrimestrales y correlativas denominadas “Análisis y Diseño de Sistemas” de 180 horas e “Ingeniería de Software” de 170 horas, busca articular, integrar y aplicar los contenidos brindados a los estudiantes en un contexto práctico y cercano a la realidad del desarrollo de software en la industria.

Este taller se enfoca en la transversalidad de los contenidos curriculares, permitiendo que los estudiantes adquieran y apliquen conocimientos de diversas áreas a lo largo de su formación. El trabajo propuesto implica el uso de metodologías ágiles, tecnologías y herramientas actuales para el desarrollo de software y la gestión de proyectos. En los últimos años, el enfoque principal ha sido el desarrollo de ambientes frontend y backend necesarios para la implementación de aplicaciones web. Este enfoque interdisciplinario y práctico no sólo consolida los conocimientos adquiridos en distintas materias, sino que también prepara a los estudiantes para enfrentar desafíos reales en su futura vida profesional.

La ejecución del taller se divide en dos etapas. La primera, en la asignatura “Análisis y Diseño de Sistemas”, se focaliza en las actividades establecidas por una metodología ágil que guía la construcción de un producto de software, incluyendo

todas las etapas del ciclo de vida de desarrollo, desde la ingeniería de requerimientos [17] hasta la prueba del producto. En la segunda etapa, en la asignatura “Ingeniería de Software”, se incluyen las actividades necesarias para una gestión exitosa del proyecto. Esto implica trabajar con técnicas y tecnologías que incorporan actividades como la planificación del proyecto de software, incluyendo estimación, planificación, análisis de riesgos, gestión de la calidad y gestión del cambio.

El equipo docente, con una amplia experiencia que supera la década en el área, introduce actualizaciones y nuevos retos constantemente, lo que convierte al taller en un espacio dinámico y en constante evolución, adaptado a las necesidades de la industria. Esto contribuye a la formación de los estudiantes, ofreciendo un entorno de aprendizaje moderno y alineado con las exigencias 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 aplicar sus conocimientos en diversas áreas temáticas estudiadas a lo largo de la carrera. Esto incluye la resolución de problemas, bases de datos, diseño de algoritmos, y otras disciplinas, fomentando una comprensión integral e interdisciplinaria que enriquece su formación y los prepara para enfrentar desafíos reales en su futura vida profesional.

El trabajo colaborativo es una pieza clave del taller, ya que permite a los estudiantes trabajar en equipos de desarrollo, estableciendo objetivos comunes, analizando posibles soluciones, intercambiando conocimientos, resolviendo conflictos, fundamentando decisiones, y asumiendo diferentes roles definidos por la metodología aplicada, entre otros. Esta experiencia es esencial para que los estudiantes desarrollen competencias interpersonales y sociales fundamentales, como la comunicación efectiva, la colaboración en equipo y la capacidad de resolver problemas de manera eficiente. Estas habilidades son indispensables para la formación de equipos de trabajo cohesionados y exitosos en su futura vida profesional.

El Taller AyDS-IS se lleva a cabo con un enfoque dinámico y exploratorio, que motiva a los estudiantes a adquirir estrategias para mantenerse actualizados en un campo muy evolutivo, como es la industria del software. La capacidad de adaptarse y aprender nuevas tecnologías es crucial en el mundo actual, fomentando así el desarrollo de estas habilidades necesarias para mantenerse competitivos en el mercado laboral.

A continuación, se detalla la propuesta, describiendo el contexto de aplicación y los contenidos abordados. Seguidamente, se presenta la metodología de trabajo junto con las técnicas, tecnologías y herramientas utilizadas. Finalmente, se exponen algunos resultados observados en las diversas ediciones del taller.

## 2. Contexto y Desarrollo de la Propuesta

A continuación, se detalla el contexto en el cual se implementa esta propuesta, junto con los contenidos curriculares abordados.

### 2.1. Contexto

La propuesta didáctica se implementa en las asignaturas "Análisis y Diseño de Sistemas" e "Ingeniería de Software" de la Universidad Nacional de Río Cuarto (UNRC), esenciales para la formación en computación y desarrollo de software. "Análisis y Diseño de Sistemas" se ubica en el primer cuatrimestre del tercer año, mientras que "Ingeniería de Software" es su correlativa en el segundo cuatrimestre. El equipo docente está jerárquicamente compuesto por una Profesora Adjunta, responsable del desarrollo de los contenidos teóricos, una profesora con cargo de Jefa de Trabajos Prácticos y dos licenciados Ayudantes de Primera encargados de supervisar, coordinar y llevar adelante las actividades prácticas. Además, cuenta con un licenciado Ayudante de Primera dedicado exclusivamente a coordinar los equipos de estudiantes en el taller. Este último cuenta con una extensa trayectoria profesional en el desarrollo de software industrial, lo que facilita una continua actualización tecnológica en los contenidos educativos. El taller es desarrollado por su coordinador con la colaboración de dos ayudantes que también trabajan en los prácticos de la materia, lo que permite relacionar e integrar los contenidos prácticos con el taller.

### 2.2. Contenidos

Cada año se propone una situación problemática que debe ser abordada mediante una solución informática a través de un proyecto de software, el cual abarca todas las etapas del ciclo de vida del desarrollo de software, así como las actividades de gestión necesarias. Por ejemplo, en las ediciones de 2023 y 2024, se planteó el desarrollo de un sistema educativo basado en una 'Trivia Interactiva'. En este tipo de trivia o juego, se describe claramente la modalidad de la actividad, que incluye preguntas y respuestas en un formato dinámico y participativo.

Cada equipo decide el dominio o área temática que desea impartir. El sistema debe permitir gestionar usuarios, proporcionar contenido educativo sobre el tema elegido y luego ofrecer funcionalidades de evaluación para medir el grado de aprendizaje alcanzado por los usuarios. La libertad de elección del tema entusiasmó enormemente a los equipos quienes desarrollaron una amplia variedad de sistemas educativos. Entre los temas seleccionados se incluyen 'lengua de

señas', 'artes culinarias alrededor del mundo', normas de tránsito, lenguajes de programación y enseñanza de matemáticas, entre otros.

Cada equipo se permite 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 y mejorar 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. En la edición 2024, se incluyó la teoría de Product Discovery como una fase crucial en el desarrollo de software. Esta fase se centra en identificar y validar las necesidades del usuario y las oportunidades del mercado antes de comenzar con la construcción del producto. El objetivo principal es asegurar que el producto final resuelva problemas reales para los usuarios y tenga un ajuste adecuado en el mercado.

El proyecto se desarrolló con el **framework Sinatra** [14], que facilitó un desarrollo web ágil. Ruby fue el lenguaje principal [13], permitiendo un desarrollo eficiente y flexible. La gestión del proyecto se realizó con Pivotal Tracker, mientras que **GIT** y **GitHub** se encargaron del control de versiones [12]. Estas herramientas se integran para asegurar el seguimiento y control de las actividades. La calidad del código se mantuvo mediante el uso de gemas de RSpec para pruebas, lo que permitió identificar fallas de manera temprana. La combinación de estas tecnologías fue clave para el éxito del proyecto.

Las herramientas y tecnologías empleadas en las ediciones 2023 y 2024 son similares a las de años anteriores, aunque se han añadido nuevas herramientas para mejorar la calidad del software. Entre ellas, destaca SimpleCov, una gema que mide el grado de cobertura de los tests. Actualmente, el taller 2024 está en curso y promete seguir incorporando nuevas herramientas.

El **Manifiesto Ágil** establece los principios fundamentales que guían el desarrollo de software ágil, los cuales fueron adoptados durante el proyecto [1]. Este enfoque permitió una mayor adaptación a los cambios y la entrega rápida de valor.

Además, el uso de **Active Record** [2] y **Sinatra Active Record** [3] como ORM facilitó la persistencia de datos en las aplicaciones desarrolladas. Esto permitió una gestión eficiente de las bases de datos sin tener que escribir consultas SQL manualmente.

La integración de **Docker** [4] fue esencial para el entorno de desarrollo, asegurando que los equipos pudieran trabajar en entornos consistentes. **HTTP/1.1** [5] se utilizó como protocolo principal de comunicación para las aplicaciones web.

El uso de **RVM (Ruby Version Manager)** [6] permitió a los equipos gestionar diferentes versiones de Ruby a lo largo del proyecto, mientras que **Bundler** [7] facilitó la gestión de dependencias y gemas en los proyectos.

El enfoque de desarrollo también estuvo alineado con los principios de **Extreme Programming (XP)** [8], que enfatizan la adaptabilidad, la colaboración y la entrega continua de valor.

El Taller AYDS-IS ofrece una amplia gama de temas vinculados al desarrollo de software, basados en los contenidos curriculares de las asignaturas 'Análisis y Diseño de Sistemas' e 'Ingeniería de Software', abordados tanto de manera teórica como práctica. Los temas principales del taller incluyen:

### 2.2.1 Product Discovery (Implementada en la edición 2024)

Esta teoría fue aplicada en las actividades preliminares al trabajo de desarrollo en el taller. Se ejecutaron diversos pasos para llevar a cabo el Product Discovery de manera efectiva [9][10]:

- Investigación de Usuarios: Se realizaron entrevistas y encuestas para entender a los usuarios, sus problemas y necesidades. Estas actividades fueron resueltas en conjunto con los docentes de la cátedra, quienes aportaron su experiencia y conocimientos para guiar el proceso de investigación y asegurar que se cubrieran todos los aspectos relevantes.
- Generación de Ideas: Se llevaron a cabo sesiones de brainstorming (lluvia de ideas) con equipos multidisciplinarios para generar múltiples soluciones posibles.
- Prototipado y Pruebas: Se crearon prototipos rápidos que fueron probados con usuarios reales para obtener feedback temprano.
- Validación del Mercado: Se evaluó la viabilidad del producto en el mercado a través de análisis de la competencia y estudios de mercado.
- Iteración: Se mejoraron las ideas y prototipos basados en el feedback recibido hasta llegar a una solución viable.

Estos pasos aseguraron que el producto final no solo fuera técnicamente sólido sino también relevante y valioso para los usuarios. Aplicar Product Discovery en las etapas iniciales del proyecto de desarrollo de software permitió reducir riesgos, optimizar recursos y asegurar una mejor experiencia de usuario.

La combinación de estos conceptos proporciona a los estudiantes una comprensión integral y práctica del desarrollo de software. Al aprender sobre la gestión de proyectos, planificación y estimación en desarrollos ágiles, los estudiantes obtienen herramientas para organizar y dirigir proyectos de manera efectiva, garantizando que se cumplan los plazos y objetivos establecidos. La integración de patrones de diseño mejora la calidad, eficiencia y flexibilidad del código, mientras que el uso de UML permite documentar y especificar sistemas de forma clara y estructurada.

Además, la gestión de configuración de software enseña a controlar y rastrear cambios en el código, utilizando herramientas y flujos de trabajo estándar en la industria, como Git y sus modelos de ramificación. El enfoque en testing y la generación de test suites asegura la robustez del software mediante la identificación temprana de errores, mientras que el análisis de métricas orientadas al proceso, al producto y a las personas permite una evaluación precisa del progreso del proyecto, la calidad del software y el rendimiento del equipo.

Juntos, estos conceptos proporcionan a los estudiantes una formación completa que abarca desde la planificación y el diseño hasta la implementación, evaluación y gestión del software, preparándolos para enfrentar los desafíos del desarrollo de software en un entorno profesional.

## **2.2.2. Metodologías Ágiles (particularmente SCRUM) [9][12]**

En el ámbito actual del desarrollo de software, las metodologías ágiles han adquirido gran relevancia por su capacidad para adaptarse a cambios rápidamente y entregar productos de alta calidad en plazos reducidos [11]. SCRUM, una de las metodologías ágiles más prominentes, se explora en profundidad en el taller. Se estudian sus principios y prácticas para que los estudiantes adquieran un entendimiento sólido sobre su aplicación efectiva.

## **2.2.3. El Lenguaje de Modelado Unificado (UML) [17]**

Este lenguaje ofrece una variedad de modelos gráficos para documentar y especificar sistemas en diferentes etapas del ciclo de vida de un producto de software. Durante la fase de análisis del proyecto, se utilizan diversos diagramas UML, como Diagramas de Clases, Diagramas de Objetos, Diagramas de Actividades y Diagramas de Secuencia, entre otros. Estos diagramas sirven como artefactos que describen las clases del sistema, sus relaciones y atributos, así como las instancias y sus interacciones en momentos específicos del proceso.

## **2.2.4. Patrones de Diseño**

Los patrones de diseño son soluciones validadas y reutilizables para problemas comunes en el desarrollo de software. En el taller, se aplican estos patrones y se fomenta su uso para optimizar diversos aspectos del desarrollo del software. La implementación de patrones de diseño mejora la calidad del código al proporcionar soluciones robustas y probadas, incrementa la eficiencia al evitar la reinención de soluciones, y facilita la flexibilidad del sistema al permitir adaptaciones y extensiones sin afectar otras partes del código. Además, contribuye a la mantenibilidad del software, al hacer que el código sea más comprensible y fácil de mantener, y promueve la consistencia en el desarrollo, facilitando la colaboración y la integración entre componentes del sistema [8] [11].

## **2.2.5. Testing y Generación de Test Suites**

La calidad del software es crucial, y el testing desempeña un papel fundamental en su aseguramiento. En el taller, los estudiantes aprenden a generar test suites basadas en los criterios de cobertura de caja blanca, que se estudian en las asignaturas de Análisis y Diseño de Sistemas. Esta práctica garantiza que diversas partes del código sean evaluadas, lo que contribuye a mejorar la calidad y robustez del software.

Además, para optimizar la eficiencia y la consistencia 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, facilitando así la detección temprana de errores y la mejora continua del software.

## **2.2.6. Gestión de Proyectos, Planificación y Estimación en Desarrollos Ágiles**

Una gestión efectiva de proyectos es crucial para el éxito en el desarrollo de software. En el taller, se abordan técnicas de planificación, como los diagramas de PERT y Gantt, que ayudan a visualizar y organizar las tareas del proyecto. También se exploran técnicas de estimación propias de las metodologías ágiles [10], como el Planning Poker, para asegurar el cumplimiento de los plazos establecidos. Estas prácticas permiten una mejor previsión y control de los tiempos del proyecto, optimizando la planificación y ejecución del desarrollo de software.

## **2.2.7. Gestión de Configuración de Software [12]**

La gestión de configuración se centra en el control y rastreo de cambios en el código y otros artefactos del proyecto. En el taller, se instruye a los estudiantes en las mejores prácticas para esta gestión. Se detallan aspectos técnicos relacionados con la creación y manipulación de ramas en sistemas de control de versiones, como Git, y se introducen los principales flujos de trabajo adoptados en la industria, incluyendo el flujo de trabajo centralizado, el modelo de ramificación por función (Feature Branch Workflow) y el modelo de ramificación Git-Flow, entre otros. Estas prácticas son fundamentales para mantener la integridad y la organización del desarrollo del software.

## **2.2.8. Métricas Orientadas al Proceso, al Producto y las Personas**

Las métricas en el desarrollo de software son fundamentales para evaluar el progreso del proyecto, medir la calidad del software y analizar el rendimiento del equipo. En el taller, se exploran diversas métricas orientadas al proceso, al producto y a las personas, y se enseña cómo aplicarlas de manera efectiva. El objetivo es mejorar tanto el proceso de desarrollo como la calidad del producto final, proporcionando una visión clara y cuantificable del desempeño en todas las áreas involucradas.

## **3. Metodología**

El Taller AYDS-IS es una parte crucial del proceso evaluativo de las asignaturas relacionadas, complementando parciales, exposiciones y problemas integradores. La planificación de los contenidos de estas asignaturas se realiza estratégicamente, de manera que los temas se aborden antes y durante el taller. Esto facilita a los estudiantes la conexión entre la teoría y la práctica.

El taller se organiza en reuniones semanales de tres horas. Durante estas sesiones, el docente coordinador presenta el proyecto y detalla la metodología, herramientas y actividades a seguir. Los estudiantes trabajan en sus proyectos, recibiendo orientación del cuerpo docente y adoptando las herramientas y técnicas propuestas. Este enfoque fomenta la aplicación práctica inmediata de lo aprendido y promueve una interacción activa.

Además, se ofrecen horarios de consultas virtuales y presenciales para resolver dudas y proporcionar asesoramiento adicional. Cada equipo debe dedicar aproximadamente 6 horas semanales fuera de las sesiones presenciales para investigar, reunirse y tomar decisiones. Durante el taller, los equipos analizan el problema, desarrollan modelos y presentan sus propuestas de solución en las reuniones. Junto con el cuerpo docente, definen el documento de Especificación de Requisitos de Software (ERS) y el Producto Mínimo Viable (MVP), basándose en los plazos y la complejidad del proyecto. Posteriormente, se estiman y planifican los *sprint*, y cada equipo inicia el primer sprint con diseño, implementación y pruebas de las funcionalidades planificadas. Al finalizar cada sprint, se realiza una reunión de retrospectiva para evaluar el progreso, identificar dificultades y ajustar estrategias para el próximo sprint.

En el primer cuatrimestre de 2024, y en reconocimiento a los excelentes resultados obtenidos en la edición 2023, se volvió a proponer el desarrollo de un software de preguntas y respuestas (trivia). Este software permite a los usuarios autoevaluarse y profundizar en conocimientos sobre un tema específico, brindando a los estudiantes la libertad de seleccionar el dominio de su proyecto. Esta iniciativa promueve la creatividad y la innovación, ya que permite a los estudiantes trabajar en áreas de su interés y aplicar soluciones personalizadas en función de sus habilidades y preferencias.

Al finalizar cada cuatrimestre, los equipos presentaron su proyecto, mostrando el producto final y compartiendo sus experiencias, dificultades y conocimientos adquiridos. El taller permite a los estudiantes integrar y aplicar de manera práctica los conocimientos técnicos adquiridos, brindándoles una formación sólida y preparándolos para los desafíos del desarrollo de software en el mundo laboral..

## **4. Técnicas, herramientas y tecnologías utilizadas**

El desarrollo del taller se distingue por el uso de un stack de herramientas altamente dinámico, que se actualiza y adapta continuamente a las últimas tendencias en desarrollo de software. Esta flexibilidad se refleja en la actualización o cambio de diferentes sistemas de Mapeo Objeto-Relacional (ORM), gestores de bases de datos (DBMS), frameworks y lenguajes de programación empleados. Además de las herramientas específicas para la codificación del software, se emplean diversas herramientas para la creación de modelos y la gestión y planificación del proyecto en general.

A continuación, se enumeran las herramientas y tecnologías incorporadas en las ediciones recientes del taller:

### **4.1. Lenguajes de Programación**

Los estudiantes han trabajado con una gama diversa de lenguajes de programación, cada uno con su función específica en el desarrollo de software. Ruby, conocido por su flexibilidad y simplicidad, se utiliza principalmente en el desarrollo web y en otras aplicaciones de propósito general. JavaScript es una herramienta clave en el desarrollo web, empleado para implementar la lógica del cliente y manejar la interactividad en el navegador. HTML se utiliza para estructurar y organizar el contenido en las páginas web, mientras que CSS se emplea para definir el estilo y la presentación visual de los elementos en las interfaces de usuario web.

### **4.2. Frameworks Web**

El taller se centra en el desarrollo de aplicaciones web, empleando frameworks específicos para facilitar este proceso. En las ediciones recientes, se han utilizado los frameworks de desarrollo web Spark y Sinatra. Spark es un microframework para Java, inspirado en Sinatra, que se destaca por su simplicidad y eficiencia. Al igual que Sinatra, Spark permite a los estudiantes construir aplicaciones web robustas de manera rápida y efectiva. Sinatra, por su parte, es un microframework

para Ruby, conocido por su facilidad de uso y su capacidad para desarrollar aplicaciones web de forma ágil. Es particularmente adecuado para proyectos pequeños y prototipos. La selección del framework web impacta significativamente en la arquitectura de la aplicación y en las herramientas disponibles para su desarrollo.

### 4.3. Object-Relational Mapping (ORM)

En el desarrollo del taller, se emplean técnicas de Object-Relational Mapping (ORM) para gestionar la interacción entre el modelo de datos y la base de datos. Los ORM facilitan la conversión de datos entre el modelo de objetos utilizado en el código y la base de datos relacional subyacente.

En las ediciones recientes del taller, se han utilizado dos ORM: SEQUEL y ActiveRecord. SEQUEL [16] es un ORM para Ruby conocido por su flexibilidad y simplicidad. Permite a los estudiantes realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) y manejar consultas complejas de manera eficiente, utilizando objetos y métodos en lugar de escribir consultas SQL directamente. Esto simplifica la interacción con la base de datos y mejora la mantenibilidad del código.

Por otro lado, ActiveRecord [2][3]es un ORM integrado en el framework Ruby on Rails. Proporciona una interfaz orientada a objetos para interactuar con la base de datos, facilitando la creación de consultas complejas mediante un enfoque basado en objetos. ActiveRecord permite a los estudiantes gestionar la persistencia de datos de manera intuitiva y directa, alineándose con la filosofía de Rails de convención sobre configuración.

### 4.4. Gestión de proyectos

Una gestión efectiva del proyecto de desarrollo es crucial para alcanzar los objetivos establecidos durante el taller. Para facilitar esta gestión, se emplean herramientas especializadas. Entre las aplicaciones utilizadas, se destacan dos gestores de proyectos adaptados a metodologías ágiles: Pivotal Tracker y Trello.

- Pivotal Tracker permite a los equipos planificar, priorizar y rastrear el progreso de sus proyectos de manera eficiente. Los estudiantes pueden crear historias de usuario para representar las funcionalidades del sistema, asignar puntos de historia para estimar la complejidad de cada tarea y planificar iteraciones. Esta herramienta ofrece una visión clara y en tiempo real del avance del proyecto, lo que facilita la toma de decisiones y la identificación temprana de posibles problemas.

- Trello es una herramienta ampliamente utilizada para la gestión colaborativa de proyectos. Su interfaz visual e intuitiva permite a los estudiantes crear tableros que reflejan las distintas fases del desarrollo, listas que organizan las tareas y tarjetas que representan las funcionalidades a implementar. Trello facilita la asignación de tareas a miembros del equipo, el establecimiento de fechas límite y el seguimiento detallado del progreso, promoviendo una colaboración efectiva y un registro claro de las actividades.

Ambas herramientas aportan un enfoque estructurado y flexible a la gestión de proyectos, permitiendo a los estudiantes organizar y controlar eficazmente sus proyectos de desarrollo.

### 4.5. Herramientas para la gestión de configuración de software

La gestión eficiente de la configuración del software es fundamental para el desarrollo colaborativo de proyectos. En el taller, utilizamos Git como sistema de control de versiones. Git establece un entorno distribuido y ampliamente adoptado 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 forma ordenada. Además, pueden deshacer cambios en caso de errores, proporcionando una seguridad adicional en el proceso de desarrollo.

Los proyectos de los estudiantes son gestionados mediante Pivotal Tracker (u otro gestor de proyectos) y alojados en GitHub. Esta última es una plataforma de alojamiento y colaboración que utiliza Git como sistema de control de versiones. Los equipos en el taller se benefician de GitHub, especialmente en la gestión de workflows para el manejo de ramas. Los sistemas de gestión de proyectos se vinculan a los repositorios en GitHub para mantener una documentación en tiempo real del avance de cada tarea dentro del proyecto.

## 5. La mirada de los estudiantes sobre la experiencia

Este año, como parte del proceso de indagación para el seguimiento y la mejora de la enseñanza en el marco de nuestras materias, decidimos implementar una encuesta a los estudiantes que cursaron Análisis y Diseño de Sistema al finalizar el primer cuatrimestre.

El objetivo de la misma es conocer sus opiniones y valoraciones respecto de: lo que ellos consideran que les ha permitido aprender el cursado de la materia en general y el desarrollo del proyecto de software en particular; de las posibilidades de integración de contenidos que se generan a partir de la propuesta de la materia; conocer cómo les resultó la experiencia del cursado; y recuperar también sugerencias o ideas que ellos consideran que podrían ayudarnos a mejorar el desarrollo de la materia.

La encuesta se formuló con 5 preguntas, cuatro de respuesta abierta y una de selección múltiple, se planteó para ser respondida de forma anónima a través de un formulario web, que se compartió en el grupo de Slack de la materia y también se envió por email a los 37 estudiantes que cursaron la materia. Respondieron a la encuesta 21 estudiantes.

Como primera pregunta de la encuesta les pedimos que señalaran un tema, una habilidad y una actitud que el cursado de la materia Análisis y Diseño de Sistemas les ha permitido aprender o desarrollar, y que ellos consideraban significativos para su futuro desempeño profesional. Cabe mencionar en este punto que no todos respondieron completamente a la pregunta identificando un tema, una habilidad y una actitud, y que en varias respuestas estas dos últimas se plantean de manera conjunta.

En relación a la consulta sobre un tema que ellos rescatan por considerarlo significativo para su futuro desempeño profesional la mayoría se refirió a la comprensión de lo que implica el proceso de desarrollo software de las etapas que lo conforman (7 estudiantes) y a metodología SCRUM que utilizaron en el taller para la realización de su proyecto (5 estudiantes). Entendemos que es natural que esto suceda ya que es la primera materia de la carrera donde se aborda en detalle el proceso de desarrollo de software y las diferentes metodologías posibles, aplicando esos conocimientos en un proyecto concreto. En menor medida algunos estudiantes rescataron como temas significativos la captura de requerimientos puntualmente (3 estudiantes), el uso de la notación gráfica UML (2 estudiantes), y patrones de diseño (2 estudiantes).

Respecto a las habilidades que consideran significativas para su desempeño profesional y que ellos entienden que la materia les ha permitido desarrollar, la mayoría de las respuestas se agrupan en torno a las habilidades necesarias para: realizar presentaciones y exposiciones orales (5 estudiantes), para el trabajo en equipo (5 respuestas), realizar modelado gráfico mediante diagramas UML que ayudan a entender mejor los problemas, a crear soluciones, y que facilitan la comunicación de los equipos (4 estudiantes). En un caso mencionaron a la colaboración, y en otro a la capacidad de análisis, como habilidades significativas.

En cuanto a las actitudes que los estudiantes consideran que han podido desarrollar durante el cursado de la asignatura, las respuestas son más dispersas y se mencionan como significativas: el poder perder el miedo y controlar sus nervios al hacer una presentación oral, el poder dividir y delegar tareas, el ser críticos respecto del proceso de desarrollo de software, el ser proactivos, el ser constantes frente a las dificultades.

Citamos a continuación algunas respuestas de los que dan cuenta de lo planteado anteriormente.

*“Metodología SCRUM.” E1*

*“El tema más importante me parece el hecho de cómo construir un software, en el sentido que no es tan directo de solo programar el código, sino de todas las cosas que hay por detrás y todo el proceso. Una habilidad y también una actitud es el hecho de las presentaciones orales que hicimos en la materia, ya que ayudaron a perder un poco ese miedo/nerviosismo que tiene uno cuando no está tan acostumbrado a esas situaciones.” E2*

*“Cursar la materia Análisis y Diseño de Sistemas me permitió tener la habilidad de poder plantear un problema de una forma visual a través de diversos diagramas, lo cual facilita el proceso de la creación de código y considero que es bastante importante para mi futuro desempeño profesional por dos cosas: para poder entenderme mejor con los demás desarrolladores con los que pueda trabajar y para facilitarme mis tareas de implementación.” E9*

*“El cursado de AyDS me permitió aprender sobre el ciclo de vida del desarrollo de software, desde la recolección de requisitos hasta la implementación y mantenimiento. Una habilidad que aprendí fue modelar sistemas utilizando diagramas UML. La materia me ha inculcado una actitud proactiva y analítica, permitiéndole abordar problemas complejos de manera estructurada y eficiente.” E12*

*“Un tema que considero significativo es principalmente conocer las diferentes etapas de un desarrollo de software, por el hecho de que a futuro tendré que tener en cuenta en qué punto de progresión se encuentra y cómo afrontar un obstáculo en cada etapa. Como habilidad el trabajo en equipo, ya sea como coordinarse con los diferentes miembros para dividirse las tareas o conocer en qué etapa del desarrollo de un software se encuentran y tomar decisiones para mejorar la producción de manera conjunta del mismo. Por último, una actitud que me llevo como aprendizaje es la presentación del producto final, ya que es una situación que probablemente vuelva a presentarse en el futuro y haber tenido aunque sea una pequeña experiencia de cómo llevar a cabo y afrontar una presentación me ayuda reconocer mejoras que puedo hacer para las próximas.” E18*

También les preguntamos a los estudiantes si consideran que el desarrollo del proyecto a lo largo del cuatrimestre les permitió ir relacionando e integrando los contenidos que se iban abordaban en las clases teórica, prácticas y del taller de la materia, y les pedimos que justificaran esa respuesta. Casi la totalidad de los estudiantes que respondieron lo hicieron afirmativamente a esta consulta (20 estudiantes) ofreciendo justificaciones variadas más o menos precisas, mientras que sólo 1 estudiante expresó que no logró establecer relaciones ni integrar esas instancias, justificando su respuesta de la siguiente manera:

*“No logré relacionar los contenidos adecuadamente entre el proyecto y la teoría y práctica. No había necesidad de volver a los teóricos o prácticos en la mayor parte del proceso del proyecto, fue mucho más intentar que las cosas funcionaran en los lenguajes de programación que relacionar conceptos” E4*

Quienes respondieron afirmativamente ofrecieron justificaciones como las siguientes:

*“Si, totalmente. El proyecto fue de gran ayuda para integrar conocimientos, así también como para resolver un problema que se acercaba más a la realidad.” E3*

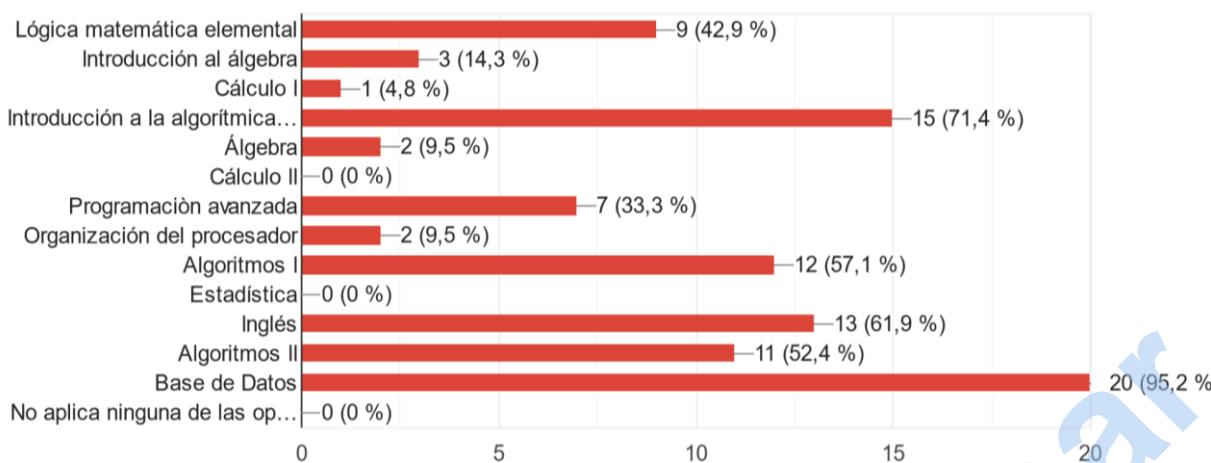
*“Sí, la mayoría de los temas abordados se pudieron relacionar a medida que avanzamos en las distintas etapas del proyecto, desde el análisis y diseño con los diagramas, el srs, etc, la implementación del sitio web abordando todos los contenidos del taller hasta el testing. Y por supuesto scrum que se aplicó a lo largo de todo el proceso de desarrollo. Entonces de alguna u otra forma casi todos los contenidos se relacionaron en el proyecto. En el caso de nuestro equipo lo único que no llegamos a aplicar fue algún patrón de diseño al diagrama de clase de la app.” E6*

*“Si totalmente, es mucho más llevadero realizar un proyecto que te impulsa a comprender la teoría, práctica y demás.” E8*

*“Si, cada una de las diferentes clases ayudó en diferentes puntos a llevar el desarrollo del proyecto. Las clases teóricas me dieron la comprensión de los contenidos y brindó el conocimiento necesario para entender el ambiente del desarrollo de un software, las diferentes etapas del mismo y herramientas para trabajar. La puesta en práctica de los conceptos fue llevada a cabo en las clases prácticas donde a partir de diferentes situaciones que se nos proponían, debíamos poner a prueba los conocimientos obtenidos y las herramientas que teníamos que usar (Búsqueda de requerimientos de un software, diagramas, patrones de diseño, pruebas de software) para llegar a una solución. Por último, el taller nos dio la experiencia de cómo trabajar con un equipo de trabajo en el desarrollo de un software y los distintos papeles que podemos tomar los miembros del mismo y además el soporte de cómo llevar a cabo cada etapa mientras avanzábamos en el proyecto, ya sea ofreciendo herramientas para su desarrollo o ayudas para pasar un obstáculo que se nos presentara.” E18*

En cuanto a las posibilidades que la estrategia del desarrollo de un proyecto de software le ofrece a los estudiantes para integrar conocimientos, relacionarlos, comprenderlos mejor, quisimos conocer e ir un poco más de nuestras asignaturas, y les pedimos que señalen otras materias de las cuales habían podido relacionar e integrar conocimientos. Casi la totalidad (20 estudiantes) semana una fuerte integración con Base de datos, materia que cursan en simultáneo a la nuestra, mientras que en menor medida (11 estudiante) ven relación con Algoritmos II que también es del mismo año. Entre las materias de años anteriores de las cuales han podido integrar conocimientos las más señaladas son Lógica matemática elemental (9 estudiantes) e Introducción a la algorítmica y programación (15 estudiantes) ambas de primer año, y en el caso de segundo año mencionan Inglés (13 estudiantes) y Algoritmos i (12 estudiantes).

En el siguiente gráfico se muestra el detalle de las respuestas.



También en la encuesta les preguntamos a los estudiantes cómo describirían su experiencia de cursado del taller, para conocer qué vivencias estaban teniendo, que apreciaciones surgían al respecto. Quienes respondieron la encuesta en general coinciden en hacer valoraciones favorables respecto de la experiencia de cursado expresando que fue muy buena (4 estudiantes), con mucho aprendizaje (7 estudiantes), sumamente enriquecedora (3 estudiantes), muy entretenida (3 estudiantes), excelente (2 estudiantes), buena (2 estudiantes), positiva (2 estudiantes), gratificante (1 estudiante), práctica (1 estudiante), interesante (1 estudiante).

No obstante estas valoraciones positivas, algunos estudiantes también al momento de responder referencian algunas dificultades, tales como. que el cursado requiere estar al día, que es exigente, con entregas en tiempos breves, estresante, pesada en cantidad de conceptos nuevos y tareas, que a veces se les dificulta estar al día con otras materias.

Compartimos a continuación algunas de las respuestas que dieron los estudiantes y que dan cuenta de lo antes mencionado.

*“Gratificante” E3*

*“Los tiempos estaban muy justos, pero en general fue buena” E4*

*“Fue muy entretenida, con mucho aprendizaje, costó un poco mantenerse al día con el cursado de otras materias por qué es una materia exigente pero el desarrollo del proyecto me encantó porque nos dió la primer forma de cómo trabajar en un proyecto un poco más grande que contiene muchos conocimientos nuevos y también previos” E5*

*“Fue buena, ya que fue lo más cercano a un proyecto de la vida real que tuve, muy útil para aprender a trabajar en equipo y a administrar tiempos.” E10*

*“Mi experiencia de cursado del taller fue muy enriquecedora, ya que me permitió aplicar los conocimientos teóricos en situaciones prácticas, trabajar en equipo y recibir retroalimentación constante de los profesores. Fue un espacio dinámico y colaborativo que potenció mi aprendizaje.” E14*

*“Creo que el taller fue una experiencia interesante pero a su vez estresante, creo que sí hubieran más clases destinadas a desarrollar el contenido utilizado habría sido más ameno el transcurso del proyecto” E15*

*“Fue una experiencia distinta y positiva a comparación de todas las que hemos tenido en el cursado de la carrera, ya que nos dio un acercamiento de cómo trabajar y desarrollar un producto de software con un equipo de trabajo como profesionales, sus diferentes etapas y cómo organizarse para mejorar el desarrollo del mismo.” E18*

Finalmente en la encuesta les preguntamos a los estudiantes si les gustaría dejarnos alguna sugerencia o comentario para mejorar la propuesta del taller y/o de la materia en general, habilitando así un espacio abierto para que pudieran plantear ideas que nos ayuden seguir mejorando la enseñanza en el contexto de las materias que tenemos a cargo.

En general la mayoría de los estudiantes respondió pero manifestando que no se le ocurrirán sugerencias y reiteraban valoraciones positivas en el mismo sentido de las que ya habían hecho anteriormente en otras preguntas. Solo unos pocos estudiantes plantearon algunas sugerencias en relación a: poder hacer más seguimiento a algunos grupos de trabajo que presentan dificultades, a poder indagar sobre conocimientos previos antes de desarrollar los temas, a acotar la cantidad de horas de las clases teóricas, a incrementar más horas en la semana para la clase del taller y enseñar más HTML y CSS.

Citamos aquí algunas de esas expresiones que dan cuenta de las sugerencias realizadas:

*“Mayor seguimiento por parte de los profesores sobre los grupos para evaluar si se están aplicando correctamente los pasos a seguir en el desarrollo de software utilizando metodologías ágiles.” E1*

*“Intentar de exigirle un poco más a algunos grupos, por qué a veces algunos avanzaban mucho en el proyecto y otros quedaban bloqueados con algo y como que no se ponían a intentar ni preguntaban a otros grupos de cómo se hace o cómo podrían hacerlo.” E5*

*“Creo que en general la propuesta de la materia está bastante buena, con respecto a los temas tratados tanto en el práctico, teórico y taller no cambiaría nada, pero algo ajeno a los temas que no me gustó tanto fue el horario del teórico, que sean entre 2 y 3 horas teóricas a las 13:00 es un poco pesado, lo que me llevó más de una vez a no prestarle tanta atención.” E9*

*“Preguntar en general el nivel de aprendizaje que tiene el curso sobre ciertos temas para ver si se tiene que explicar algo básico o empezar directamente con el tema más avanzado” E11*

*“Cómo dije anteriormente, más clases para el taller en la semana y sobre todo enseñar más HTML y CSS que son necesarios para el desarrollo del proyecto.” E15*

Resulta muy valioso para nosotros como equipo poder recuperar la mirada de los estudiantes sobre los que estamos haciendo en la materia, a través de espacios de diálogo en las clases o de la encuesta que implementamos, para a partir de esas valoraciones y sugerencias, corroborar, mantener y mejorar lo que resulta significativo para sus aprendizajes, y tratar minimizar las dificultades y problemáticas que están detectando.

## 6. Conclusión

A lo largo de las diversas ediciones del Taller AYDS-IS, se han observado resultados muy positivos y alentadores. Este taller, que forma parte integral del enfoque didáctico para la integración transversal de Análisis y Diseño de Sistemas e Ingeniería de Software en la carrera de Analista, ha demostrado ser una herramienta eficaz para mantener una dinámica de trabajo altamente eficiente entre los equipos de estudiantes, a pesar de su diversidad.

Una de las principales ventajas del taller es la experiencia adquirida con metodologías ágiles. Los equipos tienen la oportunidad de trabajar con SCRUM, una metodología que les permite comprender los beneficios de una gestión de proyectos adaptable, colaborativa y con una comunicación efectiva. Estos elementos son fundamentales para el éxito en el desarrollo de software y se alinean con los objetivos del enfoque didáctico transversal.

El trabajo en equipos colaborativos fomenta la toma de decisiones conjunta y la distribución eficiente de tareas y responsabilidades. Esta dinámica fortalece las habilidades de comunicación y preparación profesional de los estudiantes, aspectos clave en la formación de analistas de sistemas y desarrolladores de software.

Además, cada equipo tiene la oportunidad de autogestionarse, organizando su trabajo y definiendo sus propias estrategias para cumplir con los objetivos del proyecto. Esto les permite desarrollar habilidades y actitudes para el aprendizaje autónomo, esenciales en el entorno profesional actual, y está en línea con el enfoque didáctico transversal de la carrera.

El uso de nuevas tecnologías y metodologías, aunque implica una curva de aprendizaje inicial de media a alta, está cuidadosamente planificado en el taller. Este aspecto asegura que los estudiantes inviertan el esfuerzo necesario para adaptarse y comprender completamente su funcionamiento, fortaleciendo así su formación práctica y teórica.

Es importante destacar que la dinámica y organización del taller, junto con el desarrollo de toda la asignatura y los cambios en la modalidad de evaluación, centrada en el proceso de aprendizaje, han fortalecido y mejorado significativamente las tasas de graduación de la carrera de Analista en Computación. Los estudiantes están mejor preparados para abordar el Proyecto Final de la carrera con mayor eficiencia, lo que genera un impacto positivo en la finalización de la Licenciatura. Además, se observa un impacto favorable en 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

- [1] K. Beck, M. Beedle, A. V. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, and D. Thomas, “Agile Manifesto for Software Development.” [Online]. Available: <https://agilemanifesto.org/>. [Accessed: Jul. 26, 2024].
- [2] Active Record. [Online]. Available: <https://github.com/rails/rails/tree/main/activerecord>. [Accessed: Jul. 26, 2024].
- [3] Sinatra Active Record. [Online]. Available: <https://github.com/sinatra-activerecord/sinatra-activerecord>. [Accessed: Jul. 26, 2024].
- [4] Docker. [Online]. Available: <https://www.docker.com>. [Accessed: Jul. 26, 2024].
- [5] Hypertext Transfer Protocol -- HTTP/1.1. “RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1.” [Online]. Available: <https://www.ietf.org/rfc/rfc2616.txt>. [Accessed: Jul. 26, 2024].
- [6] Ruby Version Manager - RVM. [Online]. Available: <https://rvm.io/rvm/install>. [Accessed: Jul. 26, 2024].
- [7] Bundler (RubyGem). [Online]. Available: <https://bundler.io/gemfile.html>. [Accessed: Jul. 26, 2024].
- [8] K. Beck, *Extreme Programming Explained: Embrace Change*, 2nd ed. Addison-Wesley, 2004.
- [9] K. S. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley Professional, 2012.
- [10] M. Cohn, *Agile Estimating and Planning*. Prentice Hall, 2005.
- [11] J. V. Sutherland, *SCRUM: The Art of Doing Twice the Work in Half the Time*. Crown Business, 2014.
- [12] Scrum Alliance. [Online]. Available: <https://www.scrumalliance.org/>. [Accessed: Jul. 26, 2024].
- [13] Ruby. [Online]. Available: <https://www.ruby-lang.org/en/>. [Accessed: Jul. 26, 2024].
- [14] Sinatra. [Online]. Available: <http://sinatrarb.com/>. [Accessed: Jul. 26, 2024].
- [15] ERB. [Online]. Available: <http://www.kuwata-lab.com/erubis/users-guide.html>. [Accessed: Jul. 26, 2024].
- [16] Sequel. [Online]. Available: <http://sequel.jeremyevans.net/>. [Accessed: Jul. 26, 2024].
- [17] IEEE, “IEEE 830-1998 - IEEE Recommended Practice for Software Requirements Specifications,” IEEE, 1998. [Online]. Available: <http://standards.ieee.org/findstds/standard/830-1998.html>. [Accessed: Jul. 26, 2024].

# Co-diseño de materiales educativos sobre computación en un Jardín de Infantes de la Provincia de Buenos Aires

Lucila Dughera<sup>1</sup>, Fernando Raúl Alfredo Bordignon<sup>2</sup>

<sup>1</sup>CONICET, CABA, Argentina

<sup>2</sup>Universidad Pedagógica Nacional (UNIPE), Pilar, Argentina  
[ldughera@gmail.com](mailto:ldughera@gmail.com), [fernando.bordignon@unipe.edu.ar](mailto:fernando.bordignon@unipe.edu.ar)

## Resumen

En esta comunicación se presentan las primeras experiencias derivadas del desarrollo del proyecto institucional ¡Robótica para aprender! en un Jardín de Infantes de la provincia de Buenos Aires. El objetivo principal del proyecto reside en incorporar a las prácticas docentes los nuevos contenidos curriculares, en particular los relacionados con el Pensamiento Computacional, establecidos en el Diseño Curricular del año 2023 y en los Núcleos de Aprendizaje Prioritarios de Educación Digital, Programación y Robótica. Si bien en el jardín de infantes desde el año 2021 se viene trabajando con la temática en sala de cinco años, a partir de este año dicho proyecto se extendió en términos institucionales a todas sus salas. Decisión que implica un trabajo en distintas dimensiones, a saber: planificación, diseño y producción de materiales educativos, desarrollo de secuencias didácticas, evaluación y capacitación. Esta comunicación tiene dos propósitos: a) compartir algunas experiencias genuinas de trabajo en relación al desarrollo de saberes vinculados a la computación y b) presentar el proyecto institucional denominado ¡Robótica para aprender! Una experiencia de trabajo pedagógico conjunta entre la comunidad docente del jardín de infantes e investigadores externos, que se basa en el co-diseño de materiales educativos y actividades.

**Palabras clave:** Pensamiento computacional; Nivel Inicial; Co-diseño de materiales educativos.

## 1. Introducción

Vivimos en un mundo tecno-social donde los niños y niñas se desarrollan en un espacio y tiempo caracterizado por el acceso permanente a pantallas digitales. En general, a través de ellas acceden a múltiples aplicaciones, informaciones y/o potenciales relaciones con otras personas. En este escenario, la alfabetización tradicional, centrada en las tres “R” (lectura, escritura y aritmética) ya no alcanza para formar a los ciudadanos actuales [1]. Estas generaciones necesitan construir nuevos saberes que les permitan, por ejemplo, comunicarse adecuadamente a través de tecnologías digitales, colaborar con sus pares, ser creativos, resolver problemas y desarrollar planes, entre las principales.

En este sentido, Vee [2] indica que la programación es una alfabetización sobre la que es posible construir otras actividades y conocimientos, de la misma manera que sucede con la escritura. En su mirada crítica sobre la programación señala que es un fenómeno importante de comunicación y no simplemente una nueva habilidad tecnológica, esto debido a que la programación en sí opera como infraestructura ya que funciona como la base de todas nuestras comunicaciones mediadas. En efecto, una formación que promueva una cultura digital, entendida como saberes y habilidades emergentes del uso cotidiano e intensivo de las tecnologías digitales, y desarrolle habilidades de resolución de problemas en alianza con la computación es necesaria. Así, pensar en estas alfabetizaciones, sobre todo su comienzo desde el jardín de infantes, podría colaborar de manera activa en la formación de ciudadanos situados, ya que complementa y expande sus saberes en relación a comprender e intervenir el mundo en el que viven.

El aprendizaje de temas propios de la computación, en particular el desarrollo del Pensamiento Computacional<sup>1</sup> [3][4], se percibe como un conjunto de conocimientos y habilidades que permiten comprender las tecnologías digitales actuales y la

<sup>1</sup> Entendemos el Pensamiento Computacional como el desarrollo de saberes en relación a la resolución de problemas (abstracción, descomposición, secuenciación, generalización y detección y corrección de errores en planes), el pensamiento

informacionalización de casi cualquier aspecto de la vida e interactuar de una manera más profunda con dichas tecnologías. Entendiendo que la computación es una tecnología diferente, particular en sus efectos, dado que a las personas les permite extender sus capacidades cognitivas. En particular, el período de la infancia es una fuente de oportunidades para el desarrollo cognitivo y socioemocional de los niños. En los primeros cinco años de vida, el tener una atención infantil adecuada, de calidad y una apropiada formación han demostrado tener efectos positivos a largo plazo en la vida de las personas. Las habilidades de función ejecutiva [5], procesos cognitivos necesarios para regular los pensamientos, las emociones y el comportamiento orientado a objetivos, pueden experimentar una mejora significativa en los sujetos, en particular, sobre el rendimiento académico. Así, las funciones ejecutivas básicas, como el control inhibitorio, la memoria de trabajo y la flexibilidad cognitiva<sup>2</sup> y funciones ejecutivas superiores, como la planificación y la resolución de problemas y el razonamiento<sup>3</sup>, pueden desarrollarse adecuadamente con actividades escolares planificadas que giren en torno a la alfabetización en computación [6] [7] [8] [9]

En los últimos años han surgido y se han desplegado, en distintos países (por ejemplo, India, Israel, Brasil, Estonia, Costa Rica, Australia, Vietnam, Reino Unido y Nueva Zelanda), políticas educativas en relación a la enseñanza de los conceptos básicos de las Ciencias de la Computación con foco en la escuela obligatoria. En este sentido, distintos gobiernos han reformulado sus políticas educativas y han incorporado saberes en torno a la computación al currículum escolar [10] [11] [12] [13] [14] [15].

En particular, en la República Argentina, en el año 2018, el Consejo Federal de Educación aprobó un documento denominado Núcleos de Aprendizajes Prioritarios (NAPs) para Educación Digital, Programación y Robótica [16]. Estos proponen una serie de temas y objetivos a incorporarse a los niveles Inicial, Primario y Secundario del país. Los NAPs representan un potencial avance desde las políticas públicas, ya que expanden la idea de alfabetización, pero, a la vez, suponen un conjunto importante de desafíos educativos para su implementación real. Siguiendo tales definiciones, algunas jurisdicciones del país han ido agregando apartados a sus diseños curriculares en los distintos niveles de la educación obligatoria (por ejemplo, Buenos Aires, Corrientes, Córdoba, San Luis y la Ciudad de Buenos Aires ya han publicado recomendaciones curriculares posteriores a los NAPs). Sin embargo, tal como ha quedado demostrado en la implementación de diferentes políticas públicas no alcanza únicamente con tener los NAPs. Sino que además se requieren de un conjunto de aspectos para llevar adelante efectivamente este tipo de decisiones políticas. Por ejemplo, diseños curriculares, documentos de estudio, docentes capacitados y lineamientos y materiales didácticos para la enseñanza y el aprendizaje, equipos de conducción dispuestos a liderar este tipo de proyectos y comunidades educativas comprometidas con el enriquecimiento de sus prácticas. En resumen, estos elementos deben ser investigados, pensados, creados e implementados para construir conocimiento pedagógico en el área ¡Ese es nuestro desafío educativo!

En el año 2022, el Consejo General de Cultura y Educación de la provincia de Buenos Aires aprobó un documento titulado “Diseño curricular. Educación inicial” [17] en el cual se incluye un apartado denominado “Educación Digital”. En una cita a los NAPs mencionados anteriormente se plantea el espíritu del desafío institucional docente para incorporar nuevos saberes, indicando que:

Se trata de asumir y acompañar el desafío de abordar nuevos problemas pedagógicos, propios del mundo contemporáneo, que tienen sus historias y recorridos particulares en las instituciones educativas, pero que actualmente han cobrado relevancia social y educativa (ibid).

En un planteo que trata de superar modelos de alfabetización basados en solamente la formación de usuarios y el mero uso instrumental de las tecnologías digitales, se plantea como objetivo de formación a “niñas y niños autónomas y autónomos que puedan pensar y actuar críticamente, plantear y resolver problemas, colaborar, expresarse, comunicarse, participar. Estudiantes que puedan desempeñar un rol activo: que entiendan las lógicas de la tecnología, puedan explorar cada vez que aparecen nuevos desarrollos, elegir cómo usarla (o no) e incluso crearla” (ibid)

---

lógico y el pensamiento divergente. La investigadora sobre didáctica de la computación Umaschi Bers indica que el “*pensamiento computacional es pensar como un programador pero no necesariamente es sentarse a programar. (...) El objetivo es aprender a pensar como un programador para resolver problemas de la vida en general.*” [1]

<sup>2</sup> El control inhibitorio se refiere a la capacidad de controlar las respuestas automáticas e inhibir los estímulos de distracción en función de mejorar la atención. La memoria de trabajo es la capacidad para registrar y mantener información con el fin de manipularla y la flexibilidad cognitiva se refiere a la capacidad para cambiar la atención entre tareas y ajustar nuestras respuestas ante los cambios.

<sup>3</sup> Las tareas de planificación implican el desarrollo de habilidades que permiten pensar planes de acción para llegar a determinados objetivos. La resolución de problemas tiene que ver con el pensamiento creativo, a partir de abordar un problema y tratar de diseñar e implementar una solución. El razonamiento o pensamiento lógico es entendido como el proceso cognitivo donde se extraen conclusiones a partir de premisas, principios o pruebas, utilizando información previa para inferir una nueva conclusión.

En los últimos años, jardines de infantes de la provincia de Buenos Aires han sido dotados de una serie de recursos educativos y capacitaciones centradas en lo que se conoce como robótica educativa (Plan Aprender Conectados). En particular, orientando las prácticas docentes hacia la operación de artefactos digitales tangibles interactivos (por ejemplo, robots tipo Bee Bot<sup>4</sup>). Así, los jardines de infantes fueron dotados de un Aula Digital Móvil que entre otros artefactos incluía conexión a Internet, tabletas, un proyector, una netbook y un equipo de sonido. En particular, el Jardín de Infantes 913 (Belén de Escobar, Provincia de Buenos Aires) desde mediados del año 2020 comenzó a integrar a sus prácticas, en sala de cinco años, los robots provistos. Comenzando así un importante camino de exploración e integración de nuevos saberes y prácticas docentes.

Este escrito tiene dos objetivos, por un lado, compartir algunas experiencias genuinas de trabajo de un jardín de infantes público de la Provincia de Buenos Aires en relación al desarrollo de saberes de la computación y, por otro lado, presentar el proyecto institucional denominado ¡Robótica para aprender! Una experiencia de trabajo pedagógico conjunta entre la comunidad docente del jardín de infantes e investigadores externos, que se sustenta en el co-diseño de materiales educativos y actividades. Si bien es un proyecto que se inició a principios del año 2024 ya han surgido una serie de orientaciones y prototipos de materiales didácticos que entendemos pueden ser compartidos.

## 2. Descripción de la Experiencia

### 2.1. Propuesta Didáctica

En este apartado se presentan algunos lineamientos en relación al proyecto de asesoramiento y co-diseño de materiales educativos vinculados al Pensamiento Computacional. La experiencia se desarrolla en un Jardín de Infantes público y corresponde al proyecto institucional ¡Robótica para aprender!, que se implementa desde el año 2024. Sin embargo, cabe mencionar, tal como se detalla en el apartado Contexto, que los primeros acercamientos al mundo de la robótica se gestan a fines del 2020 de la mano de una maestra de sala de 5.

El proyecto ¡Robótica para aprender! es el resultado de una diversidad de actividades que se fueron desplegando en el Jardín con ocasión de la vocación manifiesta de las docentes, lxs investigadores – asesores y el equipo de conducción de hacer extensible experiencia con el pensamiento computacional de la sala de 5 al resto de las salas (4 y 3 años). Así, en un primer acercamiento a la institución, se realizaron actividades que permitieron comenzar a construir una conversación en común, específicamente entrevistas a las maestras pioneras, la directora y observaciones de clase. Luego, a partir del análisis de dichos materiales, se llevaron a cabo una serie de talleres temáticos con toda la comunidad docente y el equipo de conducción, a fin de presentar conceptos y actividades básicas sobre Computación en general y Pensamiento Computacional en particular. Desde el aspecto didáctico, se trabajaron diferentes conceptualizaciones y ejemplificaciones en base a experiencias en otras instituciones y orientaciones de referentes en la temática. Aquí, se hizo especial hincapié en que las docentes desde sus saberes y experiencias prueben, manipulen – en los casos que fuera necesario – evalúen, critiquen constructivamente los “primeros borradores” de las actividades y materiales propuestos por lxs investigadores-asesores. A partir de esas primeras devoluciones, se acordó una agenda de trabajo que tuvo entre sus principales propósitos que las docentes incorporen a sus aulas algunas de las actividades co-diseñadas en los talleres. Más aún, la retroalimentación constante de las experiencias y materiales que se propusieron es constitutiva de la relación de investigación y asesoría que realizamos. Así, quienes suscriben estas líneas desarrollamos los primeros borradores a partir de lo ya experimentado en el Jardín, curamos dichos materiales a la luz de una conversación fluida, crítica y enriquecida por las voces de la experiencia, las docentes y el equipo de conducción para que continúen con su desarrollo y puesta a prueba.

A los efectos de acordar criterios para enfocar el trabajo inicial se dio prioridad al desarrollo de propuestas de tipo desenchufado (sin uso de computadoras u otros artefactos digitales) donde los estudiantes trabajen con su cuerpo (modo kinético), en papel o manipulando objetos tangibles.

Las capacidades del pensamiento computacional, tales como la planificación, la resolución de problemas y el razonamiento pueden desarrollarse en los estudiantes a partir de una amplia variedad de materiales y actividades. Por ejemplo, la siguiente es una lista (no exhaustiva) de tareas que entendemos como pertinentes para el desarrollo de tales saberes:

- Diseñar planes para resolver problemas o tareas
- Ejecutar planes
- Detectar errores en planes e intentar corregirlos
- Trazar rutas a determinados lugares
- Hacer búsquedas de todo tipo (construyendo estrategias)
- Definir un orden sobre una determinada situación
- Clasificar objetos por distintos criterios

<sup>4</sup> Para más información, [https://www.tibot.es/blog/productos/que-es-bee-bot?srsltid=AfmBOooYn\\_JzqEaWiD12OgEiUeKGDWGJ6-ANSebwsGSLVRRmgd54TQ2d](https://www.tibot.es/blog/productos/que-es-bee-bot?srsltid=AfmBOooYn_JzqEaWiD12OgEiUeKGDWGJ6-ANSebwsGSLVRRmgd54TQ2d)

- Contar historias
- Jugar juegos para desarrollar la lateralidad y el concepto de orden
- Jugar juegos que impliquen el pensamiento lógico y el desarrollo de estrategias
- Diseñar y crear cosas (tangibles y no tangibles)

Dado que el proyecto institucional involucra las salas de 3, 4 y 5 años se priorizaron diferentes tipos de actividades según la franja etárea de cada sala. La siguiente tabla muestra los criterios de trabajo:

Tabla 1. Acuerdo de criterios de trabajo según edad de los estudiantes

Sala	Actividad
3 años y 4 años	Kinética Desarrollo de la lateralidad Baile grupal con coreografías Ejecución y evaluación de planes (por ejemplo con actividades tipo “niño robot”) Reconocimiento de distintos tipos de artefactos digitales y conceptos asociados Desarrollo del razonamiento
5 años	Kinética Desarrollo, ejecución y evaluación de planes con actividades desenchufadas y con artefactos digitales (por ejemplo robotinas) Reconocimiento y manipulación de conceptos computacionales y componentes digitales Juegos de mesa para el desarrollo del pensamiento lógico, la abstracción y la planificación.

La propuesta de trabajo conjunto para el co-diseño de materiales educativos sobre computación tuvo como referentes pedagógicos a Piaget y su “Teoría constructivista del aprendizaje” y la desarrollada por su discípulo Papert, denominada “Construcionismo” [18]. En particular la robótica educativa toma elementos del constructivismo piagetiano pero le incorpora la idea que desarrollo de nuevos aprendizajes es más eficiente cuando los estudiantes se comprometen en el diseño y construcción, por sus propios medios, de un objeto público (tangible o no) [18]. Así, nuestras propuestas y prototipos de trabajo, reconocen que los estudiantes aprenden a través de la experiencia y así construyen estructuras mentales que permiten organizar la información en nuevos conocimientos. En este sentido, entendemos que el aprendizaje es una construcción propia de cada individuo, que es realizada desde la niñez, gracias a la ayuda recibida de otros pares y de su interés, su disponibilidad, conocimientos previos y experiencias [19] [20][21]. Este enfoque reconoce dos tipos de “construcciones”, por un lado, la que implica que los niños hagan cosas en el mundo real y la que también, a la par, hacen en sus mentes. Bajo esta forma se configura un ciclo de auto-refuerzo [18] que enriquece los aprendizajes, entre la acción y la reflexión.

En nuestro proyecto, el trabajo con computación y robótica se entiende a partir del diseño e interacción con “objetos para pensar”. Es decir, en artefactos creados por terceros o el mismo estudiante, que son utilizados para pensar sobre otras cosas y así ampliar su horizonte cognitivo. Al interactuar con los objetos para pensar los niños ponen a prueba y expanden sus conocimientos a partir de diversas tareas (por ejemplo: diseño, creación, prototipado, resolución de problemas entre otras) en una actividad fundamental como es la exploración lúdica [22]. Cuando estos objetos se comparten con otras personas el aprendizaje se enriquece a partir del diálogo y los nuevos elementos de reflexión. En general, la interacción con los recursos educativos se produce de manera física cuando los aprendices manipulan diversas herramientas que les permiten crearlos, modificarlos o simplemente usarlos. Luego existe una segunda interacción de carácter lógico cuando los sujetos estudiantes crean e implementan soluciones posibles a retos dados. Tales diseños, en general, se manifiestan como una serie de pasos que le indican a un autómata o compañero acciones que debe realizar. Como observamos hay una correlación efectiva entre las acciones y el pensamiento lógico. Entendemos que los materiales y actividades que se están co-diseñando pueden llegar a formar espacios de aprendizaje enriquecidos por la computación bajo el enfoque de aprendizaje construccionista. Esto es debido a que, por un lado, permiten trasladar las experiencias con objetos a ideas que transforman las percepciones y conocimientos previos de los estudiantes y, por otro lado, aportan elementos que hacen que los niños tomen un rol más dinámico, dándoles oportunidades de ser constructores de sus propios conocimientos. Todo esto en un ambiente lúdico, creativo, donde está presente la prueba y el error y se apela con frecuencia al uso de la imaginación.

## 2.2. Contexto

Este trabajo es realizado en el Jardín de Infantes 913 “Juana Azurduy de Padilla”. La institución está situada en la localidad de Belén de Escobar y depende de la Dirección General de Escuelas de la provincia de Buenos Aires, Argentina. Es una institución educativa pública gratuita que ofrece formación en el nivel inicial. En relación a la matrícula, en el año 2024, hay 200 alumnos en tres secciones por cada turno, 25 por salón. La institución posee una fuerte historia comunitaria, ya que fue creada hace 35 años por la comunidad en una Sociedad de Fomento de un barrio de Escobar. Los niños provienen de sectores populares, centralmente de una comunidad boliviana, de la cual se destaca la preocupación de los padres por la educación y de ahí la clave del apoyo a la institución y la alta valoración de lo que pedagógicamente allí sucede.

Los primeros acercamientos que el Jardín 913 realiza con la robótica comienzan en el 2020 con Edith, maestra de sala de 5 - turno mañana. Además de su capacidad para encender la antorcha de la robótica, logra sumar a su paralela y amiga de la sala de 5 años del turno tarde, Fernanda. Así, comienzan a realizar actividades vinculadas a dicha área.

En este sentido, interesa señalar el perfil híbrido que tiene Edith. En particular, entendemos que su pasaje por la escuela técnica configura una relación con las tecnologías, junto con saberes y habilidades diferenciales:

En floricultura por ejemplo si teníamos que hacer un invernáculo a nosotros nos enseñaban desde albañilería, electricidad, plomería. Todo para construir el invernáculo, o sea todo lo que necesites, todo lo que va adentro, todo lo que lo hace funcionar. Nosotros teníamos que tener conocimiento no sólo de la parte de las plantas hasta mecánica tuvimos porque la parte de horticultura teníamos que manejar el tractor y tener que saber arreglarlo. Entonces, un tractor tiene un motor muy básico digamos entonces bueno desarmar un motor y conocer lo que era (Entrevista Edith)

Este tipo de acercamientos a los artefactos y sobre todo a las diferentes formas de resolución de problemas gestan relaciones y representaciones más cercanas con las tecnologías[23] [24]. En efecto, se va auscultando como forma de resolución de problemas el ensayo y error, el armar y desarmar, entre otros.

Además de su formación como docente de nivel inicial, se identifica que su interés por las tecnologías digitales se inició hace dos décadas: “mis hijas eran chicas, hará unos 20 años”, señala Edith. Sin embargo, en los últimos años, a partir de la expansión de dichas tecnologías cobró un nuevo impulso:

Yo creo que cuando empezamos a tener un poquito todos la posibilidad de comprarnos una computadora ahí yo empecé a decir bueno si ya la computadora puede llegar al hogar, si yo ya me puedo comprar una computadora así como me la compro yo van a empezar a comprarse todos y esto de no puedo permitirme como docente que un alumno mío de 5 años de repente sepa más que yo, por lo menos tengo que acompañarlos (Entrevista Edith)

Tal como se advierte en el relato de Edith, sus saberes y habilidades, junto con sus diferentes formas de resolución de problemas para con las tecnologías, se conjugan con una mirada particular respecto a las tecnologías digitales. Específicamente, se destaca, por un lado, la lectura que realiza respecto a la necesidad formativa y, por otro, a la centralidad de dichas tecnologías en los diferentes órdenes de la vida.

Los diferentes cursos realizados por Edith, además de saberes y habilidades relacionados con la ofimática, sientan las bases para armar una red de actores que colaboran en diferentes dimensiones en la implementación del proyecto tanto al interior de la institución como por fuera de ella. En efecto, se identifican que comulgan un conjunto de dimensiones: curricular, institucional, comunitaria y pedagógica-didáctica. Aquí, nos dedicamos a una parte de la experiencia.

## 2.3. La Experiencia

En este apartado se presentan una serie de hallazgos en relación a lo que se consideran experiencias genuinas de trabajo pedagógico. La maestra Edith comenzó a desarrollar diferentes materiales para utilizar junto con sus estudiantes. Así, incorporó a sus prácticas un conjunto de recursos educativos propios que complementaron, expandieron y enriquecieron las prácticas de enseñanza y aprendizaje, entre las cuales se destacan:

Cuando yo arranco, yo digo después de haber hecho esta capacitación de robótica yo no puedo venir y que solamente quede lo que es digamos, la programación por decirlo de alguna forma yo quiero un poquito más allá, sino para que hice la capacitación entonces largo con Robotina les muestro, empezamos a trabajar les explico como se usa empezamos a usar, y después les empiezo a decir, bueno, a ver esta Robotina nos da la posibilidad que podemos ver lo que tiene adentro y ahí empiezo a decirles y a contarles y a hacerlos pensar a ver que se

imaginaban ellos, que tenía esto este muñequito para que funcione, este robot para que funcione que debe tener y ahí empezar a decirles, bueno, a ver acá tiene tal cosa, acá tiene el otro esto es tal, esto es tal y empezar a contarles como se había armado para que funcione lo largué. (Entrevista Edith)

Tal como se advierte del relato de la maestra pionera, las primeras experiencias con robótica estuvieron de la mano y la documentación aportadas por los planes de dotación de recursos nacionales y provinciales,. Sin embargo, dicho actor fue más allá en la experiencia de aprender y enseñar, generando nuevos interrogantes que le permitieron ir construyendo un camino propio en relación a la computación en su jardín.

Yo con ese diseño después me armé el proyecto porque bueno, hasta ahí fue eso y después que hice eso yo dije bueno, yo quiero más quiero ver, cuando empecé a ver de que se desarmaban las impresoras, de que sacábamos motorcitos de que cómo los podía usar, que podía hacer funcionar diferentes cositas y que los chicos vean cómo esos motores funcionaban. Dije voy a hacer un poquito más. Está bien jugamos con la robotina, pero que sepan que ellos o sea, yo quería que en su cabecita que salgan de acá creyéndose capaz de que en algún momento de su vida pueden llegar a ser un robot perfecto porque yo sé que en primaria esto se corta entonces yo dije bueno pero que salgan de acá sabiendo que ellos pueden hacer un robot que tal vez ahora todavía no pero que ellos van a poder en algún momento de su vida hacer un robot si se los proponen entonces bueno, armé el proyecto. (Entrevista Edith)

En el testimonio anterior emerge la idea de que los estudiantes tengan experiencias lúdicas con componentes electrónicos tangibles, que les permitan deconstruir artefactos digitales y así construir una comprensión más amplia y acabada. Es decir, que los objetos sirvan para pensar sobre otras cosas.

Edith también presenta la idea de robot doméstico. Aquí, interesa destacar su vocación porque las y los estudiantes advierten que los robots están en todos lados y están integrados por diferentes componentes:

Lo primero que les traje fue la barredora y en esa fue como que yo empecé a alargar y a decirle bueno fíjense lo prendimos tiene cable, lo tengo que enchufar pero anda solo, no con el cable así que tiene una batería después viene y se carga la batería acá les fui contando cómo funcionaba fíjense qué hace cuando choca con algo. Me decían da la vuelta y sale, ¿y por qué les preguntaba? Les decía porque tiene un sensor (...) le ponían el pie para que la barredora no empezara a funcionar. Cuando terminamos, guardamos la barredora y después los largué con los motorcitos (...) Entonces, les digo si ustedes tuvieran que hacer un robot barredor qué cosas necesitarían (...)” (Entrevista Edith)

Junto con la idea de robot doméstico, se orquestó una suerte de desarmadero. Esta actividad consiste básicamente en desarmar y armar artefactos tecnológicos y ver si algunas funcionan separadas, qué se puede armar, etc.

Uso todo ese kit para mostrarle los componentes a los chicos para que ellos los clasifiquen, los miren vean, los puedan manipular entonces tengo armado todo eso para que ellos puedan estar en contacto directo con las piezas y después de que están en contacto directo con las piezas empezaron a sacar motorcitos. Yo no tenía todo entonces yo les decía, bueno esto va a ser esto que es un motor, esto que es tal cosa bueno miren, esto puede ser la carcasa, esto pueden ser unas ruedas esto puede ser un sensor entonces los chicos después de ahí sacaban puedo decir que armaban un robot, que sacaban las cosas para hacer un robot y encima ellos lo ponían yo había traído. Me acuerdo que a una tapa de helado le ponían las ruedas y alrededor le ponían los plumeritos, le ponían el motor le ponían los botones y uno lo miraba y decía, y falta que ande. (Entrevista Edith)

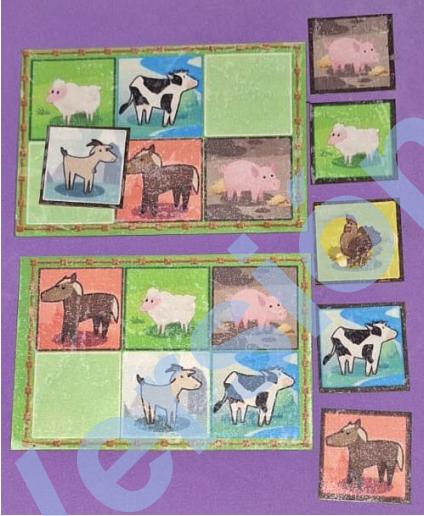
Tal como puede advertirse, la gran mayoría de los materiales y artefactos utilizados en las diferentes experiencias relatadas han sido sustentadas por las propias docentes. Sin olvidar el tiempo extra jornada laboral que supone tener los materiales en buen estado.

Las experiencias genuinas de trabajo son el resultado de exploraciones, ensayos, prototipos, prueba y error que se dieron por un conjunto de factores. Principalmente, motivaciones propias de la maestra Edith, las cuales están mucho más allá de reproducir únicamente lo que señalan los lineamientos y los materiales tradicionales, junto con la bibliografía de referencia. En efecto, estos primeros acercamientos inauguran y dejan ver que es posible realizar un proyecto genuino relacionado con la computación y sentaron las bases para el Proyecto ¡Robótica para aprender! que se lleva adelante a nivel institucional.

### 3. Resultados

Hasta el momento, el equipo asesor externo ha desarrollado una serie de materiales didácticos para promover el desarrollo del Pensamiento Computacional en el jardín de infantes. La tabla 2 ilustra algunos de los mismos, como se observa se trata de presentar experiencias de aprendizaje donde se promuevan distintas capacidades, como ser el reconocimiento de patrones, el razonamiento la planificación, la evaluación, el pensamiento algorítmico, la orientación espacial entre otras.

Tabla 2. Prototipos de materiales educativos para desarrollar capacidades en relación al Pensamiento Computacional

 <p><b>TAPA TRIÁNGULOS</b></p> <p>ACOMODA LAS TAPITAS SOBRE LOS TRIÁNGULOS HACIENDO COINCIDIR LOS COLORES CON EL SENTIDO DE LA FLECHAS</p> <p>Pensamiento lógico, orientación espacial</p>	 <p><b>UNIPE</b></p> <p>Planes y planes. Juego de armado de secuencias de instrucciones. Desarrolla el pensamiento algorítmico, la capacidad de evaluación y la resolución de problemas.</p>
 <p><b>MIS ANIMALITOS</b></p> <p>Mis animalitos. Juego de unión de pares de objetos. Desarrolla la memoria de trabajo, la flexibilidad cognitiva y el razonamiento.</p>	 <p><b>UNIPE</b></p> <p>Clasifica por color y sonido. Desarrolla la capacidad de clasificar y la concentración</p>

	
<p>¿Cuál sigue? Actividad destinada a promover la capacidad de reconocer patrones.</p>	<p>Busca los pares. Actividad destinada a desarrollar la percepción visual espacial y la abstracción.</p>

Actualmente, el Proyecto ¡Robótica para aprender! está en pleno desarrollo. Es decir, las actividades que se diseñaron a partir de los Talleres temáticos, junto con algunos intercambios, están siendo implementados -cuál prototipos de actividades-, junto con una rúbrica que se propone para realizar una evaluación de estos. Así, a partir de estas primeras implementaciones, en todas las salas del Jardín 913, se contará con mayor información acerca de las actividades co-diseñadas: dónde se incluyeron, de qué forma se presentaron, cómo se organizaron/dinamizaron y evaluaron.

## 4. Conclusiones

Esta comunicación tuvo por finalidad, por un lado, compartir algunas experiencias genuinas de trabajo de un jardín de infantes público de la Provincia de Buenos Aires en relación al desarrollo de saberes de la computación y, por otro lado, presentar el proyecto institucional denominado ¡Robótica para aprender! Una experiencia de trabajo pedagógico conjunta entre la comunidad docente del jardín de infantes e investigadores externos, que se basa en el co-diseño de materiales educativos y actividades.

En relación al primer objetivo, se recuperan las experiencias de trabajo genuino realizadas por la maestra pionera, principalmente, la idea de robot doméstico, desarmadero y reciclado y deconstrucción de artefactos. Aspectos que entendemos tienen un valor didáctico potencial ya que brindan una visión más integral y crítica de las tecnologías digitales y sus componentes y, al mismo tiempo, sientan las bases para aprender de los objetos y generar nuevos conocimientos a partir de analogías. En relación al segundo objetivo se presentaron los criterios de trabajo según edad de los estudiantes. Esto resulta un insumo de valor ya que permite secuenciar los diseños según franjas etarias y etapas de desarrollo. Bajo este marco se encuentra en proceso de co-diseño, entre la comunidad docente y los asesores pedagógicos, una serie de materiales y actividades modelo para implementar en las diferentes salas del JI 913. Luego, a modo de ejemplo, se presentaron algunos prototipos que surgieron del trabajo conjunto y actualmente se están poniendo a prueba y evaluando en las salas del Jardín de Infantes.

Tal como hemos demostrado en estas páginas la incorporación de las ciencias de la computación en el nivel inicial excede por mucho al uso de los artefactos, sino que posibilita y sienta las bases para una estructuración de las funciones superiores de pensamiento. En particular, aquí se destacan las actividades desenchufadas que pueden realizarse en las distintas salas, comenzando por la de tres años, y suponen un trabajo entre los diferentes espacios curriculares con materiales accesibles y económicos. Así, entendemos que experiencias como las relatadas aquí suponen potenciales aportes al diseño de políticas públicas en relación a la educación digital para el nivel inicial.

El proyecto continúa hasta completar una primera colección de materiales y actividades propias para el desarrollo del Pensamiento Computacional en el nivel inicial, la cual se compartirá con la comunidad docente y se espera que sea un insumo de valor para repensar y expandir las prácticas. Por otro lado, a futuro y como segunda etapa, se proyecta diseñar una serie de experimentos que permitan evaluar de manera concreta las distintas mejoras en el saber en computación y en el rendimiento académico en general a partir de incorporar estos nuevos contenidos, materiales y actividades.

## 5. Agradecimientos

Los autores desean agradecer especial y profundamente a la comunidad del Jardín de Infantes 913 “Juana Azurduy de Padilla” de la ciudad de Belén de Escobar por su generosidad y predisposición a trabajar juntos en este proyecto pedagógico.

## Referencias

- [1] E. Ferreiro, Alfabetización digital: ¿ De qué estamos hablando?. *Educação e Pesquisa*, 37, pp. 423-438, 2011.
- [2] A. Vee, *Coding Literacy. How Computer Programming Is Changing Writing*. The MIT Press, 2017
- q1) E. Saxe, A. Murillo, “Construcción: objetos para pensar, entidades públicas y micromundos”. *Actualidades investigativas en Educación*, 4(1), 2011
- [3] J. Wing, “Computational Thinking”. *Communications of the ACM*, 49(3), pp. 33-35, 2006.  
<https://doi.org/10.1145/1118178.1118215>
- [4] S. Grover, R. Pea, “Computational thinking in k-12. A review of the state of the field”. *Educational Researcher*, 42(1), pp. 38-43, 2013. doi: 10.3102/0013189X12463051
- [5] A. Diamond, “Executive functions”. *Annual review of psychology*, 64, pp. 135-168, 2013.
- [6] F. Stelzer, M. Cervigni, “Desempeño académico y funciones ejecutivas en infancia y adolescencia. Una revisión de la literatura”. *Revista de investigación en educación*, 9(1), pp. 148-156, 2011.
- [7] J. Robertson, S. Gray, M. Toye, J. Booth, “The relationship between executive functions and computational thinking”. *International Journal of Computer Science Education in Schools*, 3(4), pp. 35-49, 2020.  
<https://doi.org/10.21585/ijceses.v3i4.76>
- [8] M. Hermida, M. Garzón, M. Martínez, “Just using computers for any subject is not enough to acquire computational thinking in early childhood education”, *Universidade Presbiteriana Mackenzie; Psicología*; 25; 1; 12-2022, pp. 1-4, 2022.
- [9] Montuori, C., Gambarota, F., Altoé, G., & Arfé, B. The cognitive effects of computational thinking: A systematic review and meta-analytic study. *Computers & Education*, 210, 1–23, 2024.  
<https://doi.org/10.1016/j.compedu.2023.104961>
- [10] Fundación Sadosky (2013,) CC – 2016. Una propuesta para refundar la enseñanza de la computación en las escuelas Argentinas [en línea]. Disponible en: <http://www.fundacionsadosky.org.ar/wp-content/uploads/2014/06/cc-2016.pdf>
- [11] T. Bell, “Establishing a nationwide CS curriculum in New Zealand high schools” *Communications of the Association for Computing Machinery*, 57, pp. 28–30, 2014
- [12] M. Borchard, I. Roggi, (2017, ) “Ciencias de la computación en los sistemas educativos de América Latina”, Unesco, [en línea]. Disponible en <https://unesdoc.unesco.org/ark:/48223/pf0000372138>
- [13] S. Furber, “Shut down or restart? The way forward for computing en UK schools”. Technical Report The Royal Society, London. 2012.
- [14] European Education and Culture Executive Agency (2022, ) Informatics education at school in Europe [en linea]. Disponible en <https://eurydice.eacea.ec.europa.eu/publications/informatics-education-school-europe>
- [15] K-12 Computer Science Framework Steering Committee, The K-12 Computer Science Framework. ACM, 2016
- [16] Consejo Federal de Educación, Resolución 343/18. Consejo Federal de Educación, CFE, 2018.
- [17] Dirección General de Cultura y Educación. Provincia de Buenos Aires (2022, diciembre 20) Diseño Curricular. Educación inicial, versión 2022 [en línea]. Disponible en: <https://acortar.link/L9tBjk>
- [18] A. Falbel, ”Construcción”. Costa Rica: Fundación Omar Dengo, Programa de Informática Educativa MEP – FOD. 1993
- [19] A. Zavala Vidiella, *La práctica educativa. Cómo enseñar*. Barcelona: Graó, 2000
- [20] E. Ackermann, ”Constructivismo(s): raíces compartidas, caminos cruzados, múltiples legados”. *Construcción* 2010.
- [21] L. Vazquez, (2023, marzo 19) La repregunta. Marina Umaschi Bers: “En los niños, el pensamiento computacional se puede desarrollar sin gastar en computadoras”. *La Nación* [en línea]. Disponible en <https://acortar.link/LwuwJP>

- [22] Resnick, M & Rosenbaum, E.(2013) .“Designing for tinkerability”. En Honey, M. y Kanter, D. (Eds.) “Design, make, play: Growing the Next Generation of STEM Innovators, pp. 163-181. Routledge.
- [23] Dughera, L.; Yansen, G. y Zukerfeld, M. (eds.) (2012) *Gente con códigos. Una aproximación a la heterogeneidad de los procesos productivos de software*, Universidad Maimónides, Buenos Aires.
- [24] Dughera, L. y Bordignon, F. (2021). Saberes digitales en tiempos de incertidumbre en Aparici, R. y Martínez-Pérez, F. (coords.) *El algoritmo de la incertidumbre*, pp. 71-84. Barcelona: Gedisa.

Versión Preliminar

# **La enseñanza de la informática en la Educación Semipresencial de Jóvenes y Adultos**

Rocha Kermolj Bárbara, Molina Mariana, Martínez María, Perales Raquel,  
Dominguez Gabriel, Malik de Tchara Constanza,

Ministerio de Educación de la provincia de Córdoba. Dirección General de Educación de Jóvenes y Adultos, Córdoba ,  
Argentina  
[educacionadistanciacba@me.cba.gov.ar](mailto:educacionadistanciacba@me.cba.gov.ar)

## **Resumen**

El propósito de relatar esta experiencia es acercar a otros/as colegas docentes la construcción pedagógico - didáctica realizada en la orientación en informática del Programa de Educación a Distancia de Jóvenes y Adultos y el Programa Educativo Combinada en Línea, ambos dependientes de la Dirección General de Jóvenes y Adultos de la provincia de Córdoba.

Centraremos nuestra presentación en la articulación entre educación y trabajo en el área técnico profesional de ambos programas, donde se desarrolla la orientación en informática. Se presenta una selección de contenidos **y** recursos organizados en secuencias didácticas que apuntan, por un lado, a garantizar el acceso a la alfabetización y ciudadanía digital de las y los estudiantes de la Modalidad Jóvenes y Adultos, y por otro, a promover aprendizajes asociados a la cultura digital que puedan facilitar su inserción laboral.

## **Palabras clave**

Educación permanente de jóvenes y adultos; Educación a distancia; Educación y trabajo; Área técnico profesional; Orientación en informática

## **1. Introducción**

En este trabajo buscamos compartir nuestra experiencia en la construcción curricular y didáctica de la orientación en informática en un contexto de educación formal a distancia para el nivel secundario de jóvenes y adultos.

Somos integrantes del equipo técnico del Programa de Educación Semipresencial a Distancia y del Programa Combinada en Línea, ambos pertenecientes a la Dirección General de Educación de Jóvenes y Adultos del Ministerio de Educación de la provincia de Córdoba.

### **1.1 Marco normativo: El Programa de Educación a Distancia y el Programa Combinada en Línea**

La Ley de Educación Nacional 26.206/2006 (LEN) en su artículo 17 estructura el sistema educativo nacional en cuatro niveles (educación inicial, primaria, secundaria y superior) y en ocho modalidades. Una de ellas es la Educación Permanente de Jóvenes y Adultos (EPJA) dentro de la cual se inscribe nuestro trabajo: el Programa Semipresencial de Educación a Distancia para Jóvenes y Adultos y el Programa Educativo Combinada en Línea. Ambos programas son propuestas de educación semipresencial a distancia para finalizar el nivel secundario.

Desde su creación, ambos programas han formado parte de una propuesta integral de políticas públicas que apuntan a la responsabilidad que tiene el Estado, junto a las y los educadores, de dar respuestas institucionales específicas a la situación de personas que por diferentes razones han tenido que discontinuar sus estudios y que hoy deciden re-vincularse a la escuela, pero no pueden asistir todos los días a clases. De esta manera, el Estado hace efectivo el derecho a la educación mediante una propuesta que posibilita la terminalidad educativa.

La propuesta se caracteriza por tener un modelo de gestión conjunta que articula, mediante la firma de convenios específicos, las capacidades y acciones del Ministerio de Educación de la provincia de Córdoba y la Dirección de Educación de Jóvenes y Adultos con distintas entidades conveniantes como organizaciones sindicales, empresas, organismos públicos, cooperativas, asociaciones civiles, entre otras.

En definitiva, se trata de una propuesta que busca articular el ámbito educativo formal con el espacio social y laboral de las y los destinatarios que ya han cumplido con otros trayectos educativos dentro del sistema formal y que portan además saberes adquiridos en otros ámbitos.

El Programa de Educación a Distancia nació en el año 2001 y fue transformándose y expandiéndose en función de la demanda educativa de las y los destinatarios y las entidades conveniantes. En el año 2016 el Consejo Federal de Educación aprobó una actualización de la propuesta dando inicio a la orientación en informática. En 2017 iniciamos desde el equipo técnico el diseño de una versión del Programa para plataforma Moodle junto al Instituto Superior de Estudios Pedagógicos (ISEP) que se puso en marcha en 2021, luego de las demandas surgidas durante la pandemia de COVID-19. Luego de una prueba piloto con 8 sedes con orientación en informática, la propuesta se migró a un servidor de la Universidad Provincial de Córdoba para su expansión e implementación de las demás orientaciones. Así nació el Programa Educativo Combinada en Línea [1] que reformula y adecúa los contenidos del Programa de Educación a Distancia [2] a un entorno virtual. La propuesta en línea se basa en el mismo diseño curricular pero adquiere algunas características propias del entorno virtual, que impactan directamente en los contenidos de informática.

## **1.2 La articulación entre educación y trabajo como marco general de la enseñanza de la informática en la Educación a Distancia de Jóvenes y Adultos (EDJA)**

La educación permanente y el trabajo son pilares esenciales de toda sociedad democrática: todas las personas tenemos derecho al trabajo decente y a la educación pública y de calidad porque somos sujetos de derechos. Cuando hablamos de educación permanente nos referimos a la finalización de los estudios secundarios, a la formación superior, a alguna carrera universitaria o superior y a la formación profesional (FP).

La vinculación entre educación y trabajo es considerada un eje estratégico para el desarrollo nacional en la Ley de Educación Nacional N°26.206/2006 y en la Ley de Educación Técnico Profesional N°26.058/2005. Esta última establece que la educación técnica profesional (ETP): “*Es un derecho que se hace efectivo a través de procesos educativos, sistemáticos y permanentes*” (art.3o) y “*(...) promueve en las personas el aprendizaje de capacidades, conocimientos, habilidades, destrezas, valores y actitudes (...) que faciliten la inserción y reflexión sobre el mundo del trabajo*” (art. 40).

En nuestra provincia existe una demanda de la población en cuanto a formación técnico-profesional, ésta se acrecienta a medida que el contexto social y económico profundiza las situaciones de vulnerabilidad. En este escenario, surge la necesidad de promover el desarrollo de capacidades para la inserción social, cultural y económica de jóvenes y adultos, permitiéndoles capacitarse laboralmente y reconociendo los saberes construidos en otros ámbitos.

Tomando como referencia la resolución N°308/2016 del Consejo Federal de Educación (CFE) la articulación entre formación profesional y educación general “*debe constituirse en un espacio institucional amplio, abierto y dinámico, superador de las lógicas endogámicas de las sedes EDJA y los centros de FP. Cada Jurisdicción que adopte este dispositivo debe garantizar que su función formativa se desarrolle con una fuerte vinculación con la sociedad y el mundo del trabajo.*”

Por su parte, la Ley de Educación de la Provincia de Córdoba N°9870/2010 amplía lo previsto por la LEN. En este marco se construyeron los diseños curriculares de la modalidad, para el Nivel Secundario presencial a través de las Resoluciones N° 84/2011 [3] – Ciclo Básico y la N° 148/2013[4] – Ciclo Orientado.

Por último, las Resoluciones del Ministerio de Educación de la Provincia de Córdoba, N°152/2016 [5], N°176/2016 [6] y N° 795/2022 [7] actualizan los marcos normativos para la implementación de las ofertas de Educación a Distancia en la provincia de Córdoba para la Modalidad de Educación Permanente de Jóvenes y Adultos.

## **1.3 Los desafíos de la orientación en informática**

El diseño curricular al que remiten ambos programas incorpora a la informática como una de las ocho orientaciones disponibles para completar el nivel secundario en la Modalidad Jóvenes y Adultos (Agro y ambiente, Arte y cultura, Deportes y recreación, Economía y administración, Electricidad, Informática, Metalmecánica y Turismo).

Para la construcción curricular y didáctica de esta orientación, partimos de entender que la enseñanza y el aprendizaje en la Modalidad presentan un conjunto de particularidades específicas y enfrentan una serie de desafíos singulares relacionados a las características de las y los estudiantes, al modo de cursado, y a las condiciones materiales de las diferentes sedes.

En primer lugar, las y los estudiantes de la Modalidad conforman una población muy heterogénea de jóvenes y adultos/as mayores de dieciocho años, con trayectorias escolares discontinuas que se acercan nuevamente a la educación con la necesidad de finalizar sus estudios secundarios. Son estudiantes dispersos a lo largo de toda la provincia que presentan grandes diferencias en sus trayectorias educativas y laborales, en su realidad económica y en sus expectativas y posibilidades respecto a la finalización de sus estudios. Tal como lo afirma nuestro diseño curricular, sostengamos que existe “*En nuestra provincia una gran demanda de la población en cuanto a formación técnico-profesional, que se acrecienta a medida que el contexto social y económico profundiza las situaciones de vulnerabilidad.[8]*” Por ello nos propusimos construir una propuesta que fuera significativa para esta población educativa perteneciente a la PEA (Población Económicamente Activa) en la diversidad de sus contextos, y que favoreciera la mejora de su situación laboral o su incorporación al mundo del trabajo, con el dinamismo que caracteriza al sector de la informática.

En segundo lugar, la educación semipresencial a distancia requiere una propuesta didáctica muy distinta al de las propuestas áulicas presenciales. En esta propuesta, el material de estudio que elaboramos se asienta en la recuperación de saberes previos y promueve el trabajo autónomo y asincrónico.

Por último, la precariedad de la infraestructura de las instituciones educativas es una gran limitante para la implementación de la orientación en informática. La mayor parte de las sedes no disponen de computadoras y/o conexión a internet. Se busca resolver estas limitaciones con el aporte de las entidades convenientes o, más recientemente, la articulación con Puntos Digitales [9]. Es importante aclarar que el Programa Conectar Igualdad no alcanzó a la modalidad, aunque estaba previsto que lo hiciera en 2016.

En las siguientes páginas detallaremos las características de la propuesta que elaboramos en este contexto y con las limitaciones y desafíos mencionados.

## 2. Descripción de la Experiencia

### 2.1. Contexto y destinatarios/as

La educación semipresencial es una opción pedagógica y didáctica donde la relación docente-estudiante se encuentra separada en el tiempo y/o en el espacio, durante todo o gran parte del proceso educativo. Por ello, la dinámica se asienta en elementos mediadores –en formato papel o de carácter digital- diseñados especialmente para que las y los estudiantes puedan acceder a la propuesta educativa.

En la provincia de Córdoba, las y los estudiantes pueden optar por dos formas de cursado semipresencial enmarcadas en el mismo diseño curricular:

- El Programa de Educación Semipresencial a Distancia, que utiliza módulos -disponibles en PDF [10] - como material mediador.
- El Programa Educativo Combinada en Línea, que adapta dichos módulos, la dinámica de cursado y evaluación para un entorno digital [11].

La implementación de los programas se lleva adelante en sedes. Allí se desarrolla la actividad administrativa y se realizan los encuentros entre tutores/as docentes y estudiantes. Las sedes dependen de un CENMA (Centro educativo nivel medio de adultos) y pueden funcionar en un espacio educativo o en lugares pertenecientes a las entidades convenientes.

Actualmente existen 203 sedes del Programa de Educación a Distancia [12] y 23 sedes del Programa Combinada en Línea[13] distribuidas en 11 zonas de inspección en todo el territorio provincial, de las cuales 56 sedes tienen la orientación en informática.

Las y los estudiantes de la modalidad son sujetos mayores de 18 años -o que cumplen 18 en el transcurso del año escolar- que no han podido finalizar sus estudios en la escuela secundaria regular y/o presentan trayectorias discontinuas por diferentes motivos. Pertenecen a grupos heterogéneos con diferentes trayectorias escolares, laborales y con proyectos de vida diversos. Son sujetos de derecho y sujetos sociales históricos capaces de transformar la realidad. Por ello no deben ser abordados de manera aislada sino teniendo en cuenta sus diversidades y su contexto.

## **2.2. Propuesta didáctica**

La experiencia de la enseñanza de informática en nuestra modalidad Semipresencial de Jóvenes y Adultos se enfrenta a un conjunto de desafíos en cuanto al cursado que tienen los estudiantes. Por eso, a continuación les presentamos una breve descripción de la dinámica en la cual se desarrollan los procesos de enseñanza y aprendizaje.

El diseño curricular de nuestra modalidad tiene una estructura modular y está organizada en cuatro áreas generales y el área técnico profesional conformada por dos módulos (8 y 9 informática). Este proceso está mediado por materiales de estudio (módulos) construidos por el equipo técnico de la modalidad.

El programa es semipresencial, no se asiste diariamente a clases, sino que contamos con tutorías semanales y de carácter no obligatorio para las y los estudiantes. En el caso del Programa Combinada en Línea los encuentros se desarrollan de manera virtual sincrónica. En ambos casos son un momento de encuentro entre estudiantes, tutores/as docentes de cada área o disciplina y/o coordinador/a pedagógico/a de la sede. Se proponen como espacios de acompañamiento para el abordaje de los contenidos, la resolución de actividades, trabajos prácticos integradores y la preparación de las instancias evaluativas.

Las **tutorías** son instancias muy importantes ya que la sola interacción de los y las estudiantes con los materiales no garantiza el aprendizaje: el rol de coordinador/a pedagógico/a y tutores/as docentes es central para acompañar en la apropiación de los materiales de estudio, la resolución de ejercitaciones y actividades y la reflexión sobre el propio proceso de aprendizaje. Las intervenciones se realizan en función de las trayectorias, buscando que cada estudiante potencie sus capacidades, organice sus recursos para el estudio y ponga en juego estrategias para afrontar los desafíos que le va presentando el proceso de aprendizaje, e incrementando paulatinamente su nivel de autonomía.

## **2.3. La experiencia: Nuestra propuesta de articulación de educación y trabajo: el Área Técnico Profesional (ATP) y la orientación en informática**

### **2.3.1 - El Área Técnico Profesional**

La construcción de esta área se encuadra en el marco normativo citado anteriormente. “El Área Técnico Profesional está formada por aquellos saberes que facilitan la inserción, aproximación y reflexión sobre el mundo del trabajo. Específicamente, la adquisición de saberes que les permitan a las/os estudiantes comprender las relaciones sociales y económicas-productivas que definen el mundo laboral y el aprendizaje y ejercicio de los derechos y obligaciones propias del derecho del trabajo y la seguridad social.”[14]

El enfoque curricular supone que las y los sujetos se relacionan activamente con su contexto para transformarlo, promoviendo la reflexión crítica acerca del mundo del trabajo, de las relaciones de las personas con lo que producen, de las/os trabajadoras/es entre sí y de sus condiciones de producción y empleabilidad. Esto aporta a la formación integral de jóvenes y adultos, parte de la Población Económicamente Activa (PEA) y se puede transferir y/o relacionar con las prácticas sociales y ciudadanas del mundo del trabajo, convirtiendo los contenidos de los módulos en aprendizajes significativos.

La propuesta de ambos programas está organizada en nueve módulos de los cuales el 8 y el 9 conforman el Área Técnico Profesional. El módulo 8 está compuesto por los siguientes espacios curriculares: Ciudadanía y participación; Formación para el trabajo, Derecho del trabajo y seguridad social, Formación profesional y Problemáticas económicas actuales. Al mismo tiempo, los contenidos desarrollados en el módulo 8 se vinculan con los de cada módulo 9 (Espacio de la orientación y Espacio de vinculación con el sector) que corresponden a la orientación específica de cada sede.

### **2.3.2 - La orientación en informática**

En este marco, “la informática como campo del conocimiento, posee un valor que supera los fines meramente utilitarios o de herramientas. Abarca tanto las actividades de investigación, diseño y desarrollo, como los productos resultantes de las mismas.”[15] Por ello, nuestro enfoque busca incorporar herramientas informáticas básicas de manera transversal a lo largo de toda nuestra propuesta curricular, haciendo explícitas ciertas concepciones y representaciones sociales en torno a la relación entre la informática y la educación.

Entendemos que las tecnologías desempeñan un papel importante en la producción de conocimiento, en las situaciones didácticas propuestas se afianzan los saberes que las/os estudiantes tienen en cuanto al uso de las tecnologías, el análisis de las situaciones y la resolución de problemas con diferentes soluciones posibles. Por ello buscamos plantear la orientación de manera integral, promoviendo la construcción de una ciudadanía digital responsable, identificando las posibilidades que brinda el acceso a internet y los cuidados para hacerlo.

“Considerando las implicancias socioculturales del desarrollo de las TIC, abordamos temas tales como relaciones y redes sociales, redes de distribución de la información, consecuencias de la integración de las tecnologías en los procesos productivos.”[16]

En este sentido, la orientación en informática, aporta para la formación de un/a estudiante crítico/a, capaz no solo de transformar, sino también intervenir en el mundo actual y futuro; permitiendo complejizar el análisis y la reflexión de problemáticas referentes al acceso y recepción de las tecnologías y sus plataformas.

En el marco de esta orientación para jóvenes y adultos abordamos la formación de las/os sujetos desde una perspectiva que articula el dominio de las herramientas informáticas con las distintas manifestaciones comunicacionales, que potencie la creatividad para desenvolverse en diferentes escenarios sociales y que incrementa sus probabilidades de acceso y permanencia en el mundo del trabajo. (Resolución N° 148/2013 Ciclo Orientado).

### **2.3.4 El Área Técnico Profesional: ejes organizadores y dimensiones del mundo del trabajo. Contenidos: selección, organización y secuenciación.**

El propósito del Área Técnico Profesional es desarrollar y actualizar, en el marco de la educación permanente y continua, las capacidades de personas jóvenes y adultas para el mundo del trabajo y/o para continuar con estudios superiores (pág. 61 Res. N° 84/2011).

Se organiza en tres ejes presentes en el plan de estudios (Res N°148/2013):

- Las relaciones socioeconómicas productivas y el mundo del trabajo
- La perspectiva legal del mundo del trabajo
- Formación y preparación para un campo profesional-ocupacional específico

El Área Técnico Profesional está estructurada en dos bloques, construidos desde los enfoques de Derechos Humanos y perspectiva de género:

**1º bloque:** Tiene relación con el mundo de trabajo y algunas de sus dimensiones:

1. Las relaciones laborales.
2. Las políticas públicas, la ciudadanía y el Estado.
3. Los modelos de desarrollo socioeconómico.
4. La organización social del trabajo.

Abordamos estas dimensiones a partir de algunas situaciones problemáticas de la vida cotidiana, y específicamente las del mundo del trabajo de la orientación en informática.

**2º bloque:** Está orientado a reflexionar sobre ese mundo del trabajo y la Formación Profesional. Aquí conoceremos qué es un sector socio-productivo, cuándo y cómo surge dentro de un contexto regional, provincial y nacional, el mapa laboral y de formación de cada sector. Este bloque será desarrollado en el módulo 9 de informática.

### **2.3.5 - Módulo 9 - Informática**

Cuando iniciamos la construcción de la orientación, hicimos contacto a través de la ex Secretaría de Equidad y Promoción del Empleo con representantes del Clúster Científico Tecnológico de Córdoba con el propósito de informarnos sobre los conocimientos básicos requeridos para que nuestros/as egresados se insertaran en el sector socio productivo como trabajadores/as e indagar sobre la posibilidad de articular con el programa 111 mil Programadores [17], lamentablemente esto último no prosperó.

En la orientación en informática buscamos explorar y aproximarnos a diferentes opciones de inserción laboral y a la continuidad de los estudios superiores. En este caso, nuestra intención es profundizar conocimientos y realizar prácticas que nos permitan vincularnos con la informática para que nos orienten en la búsqueda, acceso y permanencia en el mundo del trabajo por medio de prácticas laborales y/o socioculturales.

Tomamos una posición crítica y reflexiva ante las situaciones problemáticas planteadas con respecto al desarrollo y al uso de la informática y de las TIC, ejerciendo el rol de ciudadanas/os digitales.

En el módulo 9 tenemos como expectativas de logro:

- Posibilitar a las/os estudiantes la adquisición y desarrollo de capacidades para promover la apropiación de nuevos conocimientos, favoreciendo la participación en la vida ciudadana, promoviendo la continuidad de estudios superiores y la inclusión y permanencia en el mundo del trabajo.

- Favorecer la autonomía para el uso responsable y eficiente de los sistemas digitales de información y comunicación a partir de las posibilidades que brindan las TIC para buscar, transformar, producir, publicar y compartir información en diferentes formatos y soportes, interactuando a través de los diferentes entornos y las redes digitales
- Analizar críticamente las implicancias económicas, sociales, culturales y éticas, relacionadas con el desarrollo Informático, tanto en el contexto local como global.
- Tomar una posición crítica y reflexiva ante las situaciones problemáticas planteadas con respecto al desarrollo uso de la informática y las TIC, el libre acceso a la información, identidades digitales, privacidad, seguridad, formas de producción y acceso al conocimiento y uso responsable de la informática como ejes constitutivos de la ciudadanía digital.
- Analizar situaciones problemáticas a resolver, evaluar alternativas, diseñar y desarrollar soluciones mediante la aplicación de saberes informáticos y de otros campos en prácticas laborales y socioculturales.
- Reconocer saberes adquiridos y fortalecer los aprendizajes necesarios para comunicarse, estudiar, trabajar y participar.
- Adquirir conocimientos básicos para diseñar y desarrollar aplicaciones informáticas (textos, hojas de cálculo, etc.) promoviendo la conformación de equipos de trabajo colaborativo.

El módulo 9 está organizado en cinco ejes conceptuales que acercan al/la estudiante al mundo de las tecnologías de la información y comunicación con una mirada integral sobre diferentes aspectos que lo componen. El eje 1 es el que concreta la articulación del área:

**El Eje 1 - El sector informático y sus características:** comienza con una introducción de por qué estudiamos informática y su importancia en el desarrollo integral de la/el estudiante. Se definen las Tecnologías de la información y la comunicación (TIC), considerando las posibilidades de acceso y conexión, definiendo conceptos claves como la brecha digital y políticas públicas. Al analizar las TIC y las actividades políticas del sector se retoman conceptos como el de aprendizaje móvil, tan en auge a partir de la pandemia CoVid-19. Se recuperan conceptos trabajados en el módulo 8 que se relacionan con el sector productivo. La caracterización de la 4º Revolución Industrial y su incidencia en el mundo laboral, relacionando conceptos claves como la Economía 4.0, la Economía del Conocimiento, la Economía de Plataforma y la Economía Circular. Se presenta el tema de los desechos tecnológicos y la basura electrónica con una instancia de reflexión. Se aborda la producción y desarrollo de TIC (hardware y software). Se describe brevemente la dinámica del mercado de trabajo, abordando la problemática de género en el sector informático. Al analizar el perfil de los/as trabajadores/as, se detallan roles. Finalmente analizamos la vinculación entre educación y trabajo a través de la propuesta formativa de la Formación Profesional.

A continuación, resumiremos el contenido desarrollado en los demás ejes:

**El Eje 2 - Principales funciones de la computadora:** se inicia con conocimientos básicos sobre la computadora y los dispositivos móviles. El software y el software libre fundamentado en la defensa de los Derechos Humanos para el acceso a la información, la educación y la cultura, considerando también las implicancias éticas, sociales, políticas y económicas del uso del software. Se busca fomentar el trabajo colaborativo y participativo, habilitando el espacio para la discusión de ideas y posibles estrategias de desarrollo, para evaluar el estado de situación, para relevar dificultades, reformular caminos, entre otras tareas a partir de lo que ya conocen.

**El Eje 3 - Internet un mundo de oportunidades:** se aborda el uso de Internet y sus características, la ciudadanía digital y seguridad donde la propuesta es complejizar el desarrollo de la seguridad informática incorporando aspectos legales del uso de Internet. Las redes sociales y su uso responsable. Los cuidados en el uso de internet (grooming, ciberbullying o ciberacoso, phishing y delitos informáticos). Análisis de la información que circula en Internet por diferentes medios. Otras formas de comunicación (sincrónicas y asincrónicas). Las páginas web y la conformación de dominios. Búsqueda de información. Los datos en internet y su importancia. El correo electrónico. La nube y sus opciones. El eje finaliza con la propuesta de trabajos prácticos de uso compartido en Google Drive.

**El Eje 4 - Sobre letras y números:** desarrolla el uso de procesadores de textos y planillas de cálculo. Vincula lo trabajado en el módulo 8 para confeccionar el currículo personal. En todo el eje se propone realizar actividades de fortalecimiento y afianzamiento de las opciones explicadas en el material para la aplicación en el estudio o el trabajo.

**El Eje 5 - Introducción a la programación:** se inicia con fundamentos generales, definiendo la programación, programa, lenguajes, clasificaciones, algoritmos y modelos. Representaciones y simbología – Diagramas de flujo. Tipos de datos. Operadores aritméticos, relacionales y lógicos a utilizar durante la programación. Introducción a la programación en bloques. Primeros pasos en Scratch como iniciación a la programación. Ejemplo completo de un juego.

### 2.3.6 - El Programa Combinada en Línea

En la versión online el contenido se organizó en cinco clases que proponen el abordaje de los contenidos de los ejes, actualizando noticias y referencias a políticas públicas disponibles e incorporando temas emergentes como la inteligencia artificial y las apuestas virtuales.

En la **primera clase** de la plataforma se reflexiona sobre el mundo del trabajo y la Formación Profesional: se conoce qué es el sector socio-productivo de la orientación en informática, cuándo y cómo surge dentro del contexto regional, provincial y nacional, el mapa laboral (opciones de trabajo) y de formación de dicho sector (qué competencias se necesitan y dónde se pueden adquirir para insertarse como trabajador/a en este sector). La propuesta es que en el abordaje de estos contenidos se pueda hacer una articulación conceptual permanente con el módulo 8.

Se trabajan temas como: El sector informático y sus características. ¿Por qué estudiamos informática? Las tecnologías de la información y la comunicación (TIC). Las posibilidades de acceso. Las y los trabajadoras/es del sector. Políticas públicas y mundo del trabajo. El derecho del trabajo. Convenios colectivos de trabajo. Las TIC y las actividades económicas del sector. 4º Revolución industrial: las tecnologías al servicio de la producción. La producción y desarrollo de TIC. Dinámica del mercado del trabajo. El mercado laboral informático y el género. Perfil de un/a trabajador/a del sector. La vinculación entre educación y trabajo: la formación profesional.

## 2.4 Impacto

Para evaluar el impacto de la implementación de este diseño curricular en las sedes con orientación en informática, podemos realizar una primera aproximación mirando las características de la población a la que alcanza y las estadísticas que presenta el Programa a Distancia en sus sedes con orientación en Informática.

Según los datos oficiales del Gobierno de la Provincia de Córdoba en su anuario estadístico 2023[18] el Programa de educación a distancia contó ese año con 10.437 estudiantes matriculados, de los cuales el 83,39% vivían en zonas urbanas mientras que 16,61% residían en sectores rurales. Nos parece relevante el dato sobre lugares de residencia ya que habla de una importante capilaridad del Programa de Educación a Distancia, que puede acercar la posibilidad de terminalidad educativa donde no existen otras ofertas.

A su vez, en el 2022, de las 57 sedes con orientación en informática se egresaron 692 personas, de las cuales 60,69% son mujeres y 39,31 son varones. Es decir que de concretarse una articulación con el mercado laboral, la propuesta podría realizar un aporte positivo a la paridad de género en el sector.

Estos son algunos números de alcance cuantitativo, pero estamos iniciando el proceso de consulta sobre implementación en las diferentes Sedes. Esta consulta busca obtener información de calidad que será insumo del proceso de actualización curricular que debe completarse en marzo del 2026.

## 3. Conclusiones

La orientación en informática enmarcada en el Programa de Educación a Distancia recibió aprobación plena por cuatro años en 2016 y su actualización, por cinco años en 2021. Consideramos estas resoluciones del Consejo Federal de Educación como un reconocimiento institucional importante para la modalidad de Jóvenes y Adultos sobre el trabajo realizado.

Hasta el momento ha sido una construcción que recibió aportes de diferentes actores del campo educativo y del sector socio productivo:

En primer lugar, la retroalimentación del trabajo cotidiano de las sedes. Las recontextualizaciones realizadas en diversos espacios de tutorías son compartidas con nosotras en encuentros presenciales y virtuales. Estos aportes pedagógicos-didácticos luego se plasman en formaciones situadas, en la elaboración de documentos y en el

blog<sup>1</sup> del Programa. Por lo tanto, la mirada sobre la apropiación de los contenidos de los módulos es un trabajo colectivo y que se realiza cotidianamente.

En segundo lugar, obtuvimos aportes del equipo de especialistas del Instituto Superior de Estudios Pedagógicos (ISEP) en el marco del lanzamiento del Programa Combinada en Línea, quienes señalaron varios aspectos a repensar. Entre ellos, la amplitud del término informática, problemáticas con las licencias de uso de programas en algunos contenidos de informática, y las relaciones entre los saberes informáticos y el mundo del trabajo.

Como resultado de estos diálogos, la actualización aprobada en el 2021 es una propuesta superadora de la anterior, ya que incorpora algunos contenidos de ciudadanía digital transversalmente en la formación general, permitiendo profundizar el desarrollo de los ejes de la orientación. A su vez, la última actualización de contenidos para el Programa Combinada en Línea contiene nuevas mejoras, por ejemplo, actividades vinculadas al pensamiento computacional. Como conclusión positiva, la construcción curricular y didáctica es un proceso espiralado y abierto a diferentes voces, que va encaminada a mejorar la propuesta y adaptarla a los cambios del sector.

Teniendo en cuenta la etapa de actualización curricular que está promoviendo el Ministerio de Educación de la provincia, como equipo técnico nos encontramos en proceso de relevar y sistematizar las experiencias resultado de la implementación de este diseño curricular en las diferentes sedes.

Hasta ahora identificamos dos desafíos concretos a abordar. Por un lado, que es necesario extraer contenidos y aprendizajes relativos a la ciudadanía digital e incorporarlos como contenidos transversales a la modalidad, y superar así el encapsulamiento de estos saberes valiosos en la orientación. Esto nos va a permitir enfocar la orientación en informática en saberes más específicos que se relacionen de manera activa con el mundo del trabajo. Esto nos lleva al segundo desafío, que es profundizar los lazos con el sector socio productivo para favorecer la articulación en términos de formación y empleabilidad.

Otro desafío interinstitucional sería encontrar instituciones, representantes del sector informático, espacios de formación, entre otros, que estén interesados en participar activamente en la revisión de la orientación en informática de la educación de jóvenes y adultos. Creemos por otro lado, que estas iniciativas no pueden dejarse libradas a la buena voluntad de algunos/as profesionales, sino que deben enmarcarse en convenios más abarcativos y articuladores del Ministerio de Educación.

Finalmente queremos agradecer y rescatar los espacios formativos como el de estas jornadas para generar redes, consensos y marcos orientadores sobre los desafíos de la educación en nuevas tecnologías.

#### 4. Referencias

- [1] Dirección General de Jóvenes y Adultos (2022) Programa Educativo Combinada en línea.[Online]. Disponible <https://campus.upc.edu.ar/>
- [2] Dirección General de Jóvenes y Adultos (2018) Programa de Educación a Distancia [Online]. Disponible <https://educacionadistanciacordoba.wordpress.com/modulos-y-curriculo/>
- [3] Ministerio de Educación (2011) Resolución N°84/11 Resolución N°84/11, [Online]. Disponible: <https://drive.google.com/file/d/1nEzRSD4dVbm4IjPrqMg-nd5lgOMAvhU/view>
- [4] Ministerio de Educación (2013) Resolución N°148/13, [Online]. Disponible: <https://drive.google.com/file/d/1oVxOg9n2XmA8eLzpvJy4HzcCrYPEpyWv/view>
- [5] Ministerio de Educación (2016) Resolución N°152/16, [Online]. Disponible: <https://drive.google.com/file/d/1R8x61std-kkOT8Tox8gdv4DDBhF09P3A/view>
- [6] Ministerio de Educación (2016) Resolución N°176/16 [Online]. Disponible: <https://drive.google.com/file/d/1hBM1c4ucXX5T84ggrixQu39DiRT1D82t/view>
- [7] Ministerio de Educación (2022) Resolución N°795/22 [Online]. Disponible: [https://drive.google.com/file/d/1yi\\_8n\\_akgZ9ppGBVifmGmPyh1LvUKTS\\_/view](https://drive.google.com/file/d/1yi_8n_akgZ9ppGBVifmGmPyh1LvUKTS_/view)
- [8] Ministerio de Educación (2021) Programa Educación a Distancia, Currículo Orientación en Informática P.2 [Online]. Disponible: [https://drive.google.com/file/d/19P4NUulZJ\\_9mY6uM-N8CSZyaRSxOXug/view](https://drive.google.com/file/d/19P4NUulZJ_9mY6uM-N8CSZyaRSxOXug/view)

<sup>1</sup> <https://educacionadistanciacordoba.wordpress.com/>

- [9] Ministerio de Educación (2024) Puntos digitales y pisos tecnológicos [Online]. Disponible: [https://www.google.com/maps/d/u/0/viewer?mid=1MVQe\\_n7R15yy9AitUCzpcZHD8W026U&femb=1&ll=-32.72797129301328%2C-63.5528619499999&z=7](https://www.google.com/maps/d/u/0/viewer?mid=1MVQe_n7R15yy9AitUCzpcZHD8W026U&femb=1&ll=-32.72797129301328%2C-63.5528619499999&z=7)
- [10] Ministerio de Educación (2021) Módulos del Programa de Educación a Distancia [Online]. Disponible: <https://www.cba.gov.ar/educacion-a-distancia-jovenes-y-adultos/>
- [11] Dirección General de Jóvenes y Adultos (2022) Programa Educativo Combinada en línea.[Online]. Disponible <https://campus.upc.edu.ar/>
- [12] Dirección General de Educación de Jóvenes y Adultos, Ministerio de Educación, Provincia de Córdoba (2024) Mapa interactivo de sedes del Programa de Educación a Distancia [Online]. Disponible: <https://www.google.com/maps/d/u/0/viewer?mid=1Adb6p96qyC2YZBi7pkYtvmwTjLdZ1ag&ll=-32.48561768024804%2C-62.64309207769746&z=7>
- [13] Dirección General de Educación de Jóvenes y Adultos, Ministerio de Educación, Provincia de Córdoba (2024) Mapa interactivo de sedes del Programa Combinada en Línea [Online]. Disponible: <https://www.google.com/maps/d/u/0/viewer?mid=1nLD83GDIzIwtfvURXaLM-wT4ugnOdkk&femb=1&ll=-32.36981689919154%2C-56.93900361259702&z=6>
- [14] Dirección General de Educación de Jóvenes y Adultos, Ministerio de Educación, Provincia de Córdoba (2021) Diseño Curricular Programa Educación a Distancia [Online]. Disponible: <https://educacionadistanciacordoba.wordpress.com/modulos-y-curriculo/>, pág. 69
- [15] Op.cit. pág 26
- [16] Op. cit. p. 26
- [17] Instituto Nacional de Educación Técnica (2017), Información institucional del Programa 111 Mil Programadores. [Online]. Disponible en: <https://www.inet.edu.ar/index.php/tag/111-mil-programadores/>
- [18] Dirección General de Planeamiento de la Educación, Evaluación y Prospectiva Educativa, Área de Información Educativa y Estadística (2023) Relevamiento Anual (RA) [Online]. Disponible en: <https://www.cba.gov.ar/wp-content/uploads/2024/02/Estadisticas-educativas-Anuario-2023.pdf>
- [19] OIT (2019). Trabajar para un futuro más prometedor. Comisión mundial sobre el futuro del trabajo. (1 edición)[Online]. Disponible [https://www.ilo.org/wcmsp5/groups/public/---dgreports/---cabinet/documents/publication/wcms\\_662442.pdf](https://www.ilo.org/wcmsp5/groups/public/---dgreports/---cabinet/documents/publication/wcms_662442.pdf)
- [20] Madariaga, J., Buenadicha, C., Molina, E. y Ernst, C. (2019). *Economía de plataformas y empleo ¿Cómo es trabajar para una app en Argentina?*, CIPPEC-BID - OIT. Buenos Aires, 2019.
- [21] OIT (2020, Junio) La COVID-19 y el mundo del trabajo. (Quinta edición) [Online]. Disponible: [https://www.ilo.org/wcmsp5/groups/public/@dgreports/@dcomm/documents/briefingnote/wcms\\_749470.pdf](https://www.ilo.org/wcmsp5/groups/public/@dgreports/@dcomm/documents/briefingnote/wcms_749470.pdf)
- [22] L.Rodríguez, “Saberes, saberes socialmente productivos y educación de adultos”. Universidad Nacional de Buenos Aires.pp 55-60, 2011

# Circuitos Creativos

Julián Fernández<sup>1</sup>, Marcela Rivero<sup>2</sup>

<sup>1</sup> Mi3dp, Córdoba, Argentina

<sup>2</sup> Mi3dp, Córdoba, Argentina

[jfernandez@sysinmotion.com.ar](mailto:jfernandez@sysinmotion.com.ar), [rivmarmar@gmail.com](mailto:rivmarmar@gmail.com)

## Resumen

Los Circuitos Creativos constituyen una invitación a explorar junto a nuestras infancias y juventudes, un modo de enseñar y aprender sobre y con tecnologías. Es un enfoque que se viene desarrollando y poniendo a prueba desde el año 2016 a partir de acuerdos de articulación con diferentes instituciones públicas como el Ministerio de Ciencia y Tecnología, la Universidad Nacional de Córdoba y la Municipalidad. En estas Jornadas acercamos una descripción de la experiencia de enseñanza y aprendizaje de temas vinculados a las Ciencias de la Computación transitada en las Tecnotecas de los Parques Educativos Noroeste y Sur de la capital cordobesa junto a los chicos y chicas participantes durante el período 2022-2023. Estos espacios educativos se encuentran territorialmente ubicados en puntos clave de la ciudad en los cuales persiste como problema socioeducativo la brecha digital tanto de acceso, como de uso y apropiación.

**Palabras clave:** Circuitos Creativos; Apropiación tecnológica; Ciencias de la Computación; Tecnotecas CREA; Brecha digital.

## 1. Introducción

Desde el primer homo al prosumidor/a y ciudadano/a digital de nuestro tiempo, ¿es posible pensar al ser humano sin vincularlo a sus tecnologías? ¿Y desconocer la impronta que las tecnologías tienen sobre lo que somos y sobre nuestras elecciones a futuro? Las tecnologías no sólo desempeñan un papel central en todos los procesos sociales, constituyen una dimensión de la condición humana [1]. Desde la perspectiva sociotécnica que sostienen estos autores, las tecnologías son construcciones sociales tanto como las sociedades son construcciones tecnológicas (p. 19) por lo que es preciso reconocer que no son universales, ni autónomas, ni evolutivas, ni herramientas con neutralidad socio política. Contrariamente, al ser eminentemente sociales, políticas, situadas en el marco de un territorio y de una cultura, impactan en lo que somos y en cómo nos proyectamos [2]. Posicionamientos como estos en clave local, nos interpelan a preguntarnos: ¿podemos tolerar que las infancias y juventudes cordobesas aprendan sobre tecnologías en circunstancias no exentas de profundas desigualdades? ¿por qué no decidir actuar?

El trabajo presentado en este artículo refleja parte de una travesía que desde 2016 nos empuja a la búsqueda y experimentación a partir de gestionar acuerdos de articulación con diferentes instituciones públicas como el Área de Divulgación del Ministerio de Ciencia y Tecnología (Mincyt)<sup>1</sup>, la Secretaría de Extensión de la Universidad Nacional de Córdoba (UNC), formando parte desde ese entonces del Banco de Organizaciones asociadas<sup>2</sup> y la Dirección de Parques Educativos de la Municipalidad de Córdoba. Una búsqueda que no se conforma con lo especulativo sino que pretende articular la reflexión con la acción, con la intervención en territorio de la mano de tantas chicas y chicos. Por eso, optamos por mostrar y explicar la experiencia de los Circuitos Creativos (CC) a partir de 2022 cuando se inauguró la primera de las Tecnotecas CREA (Centro de Recursos para la Experimentación y la Acción).<sup>3</sup> En primer lugar, es importante considerar que los PE son espacios comunitarios que nos convocaron a desarrollar nuestra propuesta dado que se encuentran territorialmente ubicados en puntos clave de la ciudad en los cuales persiste como problema socioeducativo la brecha digital tanto de acceso, como de uso y apropiación. La primera se refiere a la posibilidad de contar con dispositivos digitales como la computadora, la telefonía celular y las conexiones a internet. La segunda nos problematiza sobre las pragmáticas que van desde el ocio y el entretenimiento al aprendizaje, la comprensión y la participación cívica.

<sup>1</sup> Actualmente considerada Secretaría al ser absorbida por el Ministerio de Producción Ciencia e Innovación Tecnológica..

<sup>2</sup> El Banco de Organizaciones Sociales fue creado en 2017 y posee un mapa donde se encuentran georreferenciadas todas las organizaciones que trabajan en conjunto con la UNC en proyectos de extensión: <https://www.google.com/maps/d/u/2/viewer?mid=1LTdbDMDCYHxPDWJiy9FZS1vSJ4TFIK0x&ll=-31.410242545213798%2C-64.17788049311523&z=13>

<sup>3</sup> Actualmente son tres las Tecnotecas inauguradas y están situadas en PE Sur, PE Noroeste y PE Este.

Al respecto, Benítez Larghi observa un desplazamiento de la preocupación por los accesos a la preocupación por los usos y apropiaciones de los artefactos digitales [3]. Fundamentalmente nos interesa la tercera, la brecha de apropiación, es decir qué hacen las chicas y chicos que participan de la experiencia con/frente/a través de algunas tecnologías emergentes y las mediaciones que constituyen los procesos de apropiación en lo relativo a un recorte de la cultura vinculado a ellas. Inquietudes profesionales que cobran nuevos sentidos desde la presentación en 2019 del Programa Nacional de Ciencia y Tecnología y el proceso de gestación de la Propuesta Curricular de un nuevo campo para la escolaridad obligatoria, el de las Ciencias de la Computación que en este año cuenta con una propuesta curricular. Hoy, el sentido educativo de los CC es armar otro lazo con la tecnología que si bien parte de la posibilidad de interactuar con una diversidad de dispositivos, avance en su apropiación significativa en vistas a un horizonte de participación cívica. Entonces, la construcción de la presente narrativa de la experiencia pedagógica vivenciada tiene como propósito sistematizarla, documentarla, compartirla y hacerla pública como plataforma para la conversación entre colegas, el intercambio y la apertura de nuevos interrogantes.

## 2. Descripción de la Experiencia

### 2.1. Propuesta Didáctica

En cada espacio de encuentro nos sentimos, una y otra vez, provocados y convocados por los desafíos de “educar en tecnologías” y “educar con tecnologías”. Pero, ¿qué se enseña y qué se aprende en los CC?

- Experimentación de tareas colaborativas y creativas en red con pares y artefactos físicos y digitales para la creación de soluciones a problemas del entorno.
- Reconocimiento de diversos dispositivos computacionales utilizados en impresión 3D, inteligencia artificial (IA), modelado 3d, realidad virtual y realidad aumentada. Características, funciones, hitos en su evolución histórica e impacto en la vida cotidiana.
- Reconocimiento de los dispositivos computacionales como sistemas integrados por hardware y software. Software de diseño 3D. Software de IA.
- Navegación y aporte a repositorios de comunidades de diseño.
- Reconocimiento de un algoritmo como una secuencia ordenada de pasos para la resolución de una situación problemática y la ejecución de una tarea determinada.
- Ejecución de acciones de configuración en los dispositivos.
- Reconocimiento y comparación de materiales de impresión 3D, características y propiedades.
- Reconocimiento, comunicación y desarrollo de soluciones para problemas computacionales.

Consideramos necesario advertir que la experiencia es anterior a la publicación de la Propuesta Curricular para la inclusión de las Ciencias de la Computación en la educación obligatoria realizada el presente año. Sin embargo encontramos que los contenidos seleccionados para ser abordados en este período se vinculan a algunas de sus áreas y ejes.

### 2.2. Contexto

Desde 2017 venimos realizando un trabajo sostenido en articulación con los Parques Educativos de la Municipalidad de Córdoba. Se trata de espacios comunitarios que dependen de la Secretaría de Educación Municipal y ofrecen una amplia variedad de actividades deportivas, culturales, sanitarias, artísticas, tecnológicas, etc. de modo gratuito para las y los vecinos de todas las edades [4]. Actualmente hay cinco sedes en diferentes puntos de la ciudad, tal como se muestran en la Figura 1. Dos de ellos se encuentran situados dentro del anillo de circunvalación: PE Norte en Bº Marqués Anexo y PE Este en la intersección de los barrios Müller, Maldonado y Campo de la Ribera. Los tres restantes están emplazados fuera del anillo: PE Sur en Bº Congreso, PE Noroeste en Bº Villa Allende Parque y PE Sureste en Bº Avellaneda. Estos espacios educativos se encuentran territorialmente ubicados en zonas periféricas, relegadas y deprimidas de la ciudad que presentan altos niveles de vulnerabilidad social.<sup>4</sup> Geolocalización que resulta muy significativa considerando los datos arrojados por el Censo Nacional de Población, Hogares y Viviendas 2022 que advierten que en la Capital se concentra el 39% de la población cordobesa.

<sup>4</sup> Puricelli, M., & Rodríguez Saá, P. (2017) nos advierten que en abril del 2013 nace el Programa Parques Educativos previo relevamiento de información proveniente de Censos Nacionales, Provinciales y Municipales, estadísticas socio-económicas, el barómetro de la deuda social de la infancia, entre otros, para identificar zonas en la ciudad con déficits socio-educativos según: nivel de alfabetización, acceso a computadora, acceso a internet en el hogar, desarrollo de actividades físicas o participación en eventos culturales.

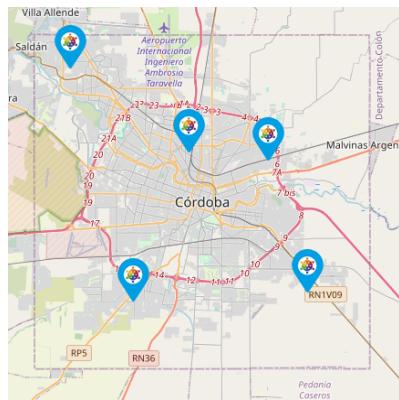


Figura 1. Ubicación de los Parques Educativos

Caminar estos territorios nos lleva a preguntarnos en clave local en torno a lo que a escala mundial se puede visualizar como una tendencia al aumento en el acceso a dispositivos digitales e internet. Es decir, identificar esta tendencia más global implica también reconocer su no universalidad. Y en ese sentido, distribuida desigualmente en las diferentes regiones y países del globo. Según datos de la CEPAL (2020), en 2019 el 66,7% de la población de América Latina y el Caribe era usuaria de Internet. Porcentaje que, si solo consideramos las infancias y juventudes, se reduce aún más.

A escala nacional y escolar, “Entre los años 2013 y 2021 se incrementó el porcentaje de la matrícula que asiste a escuelas con conectividad, pasando del 71,6 al 89,9%. Es decir que, en 2021, 9 de cada 10 estudiantes asistían a escuelas con conectividad, en su mayoría, provista por el estado (53,3%). Un 13,2% de la matrícula que asiste a escuelas estatales y un 31% de la que asiste a escuelas rurales no cuenta con conectividad en 2021” (p.9) [6].

A escala provincial, según el Censo Nacional de Población, Hogares y Viviendas llevado adelante por el Instituto Nacional de Estadística y Censos (INDEC) en el año 2022, el 91,2% de los cordobeses tienen celular, de los cuales solo el 6,8% no cuenta con servicio de internet en el hogar [5]. Sin embargo, nos parece interesante cruzar ese dato con otros referidos a nuestras infancias a escala municipal aún considerando que los destinatarios de la propuesta se incluyen en un rango más amplio de edad y asisten a escuelas de gestión y dependencia diversa. Según el Informe 2021 de las Condiciones socioeducativas de estudiantes del Nivel Inicial y Primario de los establecimientos educativos dependientes de la Municipalidad de la Ciudad de Córdoba en el nivel primario un 53,1% de los niños y niñas poseen dispositivos electrónicos. Si analizamos la información en términos del acceso o no a señal de internet, vemos que tienen conectividad aproximadamente un 36,9% de los/as estudiantes. Es decir que, si bien más de la mitad de ellos/as cuentan con dispositivos para conectarse, sólo una parte efectivamente accede a la red [7].

Hasta aquí hemos intentado historizar brevemente y caracterizar el contexto de realización de la propuesta para avanzar en el siguiente apartado en la experiencia propiamente dicha.

### 2.3. La Experiencia

Esta experiencia es parte de un proceso de construcción de un abordaje del enseñar y el aprender sobre y con tecnologías que como dijimos ya lleva ocho años. Dentro del campo de las investigaciones sobre la relación educación y tecnologías, resultan desde el comienzo de especial interés para el equipo, las que refieren a su potencial como exploratorias. Desde las posiciones más radicales como el experimento Hole in the Wall (El Agujero en la Pared) y otros de educación mínimamente invasiva en la India que llevó adelante Sugata Mitra a partir de 1999; los Fab Lab<sup>5</sup> y el movimiento Maker [8] conformado por múltiples influencias para impulsar una nueva cultura de apropiación social de la tecnología [9], entre otros que podríamos mencionar, nos interpelaron sobre cómo acercar diversas tecnologías digitales a niñas, niños y jóvenes desde un horizonte de igualdad educativa. Desde entonces y considerando el paradigma de aprendizaje de interacción entre el humano y la máquina pretendemos generar una simbiosis entre inteligencia y herramientas externas, sin las cuales la mente individual contaría sólo con sus propios medios y no funcionaría igual. Simari [10] habla de la computación particularmente como una tecnología que amplía nuestra capacidad de aprender porque permite hacer operaciones que son humanamente imposibles. Entonces, son las potencialidades y limitaciones de las tecnologías y su mediación en procesos educativos las que sostienen el carácter exploratorio que estamos convencidos, debe asumir la enseñanza cuando se concibe comprometida con el derecho a aprender de las infancias. En este campo de inquietudes profesionales comienzan a gestarse los CC como una invitación a aprender materializando ideas [11], un abordaje lúdico, desafiante y mediado por tecnologías que promueve procesos de aproximación a las tecnologías emergentes movilizados por la resolución de problemas de entorno.

Durante 2017, comenzaron los primeros contactos con autoridades de la Secretaría de Educación Municipal y se inició un ciclo de talleres en el PE Sur que albergó efectivamente a más de 200 participantes en edad de escolaridad primaria. Estos encuentros nos acercaron aprendizajes y nuevas inquietudes como la diversidad de modos en que los chicos y las chicas

<sup>5</sup> La expresión comienza a usarse en 2001. Proviene del inglés Fabrication Laboratory y se refiere a un taller de fabricación digital.

asumen los desafíos con saberes en una gran gama de niveles de apropiación, diversos modos de enfrentar creativamente la falta de recursos tecnológicos, la importancia de seguir avanzando en la ideación de situaciones problemáticas en un espacio de intersección entre el campo de la Tecnología, las Ciencias Naturales y Sociales vinculadas a los entornos singulares de quienes participaban y muy especialmente la necesidad de multiplicar la propuesta para llegar a una población mayor en número. Frente a esto, surge la posibilidad de articular también con la Secretaría de Extensión de la UNC para formar grupos de estudiantes provenientes de diversas carreras. Así, durante 2018 y 2019 se ofrecieron CC que albergaron hasta 120 participantes en simultáneo.

A principios del 2021, movilizados como toda la comunidad por las preocupaciones sociales, culturales, sanitarias y educativas en torno al impacto de la pandemia COVID19 se desarrollaron los primeros contactos con las nuevas autoridades de la Dirección de Parques a quienes presentamos los CC, sus fundamentos teóricos-metodológicos y la historia de su desarrollo. Desde entonces nuestra propuesta es considerada un “Proyecto Aliado” y comenzamos a recorrer sistemáticamente los diferentes PE desde Villa Allende a Congreso, desde Parque de la Ribera a Avellaneda. Parte de las actividades y registros fueron compartidos en el stand institucional asignado a Parques Educativos en el V Congreso Municipal de Educación y II Congreso Iberoamericano de Gobiernos Locales y Educación, bajo el lema “Pasión por Aprender” realizado al año siguiente.

En mayo de 2022 se inaugura la primera Tecnoteca CREA en PE Noroeste. Posteriormente la de PE Sur. A partir de allí se nos abrieron las puertas de estos espacios de alta disposición tecnológica, que cuentan con kits de robótica, impresoras 3D, equipos de sonido, smart, carro tecnológico, consolas de videojuegos, entre otros recursos. Antes de ello, llevar adelante nuestros CC implicaba una fuerte logística de recursos materiales para crear espacios estimulantes, innovadores e itinerantes en lo que teníamos que adaptarnos a los medios y la conectividad que hubiera en las instalaciones que nos brindaban. Algunos de sus resultados se convidaron a través de una ponencia en el VI Congreso Municipal de Educación y III Congreso Iberoamericano de Gobiernos Locales, III Seminario transformando ciudades.

En vistas a reconstruir la trama que implica la experiencia transitada entre 2022 y 2023, intentaremos anudar lo acontecido en las Tecnotecas de los PENO y PES, la descripción de la propuesta llevada adelante y sus protagonistas.

En primer lugar, resulta importante resaltar que sus destinatarios tienen entre 8 y 14 años y habitan en zonas aledañas a los PE tal como las consignadas en la Tabla 1.

Tabla 1. Población y zona de influencia. Fuente: Informe Situado. Agosto 2024. Parques Educativos de la Ciudad de Córdoba

Parque Educativo	Población menor a 14 años	Barrios que conforman su zona de influencia
Noroeste	21.325	Villa Allende Parque, Villa Cornú, Arguello, Cerro Norte, Saldán, Villa Rivera Indarte, 16 de noviembre, Villa Allende, Villa 9 de julio.
Sur	23.037	Congreso, San Pedro Nolasco, Residencial Sud, Santa Isabel 1ra, 2da y 3ra sección, Villa El Libertador, Los Olmos Sud, Cabo Farina, Mirizi, Santa Rosa Residencial, Carbo, Comercial, El Cabildo, La Esperanza, Parque Futura, Kairos, San Luis de Francia, V.I.C.O.R., Ampliación V.I.C.O.R., Santa Isabel Anexo, Colina los Pinos, Sacchi y Carrara.

Algunos CC se desarrollaron en período escolar, otros en tiempos de receso de invierno e incluso durante vacaciones de verano. El proceso de inscripción estuvo a cargo del equipo de gestión de cada sede realizando convocatorias abiertas a la comunidad y a escuelas primarias municipales. Cada CC tenía un cupo de 25 participantes. Asumimos que cada uno de ellos acerca los saberes construidos en el marco de su trayectoria escolar y su vida cotidiana, sin que sea un prerequisito para la inscripción que posean conocimientos previos acerca de los temas que se abordarán. Los estudiantes tuvieron la oportunidad de participar en hasta cinco encuentros, cada uno de los cuales se extendió por el término de dos horas, con proyectos que crecieron en complejidad durante el proceso.

A continuación, apostamos a reconstruir la propuesta a través de algunos de sus **principios** vertebradores:

**Primero la pregunta.** Es un problema ligado al contexto y considerado prioritario para la comunidad, el disparador. Genuino, en la medida que no parece obvia la solución. Factible de ser abordado tanto desde saberes adquiridos en su vida cotidiana y en la escuela, como desde su inventiva sumando incluso elementos de ciencia ficción. Desafiante, porque aún contando con saberes desde los cuales comenzar a pensar el problema, resultan insuficientes para estimular la búsqueda.

**Aprender materializando ideas.** La dimensión de lo material resulta un elemento clave. La potencialidad de esta dimensión, ha sido ampliamente explorada por el movimiento Maker y especialmente por aquel denominado “los creadores” que promueve el desarrollo de los textiles electrónicos como un tipo de actividad híbrida de creación que combinan lo físico/material y lo digital como dos modalidades distintas de aprendizaje, con potencialidad para revelar

cómo se fabrican los medios digitales cómo funcionan los ordenadores y la electrónica [12]. En el marco de los CC, se utilizan scrap de impresión 3D como materiales manipulables que se ponen a disposición en las mesas de trabajo para que los chicos y chicas puedan imaginar, ensayar, observar, construir, representar, mostrar, hacer visibles sus pensamientos y aprendizajes durante el proceso mientras descubren cómo funcionan diversas tecnologías emergentes. Por ejemplo, cuando los grupos se detienen a observar las capas o la densidad de las piezas están descubriendo el modo en que funciona este tipo de tecnologías.

**Mostrar no sólo explicar.** Anudado al principio desarrollado anteriormente, las conversaciones se apoyan en objetos para compartir los progresos individuales y/o grupales. Esto supone además de la construcción de un pitch, un ejercicio postural, de manejo del tiempo y de escucha activa valorando el feedback honesto.

**Hacer algo que se quiere hacer, que se necesite hacer y que es valioso hacer colaborativamente.** Es el carácter vivencial de estas actividades el que resulta prometedor para generar espacios de construcción de lo que Haste [13] denomina, agencia y responsabilidad. Nos referimos a reconocer el problema, elegir entre tolerarlo y/o resolverlo, hacer frente al problema, generar redes interpersonales y reflexionar sobre obligaciones y responsabilidades para avanzar en la apropiación de saberes y prácticas consideradas valiosas y al mismo tiempo, crear soluciones a problemas de entorno.

**5 Estaciones.** Cada CC está conformado por cinco Estaciones de trabajo por lo que la organización del espacio del salón, la disposición de materiales y la distribución del mobiliario resultan elementos fundamentales. Las denominamos: Estación 1:Visualiza y arma tu idea”, Estación 2: Construye y descubre como funciona, Estación 3: Crea, Estación 4: Diseña y modela tu idea, Estación 5: Materializa y se pueden observar en el Gráfico 1 que se ofrece a continuación.



Gráfico 1. Estaciones de trabajo de los Circuitos Creativos

**Ingreso aleatorio.** Al comenzar el grupo total se organiza en 5 subgrupos independientemente del género y saberes en torno a las tecnologías digitales. El punto de partida es aleatorio, es decir que cada subgrupo comienza en una estación diferente, y por consecuencia poseen distintos inicios en su camino a la solución del problema.

**Itinerario circular.** Refiere al curso de acción propuesto y la regulación del tiempo. Cada subgrupo asume el desafío de una estación hasta que cumplido el tiempo pasa a la estación siguiente.

**Iteración.** Encuentro tras encuentro poder tener la experiencia de un inicio distinto promueve procesos de ingeniería inversa. Nos referimos a construir y deconstruir creaciones.

**Exploración, descubrimiento y creatividad.** El itinerario circular propone un curso de exploración, un fluir de personas y conocimientos de los que encontramos indicios en sus ideas provisorias, producciones parciales y sus acuerdos alcanzados. Las expresiones del fluir tienen que ver sustantivamente con el conocimiento inacabado con los desafíos cognitivos que implica pensar problemas genuinos que no tienen soluciones algorítmicas sino procesos de descubrimiento que se plasman en la reflexión personal y su creatividad [14].

**Movilizar haciendo actuar y hacer actuar movilizando.** Para establecer un puente que posibilite la aproximación a las tecnologías tendiente a una apropiación más equitativa entre quienes participan de esquemas de acceso heterogéneos y desiguales (Benitez Larghi, 2014) elegimos el taller como modalidad que posibilita tensionar permanentemente, acción y reflexión junto a otros.

**Entornos estimulantes.** Cada CC constituye también un escenario y se idea en clave de inmersión sensorial con el propósito de potenciar el compromiso con la experiencia. Consideramos como afirma Rogoff [15] que todo organismo es inseparable de su entorno o en otras palabras, el entorno y el individuo están mutuamente implicados. Cualquiera sea el espacio que nos aloje, Centros Vecinales, Playones Municipales, Clubes o los Parques Educativos, la propuesta se piensa en términos de la persona más el entorno [16] o podríamos decir, los grupos más el entorno. En este sentido, las

Tecnotecas nos ofrecieron espacios de alta disposición tecnológica, localizados en salas amplias, luminosas, coloridas e intervenidas desde el diseño, climatizadas y con un mobiliario modular.

**Elementos lúdicos y agilidad.** Considerando los elementos lúdicos que identifica Fernández Solo de Zaldívar [17] podemos afirmar que se incorporan a la propuesta, un desafío por estación de trabajo y un tiempo acotado para resolverlo como mecánicas que combinadas nos permiten crear una dinámica de juego dado que los desafíos se configuran para darle sentido a la búsqueda de la solución del problema de entorno. Por otra parte, la disposición circular del mobiliario, los colores de sus manteles, los números de las estaciones, la bomba que regula el tiempo, la música de fondo y los coloridos materiales manipulables conforman una estética que colabora en la comprensión del itinerario. Elementos que se articulan a la agilidad dada a partir de la observación y reflexión grupal sobre lo acontecido.

En síntesis, cada uno de los participantes y grupos, al transitar por el CC, debe haber podido experimentar de qué se trata la/s tecnología/s que constituyen el objeto de estudio a partir de una experiencia de trabajo colaborativo mediada por tecnologías en torno a la resolución de un problema concreto.

### 3. Resultados

Entendemos que es necesario recuperar lo vivido en aquel período, resignificando y reconstruyendo datos cuantitativos y cualitativos que den cuenta de, como en todo proceso de conocimiento, de algunos hallazgos siempre provisорios en torno a los aprendizajes logrados, las fortalezas y debilidades identificadas en el proceso. Podrán observar que en algunas ocasiones referiremos a diversos proyectos y en otras tomaremos uno, por ejemplo Eco Titanes, como hilo de desarrollo de alguna reflexión.

#### Participantes y proyectos

Tecnoteca/ Parque Educativo	Cantidad de inscriptos/as	Cantidad que finalizaron	Cantidad de proyectos	Ejemplos
Sur	210	198	48	Escombrus: Eco Titán ideado por dos hermanos con la habilidad para detectar escombros como los observados en barrios aledaños a la Tecnoteca, recolectarlos y reutilizarlos para elaborar bancos para plazas y ladrillos que se ponen a disposición de las y los vecinos.
Noroeste	295	278	76	Rex: Eco Titán creado por Joaquín con la habilidad de drenar, triturar, compactar en cubos y depositar en una cinta transportadora los residuos extraídos del lago San Roque. Funcionaría con energía solar.

Estos datos nos permiten contextualizar casos como el de Ciro (PES) y Julián (PENO), quienes participaron en más de una edición de la propuesta sorprendiendo a sus adultos de referencia y haciéndonos repensar la posibilidad de plantear la propuesta en niveles.

**Apropiación progresiva de saberes.** Eco Titanes guardianes del ambiente dio lugar a la imaginación de múltiples dispositivos para reducir, reparar, reutilizar y reciclar diversos residuos que preocupan en la zona. 4R que operan como pilares de una Economía Circular que se promueve como política pública desde el Municipio. Para avanzar en la construcción de un prototipo de su idea de solución experimentaron la necesidad de trabajar en red con plataformas de diseño, software de realidad virtual para navegar las piezas y de realidad aumentada para ubicar los Eco Titanes en un territorio real. Algunos/as desde el río Suquía y los canales que recorren estas zonas hasta las sierras de Córdoba. Nos referimos a reconocer características, funciones y ejecutar acciones de configuración en los dispositivos de estas tecnologías para aportar a los procesos de pensamiento que el grupo llevaba adelante en su proceso creativo.



Imagen 1. Creaciones con residuos 3D.

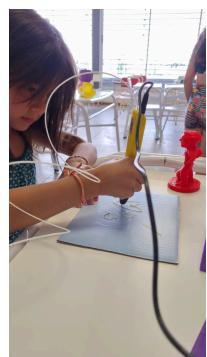


Imagen 2. Explorando lápices de impresión 3D.



Imagen 3. Bocetos de Eco Titanes.



Imagen 4: Del boceto al diseño en Tinkercad .



Imagen 5: Del boceto al diseño en Sculptris.



Imagen 6: Configurando la impresora.



Imagen 7: Navegando un diseño con realidad virtual.



Imagen 8: Explorando Realidad Aumentada.

**Trabajo en torno al error.** En el campo de la didáctica contemporánea se ha venido resignificando el papel del error. Se observa un corrimiento de la preocupación por su fijación, detección y corrección a la preocupación porque tanto docentes como estudiantes puedan comprender y atender a su origen [18]. En ese sentido, promovemos su abordaje de manera individual, grupal o colectiva en las instancias de feedback honesto. Uno de los CC, Picathon, nos permite asomarnos a procesos como el de Santino, quien imaginó diversos dispositivos para apertura de algunas de las puertas de su casa y en el camino de desarrollo de la idea, entre imaginarios, bocetarlos, diseñarlos y materializarlos los puso a prueba una y otra vez para evaluar y ajustar tamaños, medidas específicas, funcionalidad, entre otros, encarando el error como algo significativo para alcanzar la solución al problema.

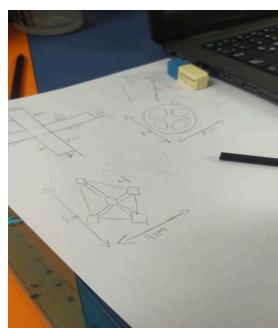


Imagen 9: Tres versiones de un mismo proyecto.

**Construcción en comunidad.** En Eco Titanes el grupo total descubrió en el encuentro final que sus creaciones podían ser complementarias. Esto nos permitió analizar cómo al igual que sus creaciones las tecnologías operan en red. Las tecnologías que crearon les permitieron iniciarse también en el reconocimiento de un algoritmo como una secuencia ordenada de pasos para la resolución de una situación problemática y la ejecución de una tarea determinada. Finalmente, en el grupo total se volvía a reunir para compartir el proceso transitado y el estado de desarrollo de la idea de solución, para recibir feedback del grupo las cuales incorporamos o no a nuestra idea de solución final. Estos espacios fueron vistos como instancias de recuperación de voces plurales, alternativas y polifónicas sobre lo que podemos crear juntos y en clave política.



Imagen 9: Creaciones varias.

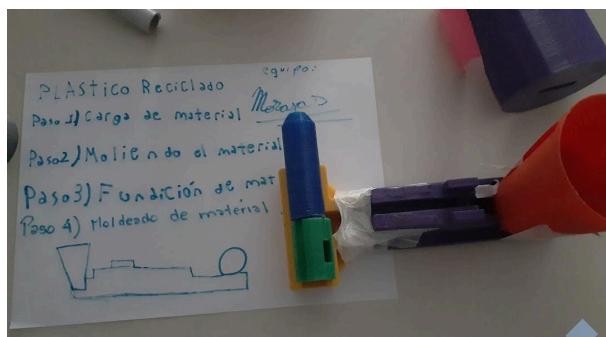


Imagen 10: Noción de algoritmo.



Imagen 11: Muestra no solo expliques.

**La enorme capacidad de aprendizaje.** Uno de nuestros argumentos clave es que los CC pueden ayudar a aumentar la participación activa de quienes aún por diversas razones no se hayan alfabetizados. Este es el caso de Ismael, un niño que armó una de las impresoras 3D. No estar alfabetizado no lo detuvo. Una muestra más de que todos podemos aprender. Hace falta “tomar al alumno como es”, justamente “para no dejarlo donde está” [19] Observábamos que en cada cambio de estación siempre retornaba a materializar. Su insistencia provocaba el sentido común. Hoy podemos decir que una de las primeras impresoras que llegaron a los parques fue armada por él. En otras intervenciones nos hemos encontrado con Ismael, esperamos que siga abierto a sorprenderse con lo que puede llegar a hacer y no deje de sorprendernos a nosotros. Esto ayuda a poner en cuestión algunas ideas cristalizadas que circulan acerca de las limitaciones de las infancias en contextos vulnerables.

**Estudiantes con conocimientos avanzados.** Podemos si les parece un poco abrir la caja negra de algunos de estos CC y rememorar que a partir de un alumno llamado Joaquín, el cual tenía conocimientos más avanzados sobre Impresión 3D, nos propone asumir un rol de ayuda a sus pares cuando se trataba de descubrir partes y funcionamientos de la misma, así surge la figura de “capitán”. No se fastidia, no se aburre. Encuentra un espacio para involucrarse desde otro lugar y sigue avanzando así en un trabajo intelectual ya que se desafía a sí mismo a saber para explicar mejor a quien todavía no. Modalidad esta que se multiplica en las ediciones posteriores.

**Tiempo de juego, placentero para el aprendizaje.** Una de las recurrencias es la participación activa sin cortes a lo largo de 2 horas de conexión con los desafíos propuestos.

**Agrupamientos.** El ingreso aleatorio en cada encuentro multiplicó las posibilidades de experimentar interacciones con pares de diferente género y edad y asumir tareas colaborativas.



Imagen 12: Encarando un proyecto grupal.



Imagen 13: Configurando la impresora 3D

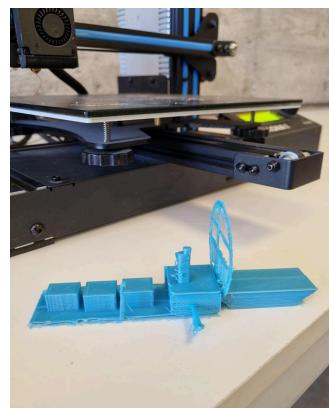


Imagen 14: Eco Titán Rex

**Las tecnologías, pretextos para otro texto.** Las tecnologías abordadas no siempre fueron las mismas. Así como en la génesis del enfoque en cada instancia de implementación nos problematiza la relevancia de incluir y el momento en que se hará cada una de las tecnologías emergentes a las que hemos hecho referencia.

## 4. Conclusiones

A partir de analizar algunos indicios de la experiencia transitada en la particular configuración de chicas y chicos, grupos, entornos, saberes, artefactos tecnológicos, objetos digitales y prácticas que van adquiriendo los CC presentados en este trabajo arribamos a las siguientes hallazgos:

- Todas las y los destinatarios independientemente de su barrio de procedencia, edad, género y saberes previos lograron conectarse con los desafíos propuestos a lo largo de dos horas sin interrupciones ni recreos.
- Aún en las estaciones donde los objetos tecnológicos físicos y/o digitales no están presentes el desafío resultó convocante.
- Los saberes y las prácticas que se adquieren al operar diversos dispositivos se fueron anudando al cómo y por qué los ponemos en funcionamiento acercándonos a algunos de los problemas que abordan los especialistas del área.
- Una incipiente agencia estudiantil se manifiesta en el mostrar y explicar los proyectos su sentido y posibles impactos sociales promoviendo actitudes de participación cívica.
- La iteración fogonea los niveles de apropiación de conceptos dándoles profundidad. Además, algo acontece en torno a la atención dado que en todos los casos entre la segunda y tercera iteración aunque se encuentran resolviendo diferentes desafíos se produce un cambio en el clima de la producción. Quizás esto sea un indicio de que las actividades mentales van tomando primacía por sobre la tarea aún tratándose de la elaboración de algo.

Como propuestas superadoras consideramos que a escala de lo pedagógico-didáctico resultaría interesante incorporar la visita de representantes del estado municipal, organizaciones sociales que trabajan en pos de una Economía Circular y especialistas del campo de las Ciencias de la Computación para que escuchen las propuestas de los grupos y se pueda generar un espacio de intercambio. Otra posibilidad es pensar en términos de CC por niveles y una propuesta específica para adolescentes y jóvenes. A escala académica nos interesa indagar en los procesos de aprendizaje y las condiciones de la enseñanza en el marco de los CC en y con la mediación de tecnologías, sin pretensión de develar verdades sino en la búsqueda de renovar preguntas a partir del análisis de casos como los tomados para esta ponencia.

## 5. Agradecimientos (opcional)

Nuestro agradecimiento especial a los creativos y creativas, sus familias, a todo el personal de maestranza, a la Dirección de Parques Educativos por creer que una política pública apostando a los procesos es posible. A nuestros colaboradores Juliana, Amilcar y a todos los que desde su lugar se suman a nuestro circuito de feedback.

## Referencias

- [1] Thomas, H., Santos, G., *Tecnologías para incluir*. Carapachay: Lenguaje Claro Editora, 2016.
- [2] Feenberg, A., *Questioning Technology*. London: New Fetter Lane, 1999.
- [3] [15] S. Benítez Larghi y C. Duck, “Capítulo 6: Uso y apropiación de tecnologías de la información y la comunicación. Una aproximación a su investigación en Argentina.” En: *Estudios sobre consumos culturales en la Argentina contemporánea*. De: Grillo, M.; Papalini, V. y Benítez Larghi, S. CLACSO-PISAC-Consejo de Decanos de Facultades de Ciencias Sociales y Humanas, 2014.
- [4] M. Puricelli, P. Rodríguez Saá. (2017). Parques educativos para la ciudad de Córdoba. TecYt, (3). [Online]. Available: <https://revistas.unc.edu.ar/index.php/tecyt/article/view/18798>
- [5] Instituto Nacional de Estadística y Censos. Censo Nacional de Población, Hogares y Vivienda. (2022) [Online]. Available: <https://censo.gob.ar/>
- [6] Montes, N., Nistal, M. y Orlicki, E. (2023). Conectividad y sistemas de gestión escolar. Observatorio de Argentinos por la Educación.
- [7] Subsecretaría de Coordinación Educativa Observatorio Educativo Municipal. Dirección General de Educación. Dirección de Aprendizaje y Desarrollo Profesional Dirección de Gestión Educativa. (2021, Mayo) [Online]. Available: <https://documentos.cordoba.gob.ar/MUNCBA/AreasGob/Edu/DOCS/Seguimos%20con%20vos%20aprendiendo%20en%20casa/Observatorio%20educativo%20municipal/Condiciones%20socioeducativas%20estudiantes%20del%20ni%20inicial%20primario/Documentos%20de%20trabajo/Informe%202021/informe-2021-1-1-1.pdf>
- [8] Dougherty, D., *The Maker Movement*, Innovations: Technology, Governance, Globalization, 7, (3), P.P. 11-14, 2012.
- [9] R. Tabarés Gutierrez, T. Bartolomé Campino, J. García Valbuena (2022, Julio). Oportunidades y retos de la cultura maker para la digitalización de la sociedad española. [Online]. Available: <https://doi.org/10.24197/st.2.2022.206-227>

- [10] G. R. Simari, “Los fundamentos computacionales como parte de las ciencias básicas en las terminales de la disciplina Informática” en VIII Congreso de Tecnología en Educación y Educación en Tecnología. Red de Universidades con Carreras en Informática (RedUNCI), julio 2013.
- [11] J. Fernández, M. Rivero, “Círculo Creativo. Aprender materializando ideas...Sí, ¿Pero cómo en nuestra comunidad?” en Experiencias y Tecnologías encaminadas a la investigación y educación. El camino hacia la formación con tecnología. ISBN 978-958-52015-2-1. Espinal Tolima: Instituto Tolimense de Formación Técnica Profesional ITFIP, Colombia/Universidad Tecnológica Nacional Facultad Regional La Rioja UTN La Rioja Argentina/Escuela Normal de Naucalpan de México, 2019.
- [12] L. Buechley, K. Peppler, M. Eisenberg y Y. Kafai. (2013, Septiembre). Textile Messages: Dispatches From the World of E-Textiles and Education (New Literacies and Digital Epistemologies) [Online]. Available: [https://www.researchgate.net/publication/262261332\\_Textile\\_Messages\\_Dispatches\\_From\\_the\\_World\\_of\\_E-Textiles\\_and\\_Education\\_New\\_Literacies\\_and\\_Digital\\_Epistemologies](https://www.researchgate.net/publication/262261332_Textile_Messages_Dispatches_From_the_World_of_E-Textiles_and_Education_New_Literacies_and_Digital_Epistemologies)
- [13] Haste, H., *Nueva Ciudadanía y educación. Identidad, cultura y participación*. Bs. As: Editorial Paidós, 2017.
- [14] [18] Litwin, E., *El oficio de enseñar. Condiciones y contextos*. Bs. As: Editorial Paidós, 2008.
- [15] Rogoff, B., “Capítulo 3: Aprendices de pensamiento.” En: *El desarrollo cognitivo en el contexto social*. Barcelona: Ed. Paidós, 1993.
- [16] Perkins, D., *La escuela inteligente*. España: Editorial Gedisa, 2010.
- [17] Fernández Solo de Zaldívar, I. (2015, May 29) “Juego serio: Gamificación y Aprendizaje”. En: Comunicación y Pedagogía no 321-322. Aprendizaje basado en Juegos. [https://www.researchgate.net/publication/312164454\\_de\\_Zaldivar\\_I\\_F\\_S\\_2015\\_Juego\\_serio\\_gamificacion\\_y\\_aprendizaje\\_Comunicacion\\_y\\_pedagogia\\_Nuevas\\_tecnologias\\_y\\_recursos\\_didacticos\\_281\\_43-48](https://www.researchgate.net/publication/312164454_de_Zaldivar_I_F_S_2015_Juego_serio_gamificacion_y_aprendizaje_Comunicacion_y_pedagogia_Nuevas_tecnologias_y_recursos_didacticos_281_43-48)[...] Organización de Estados Iberoamericanos. (2024, Agosto) [Online]. Available:<https://panorama.oei.org.ar/sugata-mitra-el-docente-debe-centrarse-en-hacer-las-preguntas-correctas/>
- [19] Meirieu, P., *Recuperar la Pedagogía*. Bs. As.: Editorial Paidós, 2016.

# El castor atareado como secuencia didáctica introductiva a las ciencias de la computación

Guillaume Hoffmann<sup>1</sup>

<sup>1</sup>CONICET - Universidad Nacional de Córdoba, Argentina  
[guillaume.hoffmann@conicet.gov.ar](mailto:guillaume.hoffmann@conicet.gov.ar)

## Resumen

Presentamos una secuencia didáctica ubicada en el primer semestre de una carrera de ingeniería en telecomunicaciones. Se trata de una actividad desenchufada dada como primera parte de la primera materia de programación de la carrera. Esta secuencia consiste en una presentación informal de las máquinas de Turing, con ejercicios consistiendo en realizar la ejecución de ellas sobre papel. La ejecución se hace desde una cinta inicialmente vacía. A través de ejercicios de dificultad creciente y la presentación de ejemplos, la actividad da lugar a discutir problemas fundamentales de las ciencias de la computación.

**Palabras clave:** Máquinas de Turing, actividad desenchufada, castor atareado.

## 1. Introducción

En este artículo, presentamos una secuencia didáctica dada como primera clase de una materia de programación. Esta materia se encuentra en el primer semestre de una carrera de Ingeniería en Telecomunicaciones. La materia, llamada Programación I, tiene un plan centrado en el aprendizaje del lenguaje C, incluyendo hasta funciones y arreglos. Los temas más avanzados del lenguaje C, como punteros y gestión dinámica de la memoria, son parte de la materia siguiente Programación II.

La materia Programación I supone los tradicionales desafíos de la enseñanza de la programación en primer año de la universidad. Además del nivel de dificultad general de la tarea, y la falta de recursos didácticos actualizados para el público, se suma la desigualdad entre quienes ya practicaron la programación en la escuela secundaria, con los demás que no lo hicieron. Otro desafío de esta materia es el poco tiempo de clases por semana, solo 3 horas (4 horas cátedra), y los exámenes parciales y recuperatorios se programan dentro de ese tiempo, lo cual, agregando los días feriados no recuperados, han dado una cantidad de 10 a 13 clases teóricas/prácticas para el cursado según el año.

La actividad que presentamos en este artículo fue dada en el 2017 y 2018, hecha en la primera semana de cursado, en un tiempo de dos módulos de 1h30. Se trata de una actividad de ciencias de la computación desenchufada (CS unplugged), ya que se hace con solo papel y lapicera. La actividad consta de una parte teórica y una parte práctica con ejercicios que consisten en ejecutar máquinas de Turing paso a paso sobre papel, y explicar *por qué* algunas de ellas tienen una ejecución infinita.

Los objetivos de la actividad son tener una primera aproximación a las máquinas de Turing, medir el “tiempo de ejecución” de programas en cantidad de pasos y tener una aproximación a la complejidad de los programas. La complejidad acá se entiende como complejidad en prever el comportamiento de un programa. Finalmente, es una actividad disparadora, con mínimos requisitos, que apunta a generar interés y contexto con respecto a los contenidos que la materia ofrece a continuación: la programación en un lenguaje compilado.

## 2. Descripción de la Experiencia

### 2.1. Propuesta Didáctica

La idea de la propuesta es de presentar las máquinas de Turing de la manera más simple posible. El objetivo no es presentar las máquinas de Turing con respecto a algún procesamiento que hagan sobre su entrada, dado que se dan con cinta inicial vacía, es decir inicializada toda en ceros. Esta idea es relacionada con la función no computable del castor atareado (busy beaver) [5].

El castor atareado es una función que mide el máximo número de pasos que una máquina de Turing puede realizar antes de detenerse, dado un número específico de estados. También llamamos castor atareado a la máquina que tiene ese tiempo de ejecución dentro de su clase. Este concepto se suele utilizar para ilustrar los límites de la computabilidad y la complejidad de ciertos problemas en informática. En efecto, la función del castor atareado es no computable, lo que significa que no existe un algoritmo que pueda calcular su valor para un número de estados arbitrario.

En el contexto de esta secuencia didáctica, no damos una definición de la función del castor atareado, tampoco presentamos el resultado de indecibilidad. Solo queremos lograr, a través de ejemplos concretos, inferir intuitivamente que el comportamiento de los programas es muy difícil de prever en general y que es cada vez más difícil de prever a medida que su tamaño, medido en cantidad de instrucciones, crece. Además, el tamaño de programa desde el cual esa complejidad se vuelve inmanejable, es mucho más pequeño que lo que una persona sin experiencia en ciencias de la computación podría suponer.

### 2.2. Contexto

La universidad de la experiencia es la Universidad Blas Pascal de Córdoba, Argentina. Es una institución de gestión privada que ofrece, entre otras, dos carreras de grado de ingeniería: informática y telecomunicaciones. La materia, Programación I, es del primer semestre de la carrera de Telecomunicaciones. La carrera tiene un total de 10 semestres (5 años). En los dos años que se dio la experiencia, participaron 13 alumnos en el 2017 y 8 alumnos en el 2018.

### 2.3. La Experiencia

La secuencia didáctica completa se compone de:

1. Una presentación teórica de las máquinas de Turing, desde una perspectiva histórica y su relación con los programas compilados. La presentación es informal en el sentido de que no se usa la tradicional definición de máquinas de Turing escrita con tuplas.
2. Una secuencia práctica con 4 ejercicios a resolver sobre papel, consistiendo en ejecutar máquinas de Turing con cinta inicial vacía.
  - a. En los primeros dos ejercicios, se escriben las configuraciones de las MT de manera “gráfica”, dibujando el cabezal de la máquina por debajo de la cinta y el estado interno de la máquina por debajo del cabezal.
  - b. Luego de estos dos ejercicios, se presenta la escritura compacta, que consiste en colocar la misma información en una sola línea.
3. Una corta conclusión teórica en torno a la indecibilidad de prever la cantidad de pasos y el carácter terminante o no de una máquina de Turing.
4. Una evaluación formativa por cuestionario web que fue evaluado en el aula, de 11 preguntas de opción múltiple.

La secuencia completa se puede llevar a cabo en dos módulos de 1 hora ½ cada uno, o sea 3 horas en total.

El material de esta secuencia se puede encontrar en <https://cs.famaf.unc.edu.ar/~hoffmann/turing/>.

## 3. Resultados

### 3.1 Resultados cualitativos

Al proponer esta actividad, hemos relevado los siguientes resultados:

- **Motivación:** la actividad fue motivadora dada que todos los estudiantes participaron de los ejercicios de ejecución de máquinas de Turing sobre papel. Además, todos los presentes contestaron al cuestionario formativo dado al final de la actividad.
- **Trabajo grupal y colaboración:** se indicó que los ejercicios se podían hacer hasta en grupo de 3. En distintas oportunidades, los estudiantes pudieron comparar las ejecuciones y corregirse cuando encontraban discrepancias en sus respuestas.
- **Reacciones y debate:** luego de la realización de los ejercicios con máquinas con ejecución finita y máquinas con ejecución infinitas, presentamos casos de “máquinas campeonas en su categoría”. En particular, con el ejemplo de la máquina binaria con 5 estados pudiendo ejecutar más de 47 millones de pasos antes de terminar,

varios alumnos manifestaron incredulidad y/o curiosidad. Este ejemplo permitió introducir (sin demostrarlo) la indecibilidad del problema de la parada. Culminó con la presentación de la siguiente cita de Alan Turing, del año 1950 [6]:

“A mi juicio, el punto de vista de que las máquinas no pueden sorprender obedece a la falacia a la que se encuentran particularmente sujetos los filósofos y los matemáticos: la suposición de que tan pronto se presenta un hecho a la mente, todas las consecuencias de ese hecho irrumpirán simultáneamente en la mente junto con él. Esta suposición resulta de gran utilidad en muchas circunstancias, pero solemos olvidar con demasiada facilidad que es falsa. Una consecuencia natural es suponer que no hay virtud en el mero cálculo de las consecuencias a partir de datos y principios generales.”

De esta secuencia didáctica, la idea es inferir que en general, un programa informático que se presenta en su forma ejecutable, es por defecto no confiable. Lo que puede otorgarle confiabilidad es una explicación clara de su comportamiento. Por otro lado, al escribir un programa, usualmente se elabora un código fuente de alto nivel, por ejemplo usando el lenguaje C, que luego se convierte en instrucciones. Esta aproximación resulta más confiable, aunque se abre ahora la cuestión de la confiabilidad del compilador. Este es tan solo uno de los varios debates que pueden surgir de esta actividad según las preguntas del alumnado y el tiempo disponible.

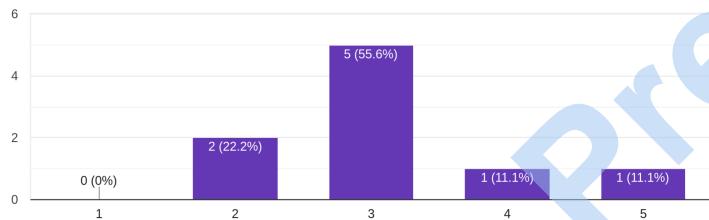
En contraste con la aproximación positivista de la ingeniería, la perspectiva de las ciencias de la computación es más pesimista y defensiva, y arroja una luz distinta sobre los productos informáticos que se venden a diario.

### 3.2 Encuesta final de la materia Programación 1

En el 2017, se procedió a una encuesta final de la materia, y fue contestada por 9 estudiantes. A continuación, los resultados.

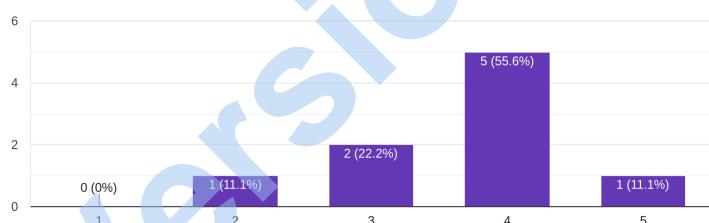
Me gustó la parte 1 de la materia: máquinas de Turing, manejo de Linux

9 respuestas



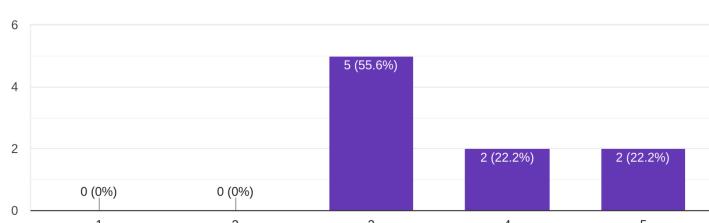
Me gustó la parte 2: primeros programas en C

9 respuestas



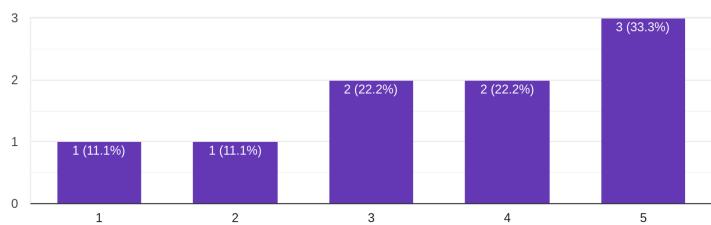
Me gustó la parte 3: estructuras de repetición

9 respuestas



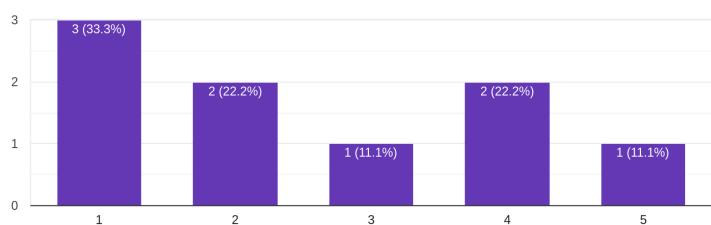
Me gustó la parte 4: arreglos

9 respuestas



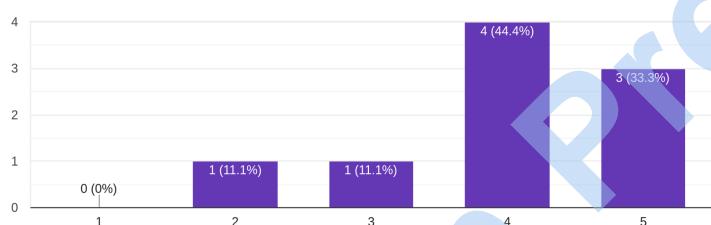
Me gustaron los proyectos de arreglos

9 respuestas



Me gustó la parte 5: funciones

9 respuestas



La mayoría de los encuestados dio una valoración promedio (3 puntos en una escala de 1 a 5) a la parte 1 de la materia, que contiene al mismo tiempo las máquinas de Turing y el manejo de los comandos Linux.

## 5. Conclusiones

Esta secuencia didáctica se dio dos veces, provocando el interés de los estudiantes en las actividades propuestas y en las consideraciones teóricas que se presentaron. El uso del tiempo (dos módulos de 1h30 reloj) fue adecuado y la actividad resulta manejable por el docente. La cooperación y retroalimentación entre estudiantes deja suponer que esta actividad puede escalar a grupos de alumnos más grandes sin problemas.

Como posibilidad de ampliar la actividad, se exploró la posibilidad de agregar ejercicios que involucren únicamente máquinas no borrrantes [4]; esas máquinas son más fácil de ejecutar sobre papel dado que lo escrito no se borra más.

Un punto para mejorar de esta secuencia, en el contexto de la materia Programación I, fue que no se retomó el tema más adelante en la materia. Sería interesante establecer comparaciones entre programas “escritos” en máquina de Turing y programas en C. Por ejemplo, un programa C es esencialmente secuencial y eventualmente puede contener bucles; en cambio una máquina de Turing es esencialmente un proceso repetitivo que eventualmente puede terminar.

Finalmente, mencionamos trabajos anteriores relacionados. En [2,3], la idea de usar el castor atareado como introducción surge de un artículo del 1995, pero con el lenguaje de programación WHILE, que es una alternativa interesante para esta actividad. El lenguaje WHILE vendría a ser una versión simplificada de un lenguaje imperativo como C, donde la única estructura condicional o de bucle es un bucle “WHILE  $X \neq 0$ ”. La idea de usar WHILE es llegar a la prueba de que la función busy beaver no es computable. Es un objetivo más ambicioso que en nuestra actividad pero la elección de WHILE lo hace más fácil.

Una experiencia más reciente de una actividad desconectada con máquinas de Turing en la escuela secundaria [1] nos advierte que esta secuencia didáctica podría ser probada también en el ciclo secundario. En [1], la experiencia también consiste en ejecutar máquinas de Turing sobre papel, pero se trata de ejecutarlas con datos de entrada. El objetivo de la actividad es que los estudiantes descubran y expliquen qué hacen las máquinas al ejecutarlas. Los autores reportan que su secuencia didáctica fue recibida positivamente, lo que podría dar lugar a nuevos desarrollos en línea con las máquinas de Turing.

## Referencias

- [1] Rodríguez, J., Parra, G., Gili, G., Parra, S., Dolz, D., & Roumec, H. (2023). Una máquina de turing en la escuela. In *XXVIII Congreso Argentino de Ciencias de la Computación (CACIC)*.
- [2] Morales, R., Ramos, G., Vico, F. J., & Triguero, F. (2003). Una alternativa docente a la Máquina de Turing.
- [3] Morales-Bueno, R. (1995). Technical opinion: noncomputability is easy to understand. *Communications of the ACM*, 38(8), 116-117.
- [4] Margenstern, M. (1995). Non-erasing Turing machines: a new frontier between a decidable halting problem and universality. In *LATIN'95: Theoretical Informatics: Second Latin American Symposium Valparaíso, Chile, April 3–7, 1995 Proceedings* 2 (pp. 386-397). Springer Berlin Heidelberg.
- [5] Aaronson, S. (1999). Who Can Name the Bigger Number?  
<https://www.scottaaronson.com/writings/bignumbers.pdf>
- [6] Turing, A. M. (1950). *Computing machinery and intelligence*.

# Talleres de acercamiento a la programación orientados a la escuela secundaria. Una experiencia desde un proyecto de extensión universitaria de la UNLP

Isabel Kimura<sup>1</sup>, Milagros Palacios<sup>1</sup>, Simón Pedro Mc Govern<sup>1</sup>, Candela Aylén Arias<sup>1</sup>, Claudia Queiruga<sup>1</sup>, Claudia Banchoff Tzancoff<sup>1</sup>

<sup>1</sup> LINTI (Laboratorio de Investigación en Nuevas Tecnologías Informáticas)-Facultad de Informática-UNLP, La Plata, Argentina  
ikimura@linti.unlp.edu.ar, milagrospalacios2000@gmail.com, simon.Mc1070@gmail.com, candelaarias167@gmail.com, {claudiaq, cbanchoff}@info.unlp.edu.ar

## Resumen

En este trabajo se describe y analiza una experiencia docente enmarcada en las actividades del proyecto de extensión de la UNLP “Extensión en vínculo con escuelas secundarias”. Esta experiencia consiste en la implementación de dos talleres que introducen conceptos claves de programación. Los mismos se orientaron a estudiantes y docentes del ciclo superior de tres escuelas secundarias de la localidad de La Plata y se desarrollaron de manera simultánea en el tiempo y en espacios áulicos separados dado que persiguieron objetivos diferentes. El taller para estudiantes se orientó a acercar conceptos como secuencia de instrucciones, estructuras de control y procedimientos, y creación de apps, mediante el uso de las herramientas de programación en bloques Pilas Bloques y MIT APP Inventor. El taller de docentes se orientó a discutir estrategias de introducción de temas de programación. Participaron del taller para estudiantes entre 18 y 30 estudiantes, y del taller para docentes entre 8 y 10 docentes.

**Palabras clave:** Programación; Extensión universitaria; Escuela secundaria; Pilas Bloques, MIT App Inventor.

## 1. Introducción

En el presente trabajo se describe y analiza una experiencia de trabajo con estudiantes y docentes, del ciclo superior, de tres escuelas secundarias, de gestión pública, ubicadas en la localidad de La Plata. Esta experiencia se enmarca en las actividades desarrolladas en el proyecto de extensión universitaria “Extensión en vínculo con escuelas secundarias”<sup>1</sup> de la Universidad Nacional de La Plata (UNLP), cuyo objetivo es generar interacciones con escuelas secundarias del área de influencia de la UNLP para trabajar en el acercamiento de la Informática como campo disciplinario. Al interior del proyecto se discute la Informática como campo de conocimiento en la escuela secundaria, entendida ésta como una disciplina fuertemente vinculada al desarrollo socio-productivo, al campo laboral y la formación ciudadana. Es en este sentido que desde el equipo de trabajo nos preguntamos: ¿qué contenidos de la Informática son relevantes en la escuela secundaria?, ¿cómo acercar la Informática al ámbito escolar?, ¿con qué materiales?, ¿con qué didáctica? y fundamentalmente ¿para qué? [1]. Las escuelas con las que se trabaja desde el proyecto son de gestión pública y están localizadas en la provincia de Buenos Aires, específicamente en las localidades de La Plata, Berisso, Ensenada y Magdalena. En este sentido, es necesario atender la diversidad de modalidades ofrecidas en esta jurisdicción y así cómo también la diversidad de enfoques adoptados por las escuelas en el proceso de incorporación de saberes de Informática, en algunos casos, inclusive, confundiéndose con el concepto de “informática de usuario” de Diego Levis [2]. En relación a la pregunta sobre los contenidos de la Informática considerados relevantes, la programación aparece claramente como uno de los más difundidos y quizás el más representativo del campo disciplinario, aunque no es el único contenido que se trabaja desde el proyecto. El enfoque de la enseñanza de la programación propuesto no se relaciona a la

<sup>1</sup> “Extensión en vínculo con escuelas secundarias” forma parte del programa de extensión de la UNLP “Educación para la inclusión” (<https://www.extension.info.unlp.edu.ar/programa-de-extension-unlp/>)

formación de programadores sino a la incorporación de competencias informáticas orientadas a comprender cómo es la construcción de artefactos digitales y colaborar en la formación de ciudadanos informados en una sociedad mediada por tecnologías digitales. La Figura 1 ilustra los temas seleccionados para enseñar programación en la escuela secundaria, enfoques didácticos, herramientas y materiales utilizados para su abordaje, atendiendo la realidad de las escuelas.

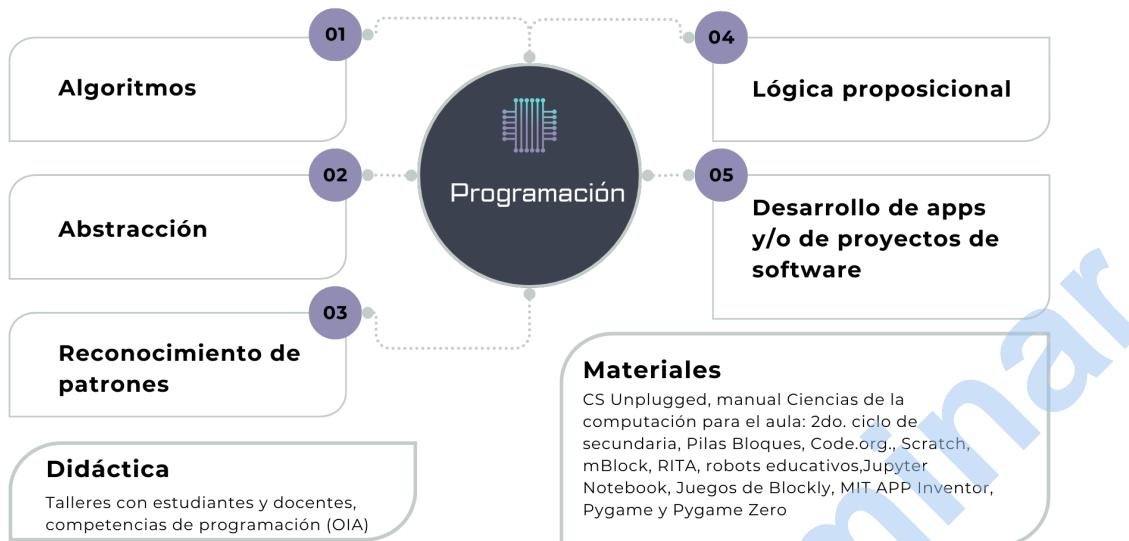


Figura 1. Acerca Programación: temas, enfoques didácticos y materiales

Concretamente la experiencia presentada en este trabajo, consistió en una instancia de trabajo con estudiantes y otra con docentes, del ciclo superior de las siguientes escuelas secundarias: ES N° 17 "Nicodemo Scenna", Liceo "Víctor Mercante" y Bachillerato de Bellas Artes "Prof. Francisco A. de Santo". La primera es una escuela provincial de nivel secundario que ofrece un ciclo superior con orientación en Ciencias Naturales y las dos últimas son escuelas secundarias preuniversitarias de la UNLP; el Liceo ofrece un ciclo superior con orientaciones en Ciencias Sociales, Ciencias Naturales y Gestión de las Organizaciones y el Bachillerato de Bellas Artes es una escuela especializada en arte, específicamente artes visuales y música. Para esta experiencia se eligieron escuelas secundarias que no son técnicas, que incluyen incipientemente la enseñanza de la Informática en distintos niveles de profundidad y contenidos.

En relación a la enseñanza de programación, en el plan de estudios de las escuelas, encontramos que: a) la ES N°17, en 4to año dispone de la asignatura cuatrimestral obligatoria NTICX (Nuevas Tecnologías para la Información, la Comunicación y la Conectividad); b) el Liceo dispone de tres espacios curriculares obligatorios relacionados a Informática: en primero y tercer año “Computación” de duración trimestral y cuatrimestral respectivamente, “Introducción a la Informática” en el ciclo de orientación en Ciencias Naturales y un espacio extraprogramático de “Programación con Python”, cuatrimestral, ofrecido a estudiantes del ciclo superior y, c) el Bachillerato de Bellas Artes no dispone de ningún espacio curricular vinculado a Informática. En esta diversidad de planes de estudios encontramos que tanto en la ES N° 17 como en el Liceo “Víctor Mercante” los estudiantes cuentan con espacios de formación en Informática, sin embargo específicamente en relación a programación, solo un grupo reducido del Liceo cursó en la escuela materias en las que les enseñaron a programar.

Los estudiantes que participaron de la experiencia se encuentran cursando el ciclo superior de la escuela secundaria. Para conocer los conocimientos previos en programación se administró una encuesta previa al inicio de la actividad que fue respondida por 22 estudiantes de los 30 que participaron de la actividad. Las respuestas obtenidas indican que sólo 3 estudiantes cursaron materias en su escuela en las que aprendieron a programar (estudiantes del Liceo) sin embargo 10 respondieron que están aprendiendo a programar por sus propios medios (YouTube, Streamers, páginas web, etc). Esta información “habla” del interés de los estudiantes en programación.

Los 10 docentes que participaron de la experiencia cuentan con formación diversa y en relación a las asignaturas a su cargo, 3 son profesores de materias de Informática (Computación y NTICX), solo 2 de ellos están enseñando a programar y los restantes son profesores de Matemática, Física, Química, Biología y Música. Sin embargo, todos acuerdan en la relevancia de incorporar la enseñanza de programación en la formación general de los estudiantes y en la falta de espacios de formación específicos para realizarlo.

A continuación se describe la experiencia, considerando la propuesta didáctica diseñada, la implementación de la misma, los resultados obtenidos y finalmente una reflexión sobre lo hecho.

## 2. Descripción de la Experiencia

La experiencia docente presentada en este trabajo consiste en el desarrollo de 2 talleres simultáneos de acercamiento a la programación, uno destinado a estudiantes y otro a docentes, en ambos casos del ciclo superior de las siguientes 3 escuelas secundarias, de gestión pública, de la localidad de La Plata: ES N° 17 "Nicodemo Scenna", Liceo "Víctor Mercante" y Bachillerato de Bellas Artes "Prof. Francisco A. de Santo". Cada taller se desarrolló en 2 encuentros de 2 horas de duración cada uno.

Como se describió en la sección anterior se trata de escuelas secundarias cuya oferta de formación es diferente y en lo específico a Informática y Programación, también difieren sus propuestas educativas. Por ello los saberes previos de los estudiantes y la formación de los docentes en esta área de conocimiento es diversa. Participaron del taller para estudiantes entre 18 y 30 estudiantes, y del taller para docentes entre 8 y 10 docentes. Esta variación se debe a que en el segundo encuentro dos docentes de una de las escuelas no pudieron concurrir y esto también determinó que sus estudiantes no concurrieran. Esta variación observada entre encuentros es frecuente que suceda en actividades de extensión universitaria con escuelas.

A continuación se describen las propuestas didácticas diseñadas para cada taller y su implementación.

### 2.1. Propuesta Didáctica

La metodología adoptada para esta experiencia fue la modalidad taller, tanto para los encuentros con estudiantes como con docentes. La intención es fomentar la participación y una dinámica de trabajo más relajada.

#### 2.1.1 Taller de acercamiento a la programación, con estudiantes

Este taller se organizó en dos encuentros de dos horas de duración cada uno:

- a) Primer encuentro: tuvo por objetivo introducir a los estudiantes en los conceptos básicos de programación mediante el uso de la herramienta Pilas Bloques.
- b) Segundo encuentro: a partir de los contenidos trabajados en el primer encuentro, el objetivo del segundo encuentro fue ponerlos en juego en el desarrollo de una app (aplicación móvil) mediante el uso de la herramienta MIT App Inventor.

Para ambos encuentros se decidió trabajar con herramientas educativas que tienen en común el enfoque de programación basada en bloques visuales y que están apoyadas en las teorías de aprendizaje constructivista y aprendizaje activo:

**Pilas Bloques**<sup>2</sup>: es una herramienta educativa y gratuita, de código fuente abierto, desarrollada en Argentina desde la iniciativa Program.AR<sup>3</sup> de la Fundación Sadosky, orientada a colaborar en el aprendizaje de programación, mediante un enfoque visual y lúdico. Utiliza bloques visuales que se encastren utilizando la metáfora de juegos de ladrillos, para crear secuencias de instrucciones de manera intuitiva, facilitando el aprendizaje de conceptos básicos como secuencias, bucles y condiciones, y desarrollando habilidades de pensamiento lógico y resolución de problemas. Está disponible online y para su descarga en computadoras con diferentes sistemas operativos.

**MIT App Inventor**<sup>4</sup>: es una herramienta educativa y gratuita, de código fuente abierto, diseñada y desarrollada originariamente por Google y mantenida desde 2012 por el MIT (Massachusetts Institute of Technology, USA). Permite diseñar y programar aplicaciones móviles mediante un entorno visual basado en bloques e instalar la app en el celular de manera sumamente intuitiva. Los conocimientos de programación se ponen en juego junto con otros elementos propios de la construcción de software como el diseño de interfaces gráficas, la gestión de los datos, el uso de sensores y dispositivos propios de los celulares (acelerómetro, cámara de fotos, etc) para desarrollar e implementar una app que funcione en celulares.

Ambos encuentros ofrecen un recorrido progresivo en conceptos de programación, comenzando con los fundamentos en Pilas Bloques y avanzando hacia la creación de aplicaciones móviles en MIT App Inventor.

#### Primer encuentro

##### a) Contenidos

- Introducción a Pilas Bloques: instalación, configuración y navegación por la interfaz.
- Creación de programas simples que usan: secuencias de instrucciones, estructuras iterativas para repetir acciones y estructuras condicionales para implementar decisiones. La creación y uso de procedimientos para introducir el concepto de modularización, también es un tema a abordar.

Para trabajar estos contenidos se seleccionaron desafíos de los ofrecidos en Pilas Bloques de la sección Principiantes, en el siguiente orden:

<sup>2</sup> Pilas Bloques: <https://pilasbloques.program.ar/>

<sup>3</sup> Program.AR: <https://program.ar/>

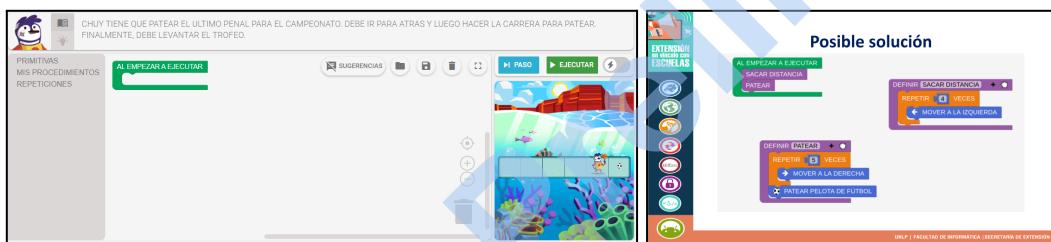
<sup>4</sup> MIT App Inventor: <https://appinventor.mit.edu/>

- Los primeros desafíos propuestos para resolver tienen como objetivo que los estudiantes se familiaricen con la lógica de la programación en bloques y con el concepto de secuencia de instrucciones. Para ello se eligieron los desafíos: del nivel principiante “Dieta a base de churrascos 1”, “Dieta a base de churrascos 3” (Figura 2, imagen de la izquierda) y “¡Marche una de lechuga y tomate! 2”.
- Se continúa con estructuras de control iterativas (repetición). Para ello se seleccionaron los desafíos: del nivel principiante “Lita, a puro vegetal 2” y “Las líneas de Coty 3” (Figura 2, imagen de la derecha).



*Figura 2. Algunos de los desafíos seleccionados para trabajar en el primer encuentro*

- Luego, con estructuras de control condicionales y para ello se seleccionaron los desafíos: del nivel principiante “Solo en ciertas ocasiones 3”, “¿Y sí no? 1”, “Más churrascos y ensaladas 3” y por último, “Agente secreto topotopo 3”.
- Posteriormente, se introdujo el concepto de procedimientos y para ello se creó un desafío inspirado en el desafío “Chuy hace jueguito” del nivel intermedio, que consiste en que Chuy tiene que hacer carrera, donde debe retroceder unos pasos para avanzar, para patear una pelota (Figura 3).
- Por último, a modo de cierre, se resolvió en forma conjunta el desafío “Contando astros” del nivel intermedio, ya que integra todos los conceptos vistos y que de forma discreta, incluye el uso de variables, elemento que se utilizará en el segundo encuentro.



*Figura 3. Desafío creado para explicar procedimientos*

### b) Metodología

La metodología de trabajo en el aula combina exposiciones explicativas por parte de los talleristas con demostraciones prácticas. En éstas, se introducen los conceptos mediante presentaciones breves y ejemplos en vivo, mientras que con las demostraciones prácticas se pretende que los participantes se familiaricen con la herramienta y las técnicas de programación a través la resolución guiada de desafíos (Figura 4).



*Figura 4. Diapositivas con explicaciones y demostraciones*

Los talleristas recorren las mesas de trabajo colaborando en la resolución de los desafíos, revisando el progreso, aclarando dudas según sea necesario y, realizando explicaciones si se detecta que determinados conceptos no están lo suficientemente claros. De esta manera se intenta que todos los grupos puedan resolver los desafíos propuestos. Una vez que la mayoría de los estudiantes tienen una posible solución, se hace una puesta en común de las distintas resoluciones a los desafíos.

No se les dio una instrucción de trabajar en grupos específicamente, pero algunos estudiantes se agruparon de a pares para resolver los desafíos.

### Segundo encuentro

El objetivo del segundo encuentro es continuar con el desarrollo de los conceptos de programación introducidos en el primer encuentro, e introducir a los estudiantes en el diseño y desarrollo de aplicaciones móviles utilizando MIT App Inventor.

### a) Contenidos

- Introducción a MIT App Inventor: incluyendo su instalación, configuración y navegación por la interfaz.
- Creación de una app desde cero: incluyendo el diseño de las pantallas (interfaz de usuario), la programación del comportamiento de los elementos de la interfaz de usuario (por ejemplo los botones), el uso de recursos como imágenes y sonidos para personalizar la app y el uso del hardware propio del celular como el acelerómetro.
- Realizar pruebas de la app construida e instalarla en los celulares.

Para trabajar estos contenidos se planteó a los estudiantes que el tema de la app a desarrollar serían las cartas Pokémon y para que sea un poco más interactivo y tenga mayor impacto, se llevaron algunas cartas reales. La propuesta de trabajo se organizó en varias etapas:

- Etapa 1: cargar en MIT App Inventor las imágenes del frente y reverso de las cartas Pokémon (previamente descargadas en la computadora) y agregarlas como recurso a la app.
- Etapa 2: diseñar la interfaz de usuario de la app incorporando un botón que contiene una imagen (de la carta). Al cliquear en el botón se cambia la imagen de la carta de reverso a frente.
- Etapa 3: agregar un sonido como recurso en la app que se reproduce al cliquear el botón.
- Etapa 4: agregar el acelerómetro como componente de la app para que al agitar el celular se gire la carta.

### b) Metodología

En este segundo encuentro se utiliza la misma metodología que en el primero. Se combinan exposiciones por parte de los talleristas en las que se presentan conceptos de MIT App Inventor y se realizan demostraciones prácticas que sirven de guía para diseñar interfaces de usuario y programar funcionalidades (Figura 5).



Figura 5. Modos en MIT App Inventor para programar

Se propone a los estudiantes que desarrollen una aplicación móvil que muestra cartas de Pokémon. Se espera que la app inicialmente muestre una carta con el dorso hacia arriba y al mover el celular (acelerómetro del teléfono) se revele la imagen del Pokémon. Además, la app incluye un botón que reproduce el sonido característico del Pokémon seleccionado. Por último, se propone a los estudiantes que resuelvan los desafíos (Figura 6) con el objetivo de poner en juego lo aprendido en la creación de la app: agregar botones volver la carta a su estado inicial, para reproducir sonido del Pokémon revelado, alinear ambos botones en la pantalla y cambiar la imagen de la carta al mover el celular (para ello es necesario manipular el acelerómetro).

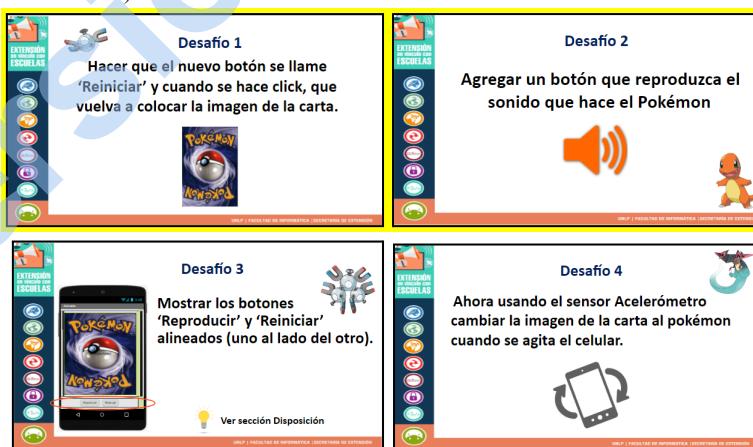


Figura 6. Desafíos a resolver en la app Pokémon

Los talleristas recorren las mesas de trabajo colaborando en el desarrollo de la app, revisando junto con los estudiantes las funcionalidades propuestas, ofreciendo retroalimentaciones continuas y alemando la discusión de ideas entre los grupos mediante preguntas cómo ¿Qué componentes de interfaz de usuario se deberían usar? ¿Qué propiedad del botón hay que modificar para asociar una imagen?

### Evaluación

A modo de evaluación del taller se administró una encuesta al finalizar el mismo con la intención de conocer la experiencia de los estudiantes al transitar el taller. Esta encuesta consta de 10 preguntas que abordan los siguientes aspectos: los contenidos aprendidos utilizando Pilas Bloques y MIT App Inventor, las dificultades experimentadas con

cada una de estas herramientas, y la opinión sobre la organización del taller. Además, los estudiantes pueden sugerir temas que les gustaría ver en futuras ediciones.

De las respuestas obtenidas se puede ver que en general calificaron el taller como muy bueno. Sin embargo, el 50% de ellos consideró que el tiempo asignado fue insuficiente. Respecto a las dificultades, el 66% encontró complicado el desafío “Contando astros” de Pilas Bloques, que integra todos los temas del primer encuentro, debido a su complejidad. En cuanto al uso de MIT App Inventor, no se encontraron mayores dificultades, aunque los estudiantes expresaron un interés en profundizar más en esta herramienta en futuras ediciones.

### **Recursos y materiales didácticos**

En ambos encuentros se usaron notebooks provistas por los laboratorios móviles de Facultad de Informática de la UNLP. Estos laboratorios consisten en un conjunto de notebooks que se trasladan al aula, previa reserva. En el aula se dispone de acceso a Internet mediante WiFi, cañón óptico instalado y mobiliario necesario para el desarrollo de la actividad como pizarras para hacer anotaciones. Las herramientas usadas se acceden online, con lo cual no fue necesario instalar en las notebooks ningún software adicional. En el caso del segundo encuentro, los estudiantes utilizaron sus celulares para probar la app desarrollada y luego instalarla en los mismos.

Se usaron presentaciones como material didáctico de apoyo para el seguimiento del taller y las herramientas Pilas Bloques y MIT App Inventor.

#### *2.1.2 Taller de acercamiento a la programación, con docentes*

Este taller se enfocó en escuelas donde no se enseña programación y su plantel docente, en su mayoría, tampoco tienen formación técnico-profesional.

Este taller, también se organizó en dos encuentros de dos horas de duración cada uno y se realizó en paralelo con el de estudiantes. Los temas abordados fueron: algoritmos, autómatas y programas, estructuras de control y procedimientos.

Se trabajó con actividades desenchufadas, que pueden abordarse sin uso de computadoras y luego se utilizaron las herramientas Pilas Bloques y Scratch. En especial, el enfoque de uso de esta última se dio en base a las propuestas surgidas durante el taller. A modo de ejemplo, la docente de Química planteó realizar una aplicación para “reconocer” elementos de la tabla periódica o fórmulas sencillas. Originalmente se planificó trabajar en este taller con las mismas herramientas que se utilizaron en el taller con los estudiantes, sin embargo al interactuar con el grupo de docentes, y recibir propuestas como la mencionada previamente, se decidió usar Scratch, dado que permite, de forma simple, abordar este tipo de actividades.

La formación heterogénea de los docentes, descripta previamente, generó debates sobre qué temas abordar de programación, con qué enfoque didáctico y en qué espacios curriculares posibles.

## **2.2. Contexto**

Los talleres se realizan en la Facultad de Informática de la UNLP en aulas que se encuentran equipadas con los materiales necesarios para llevar a cabo los talleres, pizarra con fibrones, proyector óptico y computadoras con acceso a WiFi, ambos talleres se realizaron de manera simultánea en un tiempo de dos horas (de 15:00 a 17:00).

El proyecto “Extensión en vínculo con escuelas secundarias” en el que se inscriben los talleres, busca articular con distintas escuelas de educación secundaria pertenecientes a las localidades de La Plata, Berisso, Ensenada y Magdalena, un trabajo de acercamiento situado al campo de conocimiento de la Informática. Específicamente en las experiencias relatadas en este trabajo se decidió trabajar con escuelas secundarias sin orientación ni especialidad tecnológica con la intención de introducir conceptos de programación que no forman parte de los contenidos enseñados en la escuela.

En el taller con los estudiantes participaron 30 estudiantes, 18 varones y 12 mujeres, de entre 15 a 18 años. Se administró una encuesta previa a la realización del taller para conocer los aprendizajes previos de programación con los que contaban los estudiantes, si fueron adquiridos en la escuela o en otros espacios y los intereses en el tema. Esto último sobre todo porque es común que en los proyectos de extensión los estudiantes sean “obligados” a participar dado que se propone como actividad pedagógica de la escuela. En términos generales, los estudiantes no poseían conocimientos previos de programación sin embargo coincidieron en el interés por aprender a programar.

En el taller con docentes, participaron 11 docentes: 2 de la ES N° 17 "Nicodemo Scenna", 3 del Liceo “Víctor Mercante” y 6 del Bachillerato de Bellas Artes. En relación a las formaciones, como se mencionó en la Introducción, es diversa. Las docentes de la ES N° 17 están a cargo de la materia NTICX y una de ellas cuenta con la titulación de “Especialista en didáctica de las Ciencias de la Computación” dictado en la provincia de Buenos Aires entre los años 2018 y 2021. Dos de las docentes del Liceo “Víctor Mercante” están a cargo de materias de Informática y cuentan con titulaciones universitarias en el campo, la tercera docente es profesora de Ciencias Naturales y coordinadora del espacio de Orientación de la escuela. Los docentes del Bachillerato de Bellas Artes cuentan con formación en Física, Matemática, Química, Biología y Música.

## **2.3. La Experiencia**

En relación a la propuesta didáctica, el taller con estudiantes aunque se pudo desarrollar en su totalidad requirió de un tiempo adicional para la actividad de cierre del primer encuentro, que debió postergarse para el inicio del segundo encuentro. En el segundo encuentro no se lograron desarrollar todas las actividades propuestas, en el tiempo previsto. Si bien hubo un retraso al inicio dado que se hizo el cierre del primer encuentro, las actividades que se esperaba los estudiantes realizarán requirieron más tiempo del previsto, no siendo posible desarrollar el encuentro en su totalidad en el tiempo previsto.

En relación a los contenidos del primer encuentro, se observaron algunas dificultades con las estructuras condicionales dado que los desafíos seleccionados requerían el uso de estructuras repetitivas y condicionales a la vez, siendo un error recurrente el uso de la sentencia “Si/Sino”. En este sentido será necesario revisar la selección de los desafíos. Por otra parte, en el uso de procedimientos se observó que la elección del nombre de los mismos no seguía ningún criterio que hiciera alusión a la tarea que realizaban los mismos. Al observar esta situación se conversó sobre las buenas prácticas a la hora de programar, ejemplo de ello la legibilidad.

En relación a los contenidos del segundo encuentro, se observaron algunas dificultades en la etapa 2 y la etapa 3: a) la implementación de la lógica del botón dado que el mismo botón realiza dos acciones distintas dependiendo de la imagen que se muestra de la carta (frente y Pokémon) y, b) el uso del audio, dado que se prestaba a la confusión el botón para reproducir sonido con la acción de la reproducción del mismo. La etapa 4 que proponía usar el acelerómetro para girar una carta, no pudo hacerse por falta de tiempo. Lo mismo ocurrió con el juego “Piedra, papel o tijera”.

La propuesta didáctica original del taller con docentes fue adaptada con el objetivo de trabajar con herramientas y estrategias adecuadas a las asignaturas y variedad de perfiles de los docentes participantes. Como positivo, se destacan las experiencias relatadas en primera persona sobre la enseñanza de programación, limitaciones de las escuelas y cómo éstas se van abordando. En este diálogo surgieron algunas ideas para trabajar con los estudiantes en materias como Matemática, Biología y Física, que obligó a un replanteo de los temas y herramientas a abordar. No solo se reforzaron los conceptos básicos, sino que se decidió cambiar la herramienta planificada originalmente. En vez de utilizar MIT App Inventor se decidió utilizar Scratch. Se dio un pantallazo general de la herramienta y luego se implementaron dos actividades simples.

En el proyecto se prevé la organización de más talleres en el segundo cuatrimestre, coordinando con los docentes las temáticas a abordar. En este caso surgió de la posibilidad de organizar un taller para trabajar con kits de robots educativos.

## **2.4. Documentación de la Experiencia**

Se administraron 2 encuestas en cada taller, una antes del inicio y otra al finalizar.

El objetivo de la encuesta inicial a los estudiantes fue conocer más sobre ellos, su recorrido e interés en programación. Los resultados indicaron que en su mayoría los estudiantes se encontraban cursando el ciclo orientado (ciencias naturales) y la especialidad música o artes visuales, dato que conocíamos. En relación a los aprendizajes previos en programación, el 86% contestó que no está aprendiendo programación en la escuela, con lo cual concluimos que los conocimientos previos en programación en general son muy básicos o nulos. Sin embargo, un número cercano al 50% respondió que estaba aprendiendo por sus propios medios y todos coincidían en tener interés en aprender a programar.

La encuesta del final del taller nos permitió conocer la opinión de los estudiantes luego de transitar el mismo. Todos coincidieron en que el taller fue bueno y que no tuvieron grandes dificultades para entender los conceptos. Sobre las preguntas de lo aprendido en cada encuentro, rescataron los conceptos de estructuras de control iterativas y procedimientos, programar botones y reproducir sonidos. En la pregunta sobre las dificultades, reconocen que el último desafío planteado “Contando astros ” les resultó difícil ya que era mucho más extenso que los demás y en el segundo encuentro crear y programar los componentes visuales y asignarles un sonido.

Respecto al taller con docentes, también se llevaron a cabo dos encuestas; una encuesta previa al taller donde se relevaron aspectos relacionados a las escuelas y asignaturas en las que trabajan, su percepción respecto a la importancia de enseñar a programar y, en caso de ya estar dictando algunos contenidos sobre la temática, qué metodología utilizan. Aquí surgieron tres aspectos a destacar: la heterogeneidad de los participantes respecto a su formación, que el 70% no está trabajando en estos temas, y en todos los casos que las escuelas fomentan la enseñanza programación.

Al finalizar el segundo día del taller, se aplicó una segunda encuesta en donde surgió la necesidad de un acompañamiento en la implementación de las actividades en la escuela y dejaron constancia de la intención de utilizar los robots educativos con los que cuentan en la escuela. Estos recursos, según se conversó en el encuentro, no se utilizan actualmente pero todos concuerdan que son un recurso muy interesante para introducir la temática de programación.

### **3. Resultados**

Respecto al taller con estudiantes, ambos encuentros generaron un notable interés, aunque el taller de MIT App Inventor recibió una mayor participación y entusiasmo, como se reflejó en las encuestas. Construir y poner a funcionar una aplicación móvil permite a los estudiantes observar en tiempo real los resultados de sus creaciones, lo que facilitó una experiencia de aprendizaje dinámica. En comparación con el primer encuentro, se observa que trabajar con MIT App Inventor logra captar más el interés de los estudiantes.

En relación a los objetivos de aprendizaje se observó que en general que tanto los desafíos como las actividades planteadas se pudieron resolver adecuadamente, aunque se observaron algunas dificultades fundamentalmente en el primer encuentro con el desafío integrador que requirió retomarlo en el segundo encuentro. No obstante, para futuros talleres, se sugiere mostrar diversas soluciones posibles para cada desafío y discutirlos desde diferentes puntos de vista, por ejemplo buenas prácticas de programación.

Respecto al taller con docentes, se pudo observar mucho interés, tanto en aquellos docentes que nunca habían abordado la programación como en aquellos que sí se encontraban trabajando el tema y que en el taller encontraron nuevas formas de abordaje.

Como se mencionó anteriormente, dos de las tres escuelas cuentan con kits de robots educativos, y durante esta experiencia surgió la iniciativa de plantear actividades con dichos recursos. Esto dio lugar al compromiso de trabajo en este segundo cuatrimestre.

En este sentido, dada la conformación del equipo de trabajo y los recursos del proyecto, hay un debate constante entre los temas y destinatarios de los talleres. Las escuelas técnicas con las que se trabaja, cuentan con otros recursos y formación y permiten abordar otras temáticas más complejas.

### **4. Conclusiones**

Si bien el acercamiento a la programación es un tema que se viene trabajando desde el proyecto de extensión “Extensión en vínculo con escuelas” desde diferentes propuestas, esta experiencia se destaca debido al doble enfoque, pensando en el trabajo con los estudiantes y docentes en simultáneo, pero con propuestas especialmente diseñadas para cada grupo.

En el caso de los estudiantes, los resultados no varían demasiado de experiencias anteriores: el desarrollo de aplicaciones móviles capta la atención e interés y, en este caso, la temática elegida, Pokémon, elevó dicho interés. También hay un reconocimiento por parte de los estudiantes en la programación como una actividad relacionada con las posibilidades laborales futuras, como herramienta útil para el futuro y en la curiosidad de saber de qué se trata. En términos generales, los estudiantes no poseían conocimientos previos de programación sin embargo coincidieron en el interés por aprender a programar.

En el caso de los docentes, el taller se constituyó como un espacio de reflexión sobre las razones de por qué incluir programación en la enseñanza obligatoria, qué habilidades cognitivas se pretende trabajar y qué estrategias utilizar en base a las realidades de las escuelas y de la diversidad de formación del plantel docente. Se analizaron los espacios curriculares de los planes de estudio de las escuelas que participaron del taller y las posibilidades de incluir programación. A modo de ejemplo, con un profesor de Física se analizó la posibilidad de incluir actividades de programación para enseñar temas como velocidad y tiempo. En el caso del Bachillerato de Bellas Artes, cuyo plan de estudios no contempla materias de Informática, un enfoque transversal podría ser adoptado para introducir programación en algunas materias de los espacios de formación general y específica (artes visuales y música). En este sentido, se evalúa como sumamente positivo el interés y el esfuerzo de los docentes por incluir programación en sus materias. Sin embargo, en términos prácticos este enfoque requiere de un acompañamiento continuo o de un trabajo con duplas docentes (docente de Informática y docente de la materia). En ES N° 17 y en el Liceo “Víctor Mercante” que cuentan en sus planes de estudio con espacios curriculares y extra programáticos de Informática y docentes con formación técnico-profesional, el foco de la reflexión fue contar con espacios de actualización y de nuevos materiales.

### **Referencias**

- [1] C. Queiruga, C Banchoff Tzancoff, S. Martin e I. Kimura, “Propuestas de acercamiento a la Informática para la escuela secundaria: contenidos, estrategias didácticas y materiales”. En 53 JAIIO, Simposio Argentino de Educación en Informática (SAEI), agosto de 2024. En proceso de publicación.
- [2] D. Levis, *Enseñar y aprender con informática/enseñar y aprender informática. Medios informáticos en la escuela argentina*. Tecnologías informáticas en la educación a principios del siglo XXI, Buenos Aires: Prometeo, 2007.

# **Inteligencia Artificial en la Escuela Secundaria. Experiencias de un proyecto de extensión universitaria**

Isabel Kimura<sup>1</sup>, Santiago Chaves<sup>1</sup>, Pascual Coudannes<sup>1</sup>, Valentin Domé<sup>1</sup>, Claudia Queiruga<sup>1</sup>, Soledad Gómez<sup>1</sup>

<sup>1</sup>LINTI (Laboratorio de Investigación en Nuevas Tecnologías Informáticas)-Facultad de Informática-UNLP, La Plata, Argentina

ikimura@lenti.unlp.edu.ar, santiagochaves242@gmail.com, pascualcoudannes@gmail.com, valentindome777@gmail.com, {claudiaq, sgomez}@info.unlp.edu.ar

## **Resumen**

Las experiencias docentes de este trabajo se desarrollan en el marco del proyecto de extensión universitaria acreditado y financiado por la Secretaría de Políticas Universitarias del Ministerio de Educación de la Nación Argentina, "Inteligencia Artificial en la Escuela Secundaria desde un Enfoque Regional". El objetivo del proyecto es co-construir un enfoque local y regional sobre los procesos de enseñanza y aprendizaje de la Inteligencia Artificial (IA), específicamente Aprendizaje Automático, Aprendizaje Profundo y las implicancias éticas del uso de estas tecnologías, para favorecer su introducción en los espacios curriculares de las especialidades Programación e Informática Profesional y Personal de las escuelas secundarias técnicas que participan de este proyecto. Las experiencias que aquí se analizan, buscan poner en diálogo estos nuevos saberes con situaciones-problema conocidas para los estudiantes y la comunidad de la escuela. La selección de los contenidos de IA, las adecuaciones al nivel secundario técnico con especialidad Informática, la selección de herramientas y la confección de situaciones-problema, fueron los principales elementos de las decisiones pedagógicas de estas experiencias docentes. La metodología para su desarrollo fueron los talleres con estudiantes y docentes, en espacios áulicos. Se realizaron dos jornadas-taller en la Facultad de Informática de la UNLP de las que participaron las dos escuelas integrantes del proyecto. Los talleres se organizaron en diferentes momentos e incluyen las siguientes actividades: 1) identificación de aplicaciones y dispositivos de la vida cotidiana que contienen diferentes formas de IA; 2) construcción de un modelo de aprendizaje automático para clasificación de imágenes; 3) desafiar el modelo construído y 4) debate acerca de las implicancias éticas. Para la evaluación de las experiencias se realizaron encuestas acerca de los temas trabajados y en caso de los docentes, el interés en incluir algunos de los temas en sus materias.

**Palabras clave:** Ética, Aprendizaje automático, MIT App Inventor, Google Teachable Machine, Entrenamiento de modelos

## **1. Introducción**

Esta experiencia se enmarca en el Proyecto de Extensión Universitaria titulado "Inteligencia Artificial en la Escuela Secundaria desde un Enfoque Regional", el cual fue aprobado y financiado en la convocatoria "Universidad, Cultura y Territorio 2022" de la SPU [1]. Este proyecto tiene como objetivo integrar conceptos de Inteligencia Artificial (IA), con un enfoque particular en Aprendizaje Automático (ML) y Aprendizaje Profundo (DL), en los programas curriculares de las escuelas secundarias técnicas especializadas en "Programación" e "Informática Profesional y Personal". Específicamente integran este proyecto la EEST N° 2 "Ing. Emilio Revuelto" de la localidad de Berisso, con especialidad "Informática Personal y Profesional" y la EEST N° 9 de la localidad de La Plata con especialidad "Programación". El proyecto busca colaborar con los docentes de estas instituciones para desarrollar conjuntamente materiales educativos sobre IA.

Entendemos que la IA es una tecnología muy presente en la vida cotidiana, por lo que consideramos crucial su incorporación en la educación técnica, adaptándola a un contexto local y regional. El objetivo es dotar a los estudiantes con competencias digitales que sean pertinentes en la sociedad actual. Además, el proyecto aborda aspectos éticos relacionados con la IA. Recientemente tecnologías como ChatGPT han demostrado el potencial de la IA para transformar la manera en que las personas interactúan con la información y los servicios digitales. Sin embargo, a pesar de su omnipresencia, muchas personas siguen sin conocer la tecnología subyacente detrás de estos productos y servicios. El desarrollo acelerado y la masiva respuesta de las empresas sobre desarrollos de IA, posibilitan la pregunta sobre el abordaje de estos contenidos y las lógicas en que estas tecnologías operan, tanto en ámbitos cotidianos como a nivel de desarrollo profesional. En este contexto, nuestro proyecto considera esencial incorporar temas de IA en el currículo escolar, promoviendo debates sobre su uso y entendimiento. Estas incorporaciones deberían dialogar con los conceptos fundamentales de la Informática, como algoritmos, estructuras de datos y programación, para la profundización de propuestas educativas que fortalezcan la formación ciudadana en el uso consciente y responsable de este tipo de tecnologías.

## 2. Descripción de la Experiencia

El punto de partida de las actividades del proyecto fueron los 2 encuentros realizados bajo la modalidad taller de los que participaron estudiantes y docentes de las dos escuelas que integran el proyecto. La duración de los encuentros fue de 2 horas, participaron 20 estudiantes y 4 docentes.

Durante los encuentros, los estudiantes trabajaron con herramientas como Google Teachable Machine (GTM)<sup>1</sup> y MIT App Inventor<sup>2</sup>, centrando su actividad en la creación, entrenamiento y evaluación de un modelo de ML. Además, se abordaron cuestiones éticas relacionadas con el uso de la IA. Los docentes participaron activamente en los debates apoyando la propuesta educativa.

### 2.1. Propuesta Didáctica

El taller “Crear mi propia IA” es la primera actividad del proyecto de extensión “Inteligencia Artificial en la Escuela Secundaria desde un Enfoque Regional”. Los contenidos y el material didáctico se estudiaron y se desarrollaron durante el primer semestre del 2024, con la colaboración de tres becarios de extensión. Los becarios son estudiantes avanzados de la Facultad de Informática. El trabajo de los mismos fue supervisado por docentes investigadores, integrantes del proyecto de extensión, que poseen experiencia en contenidos de IA y en educación.

Los objetivos de este taller son:

- Generar capacidades, por parte de los estudiantes y docentes, de reconocimiento de técnicas de ML y DL en aplicaciones de uso cotidiano.
- Generar capacidades para construir aplicaciones informáticas, tanto móviles o como web, que hagan uso de técnicas de ML y DL, en situaciones-problema contextualizadas.
- Construir un enfoque de la enseñanza de una IA justa que no reproduzca sesgos, teniendo en cuenta las implicancias éticas y sociales de los usos de la IA.

El taller se diseñó para ofrecer a los estudiantes una comprensión integral de la IA, combinando teoría y práctica en los dos encuentros. A continuación se detalla la planificación de los contenidos y actividades para cada día.

#### Día 1: Introducción a la Inteligencia Artificial y Creación de Modelos

##### Conceptos Fundamentales de Inteligencia Artificial

- **Definición y Aplicaciones:** introducción de los conceptos básicos de IA, qué es la IA y cómo se manifiesta en la vida cotidiana, destacando ejemplos como asistentes virtuales y sistemas de recomendación.
- **Ramas de la IA:** incluyendo el ML y el DL, diferencias y aplicaciones.
- **Aplicaciones en el Mundo Real:** diversas aplicaciones de la IA en distintos sectores, proporcionando ejemplos concretos y actuales.

##### Actividad Práctica: Creación de Modelos con Google Teachable Machine

<sup>1</sup> Google Teachable Machine es una herramienta en línea que permite crear modelos de aprendizaje automático de manera sencilla y sin necesidad de programación. Se puede entrenar modelos para clasificar imágenes, sonidos o poses mediante una interfaz intuitiva. <https://teachablemachine.withgoogle.com/>

<sup>2</sup> MIT App Inventor es una plataforma de programación visual en línea que permite crear aplicaciones móviles completas para Android. Utilizando un enfoque basado en bloques, esta herramienta facilita el desarrollo rápido y sencillo de aplicaciones, desde proyectos básicos hasta más complejos, en un tiempo significativamente menor que los métodos de programación convencionales. Se puede utilizar accediendo con una cuenta de gmail. <https://appinventor.mit.edu/>

- **Desafío 1: Construcción del Bananómetro:** los estudiantes iniciarán la práctica creando un modelo de clasificación de imágenes simple para detectar la madurez de una banana utilizando GTM.
- **Desafío 2: Desarrollo de un Modelo de IA:** los estudiantes crearán un modelo de IA propio utilizando GTM, aplicando conceptos aprendidos.
- **Desafío 3: Mejora y Revisión del Modelo:** se les pedirá a los estudiantes que mejoren sus modelos y realicen nuevas pruebas para evaluar y optimizar su resultado.

#### **Problemas y Consideraciones Éticas en la IA**

- **Sesgos en la IA:** se discutirá cómo los sesgos en los datos pueden afectar el funcionamiento de los modelos de IA y las implicaciones de estos sesgos.
- **Derechos de Autor:** se abordará la cuestión de los derechos de autor relacionados con los datos y modelos utilizados en IA.
- **Aspectos Éticos:** se explorarán las implicaciones éticas del uso de IA, incluyendo cuestiones de privacidad y seguridad.

#### **Día 2: MIT App Inventor e Integración de un Modelo de IA a una Aplicación Móvil**

##### **Introducción a App Inventor**

- **Diseño y Bloques:** se ofrecerá una introducción a MIT App Inventor. Modo diseñador para crear la interfaz de usuario y modo bloques para la programación visual.

##### **Actividad Práctica: Integración del Modelo de IA en una Aplicación Móvil**

- **Desafío 1: Aplicación "Fake Voices":** los estudiantes completarán una aplicación móvil que reproduce un discurso en diferentes voces generadas al cambiar la velocidad y el tono del discurso.
- **Desafío 2: Importación del Modelo de GTM:** se integrará el modelo creado el primer día en una aplicación móvil desarrollada con App Inventor, permitiendo a los estudiantes llevar su creación en sus dispositivos móviles.

##### **Reflexión sobre el Futuro de la IA**

- **Exploración de Tendencias Futuras:** se discutirá el futuro de la IA, incluyendo nuevas tendencias y desarrollos emergentes.
- **Impacto en la Sociedad:** se reflexionará sobre el impacto potencial de la IA en diferentes aspectos de la sociedad y el papel de los estudiantes en este contexto.

Este taller está diseñado para proporcionar a los estudiantes una introducción a los fundamentos de la IA y sus implicaciones éticas, centrándose en la creación y comprensión de modelos de IA, en lugar de solo su uso. Además, el taller abordará los peligros asociados, como los sesgos y las implicaciones de un entrenamiento inadecuado de los modelos.

La experiencia permitirá a los estudiantes crear y experimentar con su propio modelo de aprendizaje automático utilizando GTM. Los estudiantes podrán integrar su modelo en una aplicación móvil desarrollada con MIT App Inventor, lo que les permitirá llevar su creación en sus dispositivos móviles y explorar aplicaciones prácticas de la tecnología.

Se invita a los estudiantes a reflexionar sobre cómo las aplicaciones de IA en redes sociales o sistemas de recomendación pueden recoger, almacenar y utilizar datos personales, y cómo estos usos deben ser regulados para proteger la privacidad de los individuos. Al incorporar estos temas en el taller, los estudiantes no solo aprenderán sobre el funcionamiento técnico de la IA, sino que también desarrollarán una mayor conciencia sobre el impacto ético y social de las tecnologías que están moldeando el futuro. Este enfoque integral les permitirá abordar la IA con una perspectiva crítica y responsable, preparándolos para participar en la creación y uso de tecnologías de manera ética y considerada.

Se optó por la metodología de taller para la implementación de las actividades. Esta metodología facilita una participación activa de los estudiantes, permitiéndoles aprender a través de la práctica directa y la resolución de desafíos concretos. Promueve una comprensión más profunda al involucrar a los estudiantes en la ejecución de actividades prácticas y en la resolución de problemas [2].

Además, el enfoque de taller fomenta el trabajo colaborativo y el debate entre los estudiantes, enriqueciendo el proceso de aprendizaje mediante la interacción y el intercambio de ideas con sus compañeros y con los docentes.

Durante el taller, se utilizarán presentaciones de diapositivas para guiar las actividades, y cada pareja de estudiantes contará con una notebook para llevar a cabo las tareas propuestas. Las actividades se estructuraron en forma de

desafíos: primero, los estudiantes abordan los problemas de manera independiente y, posteriormente, se realiza una puesta en común para compartir y discutir las soluciones encontradas.

Consideramos que esta metodología es la adecuada porque asegura una experiencia más completa y significativa, al combinar aprendizaje práctico con colaboración y reflexión colectiva.

## 2.2. Contexto

El taller se realizó en las instalaciones de la Facultad de Informática de la Universidad Nacional de La Plata (UNLP). El aula utilizada cuenta con un proyector, pizarras de tiza y fibrón. Además, se les proveyó a cada par de estudiantes con una notebook conectado a Internet para que puedan seguir las diapositivas y realizar las actividades.

Los participantes, estudiantes de 6to año, cuentan con una representación femenina relativamente baja (6 estudiantes mujeres), representando un 30%. Los docentes presentes son del tramo de la especialidad.

Los estudiantes de ambas escuelas técnicas tienen una sólida base en programación, algoritmos, sistemas operativos, redes y bases de datos, así como experiencia en el desarrollo de software y principios básicos de seguridad informática. Estos conocimientos previos proporcionan una base sólida para abordar temas más avanzados como la IA y el ML.

## 2.3. La Experiencia

El taller se llevó a cabo en dos encuentros presenciales de dos horas cada uno, con una semana de intervalo, durante los meses de junio y julio de 2024, en la Facultad de Informática de la UNLP. En el primer encuentro, el enfoque principal fue introducir los conceptos básicos de IA y desarrollar un modelo de clasificación de imágenes. El segundo encuentro tuvo como objetivo principal que los estudiantes integraran este modelo en una aplicación móvil, utilizando MIT App Inventor.

A continuación, se detallan las actividades desarrolladas cada día, organizadas en diferentes momentos:

### 2.3.1. Taller día uno:

#### 2.3.1.1. Momento 1: Introducción a la IA

En el primer encuentro, se introdujeron los objetivos y la dinámica del taller, y se presentó a los tres becarios de extensión quienes estuvieron a cargo del dictado del taller. Para familiarizar a los estudiantes con los temas y evaluar sus conocimientos previos, se llevó a cabo una actividad inicial utilizando Mentimeter<sup>3</sup>. Las preguntas que se eligieron para la introducción fueron:

- “**Para vos ¿Qué es la Inteligencia Artificial?**”: esta pregunta inicial tuvo como objetivo explorar la percepción de los estudiantes sobre el concepto de IA. Para nuestra sorpresa, las respuestas fueron bastante satisfactorias, indicando que tenían un entendimiento sólido del término. En la Figura 1 se muestran las respuestas recogidas destacando que varios estudiantes describieron la IA como “una herramienta”.
- “**¿Dónde la podemos encontrar?**”: esta pregunta tenía como objetivo profundizar en el conocimiento de los estudiantes sobre las aplicaciones de la IA. Las respuestas fueron bastante relevantes, ya que los alumnos mencionaron una variedad de campos y dispositivos donde se utiliza IA. Entre las respuestas se destacaron especialmente las “aplicaciones”, como se ilustra en la Figura 1.
- “**¿Cómo se imaginan que la IA puede ayudar en el futuro?**”: con esta última pregunta se pretendía explorar las expectativas de los estudiantes sobre el impacto futuro de la IA. El objetivo era comprender su visión sobre el potencial de la IA. Las respuestas, como se ilustra en la Figura 1, indicaron que los estudiantes creen que la IA será fundamental para resolver problemas en diversos campos, desde la salud hasta la seguridad y la industria.

Se continuó con una actividad en la que los estudiantes clasificaron una serie de objetos para determinar si contenían o no IA para ello utilizaron Google Jamboard<sup>4</sup>. Esta herramienta facilitó una clasificación colaborativa, generando debates interesantes entre los estudiantes. Por ejemplo, decidieron ubicar el microondas en la categoría de “NO IA” debido a que la imagen mostraba un modelo antiguo, mientras que no pudieron clasificar el auto, ya que no podían determinar si era un vehículo inteligente solo a partir de la imagen. La Figura 2 muestra el resultado final de esta actividad.

<sup>3</sup> Mentimeter es una herramienta web que permite interactuar en tiempo real con una audiencia a través de encuestas. Los usuarios responden desde sus dispositivos móviles o computadoras, y los resultados se muestran en pantalla en tiempo real, facilitando la interacción y el análisis de las respuestas. <https://www.mentimeter.com/es-ES>

<sup>4</sup> Google Jamboard es una aplicación de pizarra digital desarrollada por Google, diseñada para facilitar la colaboración en tiempo real y la creación de contenido visualmente atractivo. Disponible en: <https://jamboard.google.com/>

## Para vos, ¿Qué es la Inteligencia Artificial? ¿Dónde la podemos encontrar?

20 responses

una base de datos  
un sistema automatizado  
es un sistema automatizado  
red neuronal es una base de datos  
es un sistema automático  
**una herramienta**  
un cerebro artificial una ayuda  
un sistema adaptable  
forma de brindar ayuda  
so que imita al humano  
maquinas que resuelven  
chatgpt ser informático  
conjunto if y elif

34 responses

en motores de navegación  
en redes sociales  
computadoras  
celulares dispositivos en los celulares  
en una empresa software  
en las industrias en una escuela  
**internet**  
**aplicaciones**  
en aplicaciones robots  
en un hospital Páginas web  
en búsqueda de respuestas  
en la robótica  
en base de datos

## ¿Cómo se imaginan que la IA puede ayudar en el futuro?

12 responses

descubrimientos médicos  
para la automatización  
ayudamiento de datos  
para ayudar en caso policiales  
va a simplificar todo  
razonamiento de datos en el desarrollo  
simplificación de tareas  
accesibilidad de datos  
mejoras industriales

Figura 1: Nubes de palabras del Mentimeter con las respuestas a las preguntas: "Para vos ¿Qué es la Inteligencia Artificial?", "¿Dónde la podemos encontrar?", "¿Cómo se imaginan que la IA puede ayudar en el futuro?"

Para concluir la introducción a los fundamentos de IA, se presentaron a los estudiantes diversas aplicaciones como Suno<sup>5</sup>, ThisPersonDoesNotExist<sup>6</sup>, Luma Dream Machine<sup>7</sup>, Tesla<sup>8</sup>, ChatGPT y ChatGPT-4<sup>9</sup>. En esta etapa, se consolidaron los conceptos de IA, ML y DL, abordando sus relaciones y diferencias. Los estudiantes participaron activamente en el debate, analizando qué tipo de IA se aplicaba en cada ejemplo presentado.

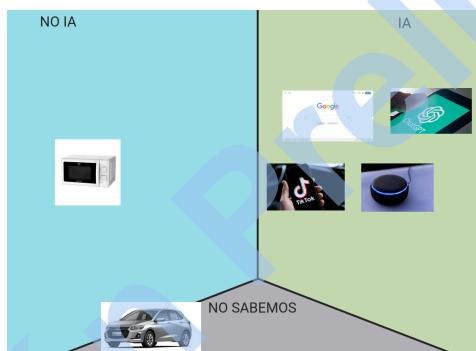


Figura 2: Resultados de la actividad sobre reconocimiento de artefactos y aplicaciones con IA usando Google Jamboard.

### 2.3.1.2. Momento 2: Introducción a Google Teachable Machine (GTM)

GTM es una plataforma web desarrollada por Google que facilita la creación de modelos de aprendizaje automático de manera rápida y sencilla, 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. Su facilidad de uso la hizo ideal para los objetivos del taller.

El primer desafío, que denominamos "Bananómetro"[3] es una actividad basada en uno de los tutoriales que acompaña a la herramienta. La misma consistió en desarrollar un modelo de clasificación de imágenes para predecir la maduración de una banana utilizando la cámara de la computadora. Los estudiantes debían clasificar bananas con diferentes grados de maduración (verde, madura, pasada y no es banana) y entrenar el modelo para distinguir entre ellas. Para el desarrollo de esta actividad se llevaron bananas reales con diferentes grados de maduración.

<sup>5</sup> Suno es un sistema de inteligencia artificial que puede crear canciones basadas en descripciones. Al ingresar un prompt o comando con el tipo de canción y un título deseado, Suno genera una canción acorde a esa descripción. <https://suno.com/>

<sup>6</sup> ThisPersonDoesNotExist es un generador de rostros artificiales que utiliza IA para crear imágenes de personas que no existen. Basado en StyleGAN, el sistema genera fotos realistas de personas inventadas. <https://thispersondoesnotexist.com/>

<sup>7</sup> Luma Dream Machine es una herramienta gratuita de IA que convierte fotos en videos animados de alta calidad. Lanzada por Luma AI en junio de 2024, permite crear animaciones a partir de imágenes y descripciones. Puedes probarla en su sitio web aquí. <https://lumalabs.ai/dream-machine>

<sup>8</sup> Tesla es una empresa que utiliza algoritmos avanzados de visión por computadora para la conducción autónoma de vehículos, incorporando los últimos avances tecnológicos en automoción. <https://www.tesla.com/AI>

<sup>9</sup> ChatGPT es un modelo de lenguaje desarrollado por OpenAI que puede mantener conversaciones, responder preguntas y generar texto en función de las indicaciones que recibe.

ChatGPT-4 es la versión más avanzada de este modelo, con mejoras significativas en comprensión y generación de texto, proporcionando respuestas más precisas y contextualmente relevantes. <https://openai.com/chatgpt/>

Se introdujeron conceptos clave como **modelos de aprendizaje automático**, que son estructuras de IA que aprenden a partir de datos para hacer predicciones; **clases**, que son las categorías en las que se clasifica la información, como los distintos niveles de maduración de las bananas; **la etapa de entrenamiento**, donde el modelo aprende a partir de datos etiquetados; y **la etapa de testeо**, en la que se evalúa el modelo con datos nuevos para medir su precisión y capacidad de generalización. En la Figura 3, se puede observar que el modelo clasifica una imagen de una banana como “Madura” con un **nivel de confianza** del 98%.

Para generar imágenes en la etapa de entrenamiento se propuso a los estudiantes usar la webcam y tomar diferentes imágenes de una banana cambiándola de posición.

Luego de haber terminado este desafío, se hizo una reflexión grupal en donde se compararon los diferentes modelos creados y se analizaron algunos de los problemas más comunes de entrenar un modelo de IA. En algunas de estas muestras aparecieron imágenes de una mano sosteniendo la banana. La pregunta que surgió fue: Y la mano, ¿forma parte de las imágenes de la muestra? ¿Cómo construyeron la muestra de la clase “No es banana”?

También, se intentó desafiar los modelos realizados, preguntando a los estudiantes qué pasaría si en vez de que las imágenes tengan el fondo blanco de la pared se usara el pizarrón verde o qué ocurriría si se mostraran dos bananas en distintos grados de maduración al mismo tiempo, e invitándolos a probar en sus modelos particulares. Esto fue con la intención de que los estudiantes tuvieran noción de que tipos de datos elegir para el segundo desafío.

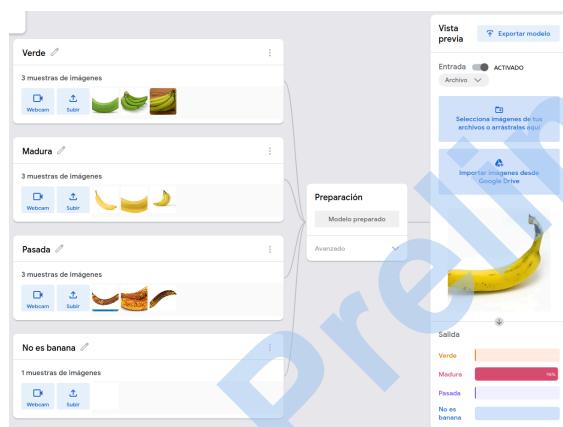


Figura 3: Toma de muestras, entrenamiento y testeо del Bananómetro.

#### 2.3.1.3. Momento 3: Entrenamiento de modelo propio

Una vez que los estudiantes se familiarizaron con GTM, el segundo desafío consistió en que entrenaran su propio modelo de clasificación de imágenes. Los estudiantes eligieron un dominio de su interés para realizar la clasificación de imágenes. Contaron con aproximadamente 45 minutos para desarrollar sus modelos en grupos de 2 o 3 estudiantes. Las imágenes podían provenir de Internet o ser capturadas con la cámara de la notebook.

Durante esta actividad, los estudiantes debían establecer el objetivo del modelos de clasificación de imágenes e identificar las clases que involucran. La selección de un conjunto de datos adecuado para el entrenamiento fue una etapa crucial. Los estudiantes aprendieron a identificar y seleccionar datos adecuados para asegurar su calidad y utilidad. En la etapa de testeо se realizaron pruebas con imágenes de entrenamiento y otras no vistas previamente, lo que permitió evaluar el rendimiento del modelo y realizar los ajustes necesarios. Se llevaron a cabo mejoras iterativas en el modelo, lo que implicó regresar a la fase inicial para ajustar y refinar el conjunto de datos, así como para recalibrar el modelo según los resultados obtenidos durante las pruebas. Con respecto a los parámetros (número de épocas, el tamaño del lote y la tasa de aprendizaje.) se decidió que no los modifiquen ya que no resultan relevantes en el objetivo del desafío.

En la tabla 1 se describen algunos de los modelos de clasificación de imágenes creados por los estudiantes utilizando GTM.

Tabla 1: Ejemplos de modelos creados por los estudiantes

Objetivo	Clases	Resultado	Observaciones
Clasificar rasgos de personas asiáticas	Coreanos, chinos y japoneses	No se pudo alcanzar	Los rasgos entre cada uno eran demasiados similares y la cantidad de datos no fue suficiente. Las imágenes encontradas no todas presentaban de forma verás los rasgos de cada uno, puesto que las fotografías con las que se contaban en su mayoría no eran de personas civiles, sino más bien de celebridades, que no representaban al promedio de la población

Clasificar autos de distintos colores	Autos rojos y autos negros	Se pudo alcanzar	Este modelo podemos decir que funcionó como se esperaba en la totalidad de los casos de testeo, pero no se llegó a encontrar un propósito para el mismo.
Clasificar expresiones faciales	Feliz y enojado	Se pudo alcanzar	El modelo fue entrenado con imágenes tomadas directamente desde la cámara de la notebook utilizada, por lo cual los rostros pertenecían a dos estudiantes nada más. A pesar de esto el modelo mostró buenos resultados al probarse con otras personas.
Clasificar verduras	Cilantro, perejil y brócoli	No se pudo alcanzar	Este modelo tenía un propósito bien definido pero por la falta de muestras significativas, y al ser clases de objetos tan parecidos entre sí, no se pudo lograr un buen resultado al intentar clasificar.
Clasificar jugadores de fútbol	Advíncula, Fabra y Merentiel	No se pudo alcanzar	Se eligieron jugadores de un mismo equipo y si bien era un modelo interesante, los resultados no fueron los esperados debido a que muchas imágenes eran bastante similares entre las clases. Todas tenían el mismo fondo, las camisetas que usaban eran las mismas y la calidad de las imágenes no ayudaba a distinguir los rasgos faciales.
Clasificar personas de distintos rangos etarios	Joven, adulto y viejo	No se pudo alcanzar	El grupo terminó eligiendo esta temática luego de algunos intentos fallidos con otras. Sin embargo, también tuvieron problemas por la falta de datos de entrenamiento utilizados, lo que hizo que el modelo se equivocara con frecuencia.
Clasificar aves, insectos y animales	Ave, insecto y animal	No se pudo alcanzar	El mayor problema encontrado fue utilizar categorías tan abarcativas y diversas. Ocurría que el modelo se confundía con los colores de las distintas aves, creyendo que algunas aves como loros eran insectos, ya que para las imágenes de aves utilizaron especies de tonos grises.
Clasificar distintos tamaños de perros	Pequeño, mediano y grande	Se pudo alcanzar	Este grupo primero intentó realizar un modelo para distintas razas de perros, pero tuvieron dificultades al utilizar clases muy similares entre sí, por lo que pasaron a usar distintos tamaños de perros. A pesar del cambio, se obtuvieron buenos resultados y se realizó una buena metodología para el armado y entrenamiento de su modelo.

De la Tabla 1 se puede observar que la mayoría de los grupos no pudo alcanzar un modelo final que cumpla con los objetivos propuestos. Esto es debido a factores como la selección de clases demasiado extensas, no pudiendo abarcar todos los tipos de ejemplos dentro de estas, clases muy similares entre sí, haciendo necesario que se utilice mayor cantidad de imágenes o cantidad insuficiente de datos de entrenamiento. Los problemas se encontraron tanto en la elección del problema a resolver como en los datos para entrenar.

Durante el proceso de entrenamiento, se ofrecieron diversas recomendaciones a los estudiantes sobre la importancia de elegir adecuadamente los datos de entrenamiento, definir claramente el objetivo del modelo y comprender cómo el modelo interpreta las imágenes para realizar la clasificación. Se enfatizó que, en algunos casos, puede ser difícil crear un modelo lo suficientemente preciso para cumplir con el propósito inicial. Por ejemplo, en el caso del modelo diseñado para clasificar cilantro y perejil, el objetivo no se logró debido a la alta similitud entre estas dos hierbas aromáticas, lo que podría requerir herramientas y técnicas más avanzadas para distinguirlas con precisión.

Cuando la mayoría de los estudiantes tenía su modelo creado y entrenado, se les solicitó que buscarán sugerencias para mejorarlo mediante el uso de chatbots de IA. Se recomendaron ChatGPT y Bing por no requerir registro previo. Los estudiantes consultaron estos chatbots y recibieron recomendaciones sobre cómo mejorar sus modelos, tales como aumentar el conjunto de datos, mejorar la calidad de los datos, optimizar la arquitectura del modelo y ajustar los parámetros. Posteriormente, aplicaron algunas de estas sugerencias, como el aumento del conjunto de datos y la mejora de la calidad de los datos, para perfeccionar sus modelos.

Una vez que terminaron de crear, entrenar y testear su modelo, se pidió a los estudiantes que se guardaran el modelo creado para poder usarlos en la siguiente jornada del taller.

#### 2.3.1.4. Momento 4: Debate

Como cierre del primer día del taller, se dedicó un espacio a la reflexión y el debate sobre diversas problemáticas de la IA, como cuestiones legales, sesgos, problemas filosóficos, discriminación y derechos de autor. Este intercambio enriquecedor reveló distintas perspectivas sobre los usos, ventajas y desventajas de la IA.

Uno de los temas destacados fue el debate sobre el **uso de la IA en las escuelas**. Se discutió si la IA debería utilizarse para realizar tareas, crear material para clases o como fuente de información. Tanto docentes como estudiantes

compartieron sus puntos de vista, y la idea predominante fue mantener una actitud crítica hacia los contenidos generados por la IA.

Otro debate relevante fue la implicancia de la IA en el contexto legal, particularmente en relación con la responsabilidad y culpa por los resultados que la IA pueda generar, como en diagnósticos médicos o juicios. Aunque no se alcanzó una respuesta definitiva, se subrayó la importancia de la **ética en el uso de estas herramientas**, dado que un uso indebido puede tener consecuencias negativas para muchas personas.

También se enfatizó la necesidad de contar con datos de entrenamiento de alta calidad para mejorar los modelos de IA. Se destacó que las muestras deben ser significativas y **minimizar los sesgos**, una enseñanza fundamental que los estudiantes deben llevarse del taller.

### **2.3.2. Taller día dos:**

En el segundo encuentro del taller, los conceptos abordados se enfocaron en la utilización de MIT App Inventor para crear aplicaciones móviles que incluyan IA y en la importación de un modelo de GTM a estas aplicaciones. Además, se discutió el futuro de la IA y su creciente adopción como tecnología innovadora.

#### **2.3.2.1. Momento 1: Aplicación con IA usando MIT App Inventor**

El encuentro comenzó con un repaso de los conceptos tratados anteriormente, como los modelos de aprendizaje, las fases de entrenamiento y testeo, y la importancia de entrenar modelos que minimicen sesgos.

Luego, se introdujo MIT App Inventor. Aunque algunos estudiantes ya estaban familiarizados con la herramienta, se ofreció una breve explicación para aquellos que no la conocían, destacando el modo Diseñador y el modo Bloques, así como las funciones esenciales para su uso.

Como primera actividad usando MIT App Inventor se presentó el desafío inicial llamado “FakeVoices”<sup>10</sup>. En esta actividad se trató el concepto de DeepFake (o ultra falso) y los conflictos éticos de este, al problematizar la creación de discursos mediante la manipulación de voces de personas reales con técnicas de IA. Está basada en una actividad incluida dentro del proyecto “Artificial Intelligence with MIT App Inventor”<sup>11</sup> que 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.

En esta actividad, se pidió a los estudiantes que completaran y usaran la app FakeVoices. El proyecto final ofrece las funcionalidades de conversión de texto a voz, mediante un conjunto predeterminado de voces sintéticas en un idioma determinado por la configuración del dispositivo y conversión de voz a texto que reconoce “on the fly” (al vuelo) el idioma del hablante y lo transcribe. Además, se permite modificar el tono y la velocidad de la voz. Se continuó trabajando en torno a la idea de la autenticidad de las voces creadas mediante dos ejemplos actuales y conocidos, en los que se manipulan voces utilizando IA. Utilizando el caso del actor Val Kilmer [4], que se trata de un uso autorizado y legítimo, que permite que el actor recupere su “identidad vocal” y, por otro lado, el rapero Jay-Z [5], que se trata de un plagio en donde se pone la voz del cantante en un discurso que él no hizo.

Durante el desarrollo de la actividad, en donde los estudiantes debían completar la app, tardaron más tiempo de lo previsto en completar las funcionalidades, quedando poco tiempo para el debate sobre las implicancias éticas en el uso de la voz.

A modo de cierre de esta actividad se llegó a la conclusión que la imitación utilizando IA es una herramienta que puede ser utilizada tanto para ayudar a las personas que lo necesitan o para causar perjuicios.

#### **2.3.2.2. Momento 2: Importación del Modelo de GTM**

Siguiendo con los desafíos se planteó como actividad importar el modelo de IA que armaron los estudiantes la clase anterior a un proyecto de MIT App Inventor para poder utilizarlo en una aplicación desde sus celulares. El primer paso fue explicar cómo descargar, agregar y utilizar extensiones dentro de la herramienta, ya que se necesita una para conectar el modelo de GTM con MIT App Inventor.

Se les propuso a los estudiantes que completen la app proporcionada, incorporando el modelo de GTM construido en el encuentro anterior. Una vez terminada, la app permite sacar una foto, analizarla y clasificarla utilizando el modelo creado por los estudiantes y mostrar los resultados obtenidos. En la Figura 4, se muestra una app creada con el modelo del Bananómetro.

Esta actividad demostró lo sencillo que es integrar un modelo de IA en una aplicación móvil utilizando MIT App Inventor, permitiendo a los estudiantes tener sus propios modelos en sus celulares.

<sup>10</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)

<sup>11</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>



Figura 4: Aplicación proporcionada utilizando el modelo del Bananómetro.

#### 2.3.2.3. Momento 3: Reflexión sobre el Futuro de la IA

Se les mostraron unas imágenes a los estudiantes junto con la pregunta “¿Qué representan estas imágenes para ustedes?”, con la intención de que se genere un debate acerca de la evolución de la visión del futuro a lo largo del tiempo y su relación con la IA.

Se discutió cómo en el pasado se imaginaban inventos como autos voladores que aún no se han materializado, y se comparó con el presente, cuestionando hasta dónde puede llegar la IA y en qué ámbitos puede ser aplicada.

Para concluir con el taller, se abordó el tema “La IA en nuestros celulares”, destacando la facilidad con la que ahora se pueden desarrollar modelos de IA capaces de funcionar eficazmente en dispositivos móviles, igual que los modelos más grandes. Se discutió cómo esta tendencia hacia la portabilidad de la IA podría transformar diversas áreas tecnológicas e industriales. Para ilustrar esta evolución, se presentaron ejemplos como Apple Intelligence, que muestra cómo Apple está integrando inteligencia artificial en sus teléfonos de gama alta, y Galaxy AI, que destaca la incorporación de IA en dispositivos móviles de Samsung.

Además, se discutió el papel que desempeñamos como usuarios de dispositivos móviles. Aunque en la actualidad la IA está presente en dispositivos de gama alta, no siempre tenemos control sobre su inclusión, se anticipa que en el futuro esta tecnología se volverá común en la mayoría de los teléfonos móviles.

## 2.4. Documentación de la Experiencia

A continuación describen los instrumentos de recolección de información utilizados para documentar la experiencia:

- **Nube de palabras en Mentimeter:** se utilizó Mentimeter para recabar información sobre los conocimientos previos de los estudiantes sobre IA y los contextos en los que la IA se aplica. Esta nube de palabras ofreció una visión general de los conceptos y aplicaciones que los participantes ya conocían.
- **Observaciones de clases:** se realizaron observaciones durante los debates y la creación de modelos de aprendizaje por parte de los estudiantes. Estas observaciones se incluyeron en la descripción de la experiencia del taller para proporcionar una visión detallada de la participación y el aprendizaje de los alumnos.
- **Encuestas a estudiantes y docentes:** al finalizar el taller, se solicitó a estudiantes y docentes que completaran un formulario de Google para evaluar su opinión sobre el taller. La encuesta fue respondida por dos docentes y doce estudiantes.

## 3. Resultados

La nube de palabras resultante del Mentimeter mostró que los alumnos estaban bastante familiarizados con el concepto de IA y entendían su potencia.

Los debates generados en las distintas actividades fueron muy enriquecedores, permitiendo a los estudiantes y docentes compartir sus experiencias y perspectivas. Los docentes, en particular, aportaron valiosas opiniones sobre la enseñanza y la ética de la IA. Uno de los temas destacados fue el debate sobre el uso de la IA en las escuelas. Se discutió si la IA debería utilizarse para realizar tareas, crear material para clases o como fuente de información. Tanto profesores como estudiantes compartieron sus puntos de vista, y la idea predominante fue mantener una actitud crítica hacia los contenidos generados por la IA. Se plantea que los docentes podrían considerar fomentar el pensamiento crítico y evitar restringir el uso de estas herramientas, con el fin de potencializar el aprendizaje.

En cuanto a la integración de los contenidos del taller en el aula, la encuesta a los docentes reveló un interés variado. Un docente expresó interés de incorporar el reconocimiento automático de imágenes, mientras que otro consideró útil MIT App Inventor debido a que los estudiantes están familiarizados con esta herramienta. Ambos coincidieron en la importancia de abordar los conflictos éticos relacionados con el uso de IA.

Las actividades del taller fueron bien recibidas por los estudiantes, quienes mostraron gran entusiasmo y motivación. La posibilidad de interactuar de manera sencilla con las herramientas elegidas fomentó una participación activa en

cada sesión. La encuesta final mostró que el 66% de los estudiantes utiliza ChatGPT para tareas escolares. Entre las actividades, el "Bananómetro" fue la más popular, con un 66,7% de preferencia, seguido por "Crear mi propio modelo" e "Incorporar el modelo de GTM a MIT App Inventor," con un 58,3% de preferencias cada una. Estos resultados, ilustrados en el gráfico de la Figura 5, sugieren que el uso de bananas reales para entrenar modelos de IA capturó especialmente el interés de los estudiantes.

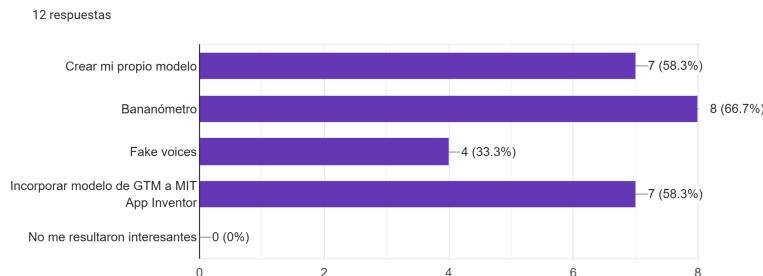


Figura 5: Respuesta de los estudiantes a la pregunta: “¿Qué desafíos te resultaron más interesantes?”

Todos los estudiantes que completaron la encuesta manifestaron interés por el tema y la experiencia. Las escuelas participantes han solicitado una nueva edición del taller para incluir a estudiantes de otras divisiones. Esta nueva sesión está programada para finales de agosto.

Como aspecto a mejorar, identificamos la necesidad de documentar con mayor detalle los debates sobre las implicancias éticas de la IA. Para las próximas ediciones, estamos considerando incorporar una matriz ética que facilite la documentación de las discusiones, ofreciendo un instrumento más estructurado para abordar estos temas.

## 4. Conclusiones

Las observaciones realizadas durante el taller revelaron que, aunque los estudiantes están relativamente familiarizados con el concepto general de IA y reconocen algunas de sus aplicaciones, los detalles sobre cómo construir y probar algoritmos de ML, así como la importancia de la interpretación de los modelos, resultaron ser temas nuevos y de gran interés para ellos. El uso de GTM facilitó la creación, entrenamiento y testeo de modelos de clasificación de imágenes, introduciendo a los estudiantes a conceptos como "clases" y "tamaño de la muestra de datos". Al integrar estos modelos en aplicaciones móviles mediante MIT App Inventor, los estudiantes pudieron ver cómo sus modelos podrían ser utilizados en aplicaciones, lo que hizo que la experiencia fuera significativa y relevante.

En cuanto al tiempo asignado para el desarrollo del taller, en general resultó adecuado, aunque sería conveniente ajustar la duración de algunos desafíos del segundo día. Además, se identificó la necesidad de desarrollar instrumentos didácticos que permitan evaluar de manera objetiva la dimensión ética de los modelos creados con las herramientas empleadas. Este será un aspecto clave a abordar en futuras ediciones del taller.

Se resaltó la importancia de entender que ningún modelo de IA es completamente libre de sesgos, ya que estos dependen de los datos con los que fueron entrenados. Los estudiantes tomaron conciencia de que no se debe confiar ciegamente en los resultados de los modelos de IA y que es fundamental hacer un uso responsable y crítico de estas herramientas.

## Referencias

- [1] “Convocatoria a Proyectos de Extensión “Universidad, Cultura y territorio” 2022”. Argentina.gob.ar. [Online]. Disponible: <https://www.argentina.gob.ar/educacion/universidades/fortalecimiento-de-las-trayectorias-estudiantiles/convocatoria-proyectos-0>
- [2] “El aula taller como estrategia pedagógica.” DGCyE Pcia Bs As. Comunicación N° 5 (2016) [Online]. Disponible: <https://abc.gob.ar/secretarías/sites/default/files/2021-08/Comunicaci%C3%B3n%20N%20C%20B0%205%20%282016%29%20-%20El%20aula%20taller%20como%20estrategia%20pedag%C3%B3gica.pdf>
- [3] “Teachable Machine Tutorial: Bananameter”. B. Warron. (2019, Nov 7). [Online]. Disponible: <https://medium.com/@warronbeebster/teachable-machine-tutorial-bananameter-4bffffa765866>
- [4] VAL KILMER y su nueva voz generada con INTELIGENCIA ARTIFICIAL. DODO Island (2021). [Video]. YouTube. [https://www.youtube.com/watch?v=Fahwy\\_sBPtQ&t=799s](https://www.youtube.com/watch?v=Fahwy_sBPtQ&t=799s) <https://www.youtube.com/watch?v=m7u-y9oqUSw>.
- [5] Jay-Z raps the "To Be, Or Not To Be" soliloquy from Hamlet (Speech Synthesis). Vocal Synthesis. (2020). [Video]. YouTube. [Online]. Disponible: <https://www.youtube.com/watch?v=m7u-y9oqUSw>.

# **Experiencia y aprendizajes del dictado del curso “La ciudadanía en un mundo atravesado por computadoras” para docentes del nivel primario**

Ana Casali<sup>1,2</sup>, Natalia Colussi<sup>1</sup>, Hernán Galardi<sup>1</sup>, Claudia Deco<sup>1</sup>, Cristina Bender<sup>1</sup>

<sup>1</sup>Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, Rosario, Argentina

<sup>2</sup>CIFASIS (CONICET-UNR), Rosario, Argentina

{acasali, colussi, galardi, deco, bender}@fceia.unr.edu.ar

## **Resumen**

Este trabajo presenta un análisis de resultados y aprendizajes que nos dejó la experiencia del primer dictado del curso: “La Ciudadanía en un mundo atravesado por Computadoras” para capacitar a docentes fundamentalmente del nivel primario. En este curso se trabajaron los conceptos de: Ciudadanía Digital, Ciencias de la Computación y TICs, Arquitectura del Ordenador, Redes e Internet, y Datos e Inteligencia Artificial. Los contenidos y recursos para el dictado del curso fueron desarrollados por la Fundación Sadosky, y dictado por un equipo de docentes de la Licenciatura en Ciencias de la Computación de la UNR quienes se encargaron de realizar las adaptaciones necesarias para el nivel educativo primario, destacando la incorporación de actividades prácticas que los maestros/as puedan llevar concretamente a sus aulas. El curso tuvo gran aceptación por parte de los cursantes, quienes expresaron mucho interés en la temática, especialmente en lo referente a las actividades de Inteligencia Artificial. A partir de los prácticos y autoevaluaciones realizadas, se evidencia la necesidad de continuar con estos trayectos para afianzar la formación docente en estos temas así como también, la importancia de desarrollar más recursos didácticos para trabajar en este nivel educativo.

**Palabras claves:** Ciencias de la Computación; Nivel Primario; Capacitación Docente

## **1. Introducción**

La necesidad de introducir la enseñanza de las Ciencias de la Computación (CC) en toda la educación obligatoria, está reconocida por la mayoría de los países y tuvo algunos avances en nuestro país al declarar el Consejo Federal de Educación (CFE) en 2015 que en su Resolución 263/15 [1] 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. Luego, en 2018 el CFE sancionó los Núcleos de Aprendizajes Prioritarios de Educación Digital, Programación y Robótica (Resolución 343/18) [2], estableciendo un plazo de dos años para su incorporación en las currículas provinciales. En este escenario surgen dos líneas fundamentales de trabajo: una la reforma curricular de los distintos niveles educativos de cada provincia y otra, la necesaria formación docente para implementar esta reforma.

Parte del equipo docente del Departamento de Ciencias de la Computación (FCEIA-UNR) viene trabajando desde 2016 en el desarrollo de distintos trayectos de formación docente del nivel primario en CC. En primer lugar, en el período 2016-2017 se diseñó y desarrolló la “Especialización Docente de Nivel Superior en Didáctica de las Ciencias de la Computación: Aprendizaje y Enseñanza del Pensamiento Computacional y la Programación en el Nivel Primario” (400 hs) la cual fue aprobada por el Ministerio de Educación de la Provincia de Santa Fe en 2017, y se dictó una primera cohorte (2017-2019), en el ISFD N°36 “Mariano Moreno” de la ciudad de Rosario, arrojando resultados muy positivos con un diseño curricular innovador. Posteriormente, este equipo fue seleccionado en distintas Convocatorias de la Fundación Sadosky (Línea B) y mediante proyectos de colaboración entre dicha Fundación y la UNR se dictaron para docentes del nivel primario los cursos: “La Programación y su Didáctica I” (2021 y 2023), “La Programación y su Didáctica II” (2022) y recientemente, un nuevo curso “La ciudadanía en un mundo atravesado por computadoras” (2023). Este nuevo curso complementa lo trabajado en los cursos previos sobre programación y retoma algunos contenidos ya trabajados en la Especialización. Todos estos cursos fueron gestionados a través de la Escuela de Posgrado y Formación Contínua (FCEIA-UNR), se dictaron de forma gratuita para los docentes gracias al convenio con la Fundación Sadosky, con una alta demanda por parte de los maestros y docentes de Tecnología del nivel primario. Destacamos que, a través de la preparación y dictado de estos trayectos formativos, se ha conformado dentro del DCC un equipo de docentes capacitado para trabajar contenidos de CC con una didáctica adecuada para los docentes del nivel primario.

En este trabajo, se analizan los resultados y aprendizajes que dejó la experiencia del primer dictado del curso “La Ciudadanía en un Mundo Atravesado por Computadoras” (LCMAC) por parte de este equipo de docentes del Departamento de Ciencias de la Computación (FCEIA-UNR) con el fin de compartirlos con la comunidad científica. En el mencionado curso se abordaron los siguientes contenidos de CC:

- La importancia de los conocimientos de CC en la construcción completa de la ciudadanía: ¿Cuáles son las diferencias entre el pensamiento computacional, las CC, y el uso de las TICs?, ¿Por qué necesitamos entender cómo se produce la tecnología digital? [3,4]
- Arquitectura del Ordenador y Organización de las Computadoras: ¿Podemos construir el mejor celular para nosotros?, ¿Qué es la obsolescencia programada? La contaminación producida por la tecnología digital y su huella de carbono. ¿Puedo distinguir en qué objeto tecnológico hay una computadora y en cuál no? [3,5]
- Redes e Internet: La infraestructura de las redes de datos (internet) nacional y mundial. Jerarquías estructurales en el servicio de internet y cómo se organizan los servicios de internet. ¿Qué es la nube?, ¿Cómo se transmite la información a través de internet? [6,7,8]
- Datos e Inteligencia artificial: ¿Qué tan inteligente es la inteligencia artificial? Dónde ya usamos la inteligencia artificial. La soberanía de los datos que producimos a diario. ¿Qué es y cómo entrenar un modelo de inteligencia artificial?. Problemas de la Inteligencia Artificial: el sesgo de datos. La importancia de la regulación de la Inteligencia Artificial [5,10,11].

Los contenidos y recursos para el dictado del curso están disponibles en el Campus FCEIA, fueron desarrollados por la Fundación Sadosky y se tomaron como base para el desarrollo del trayecto. El equipo docente a cargo del dictado estaba conformado por una docente coordinadora, una docente para el dictado del curso, y dos docentes para el desarrollo de las prácticas y corrección de las entregas. Cada uno de ellos realizó la capacitación correspondiente brindada por la misma Fundación, con un total de 20 horas y la entrega de un trabajo práctico final. El propósito fundamental que se persiguió con el dictado de este curso fue que los/as docentes puedan generar y liderar procesos de reflexión sobre el impacto que estas tecnologías producen en nuestra vida cotidiana y su relación con el ejercicio de una ciudadanía plena por parte de cada persona. En particular destacamos los siguientes objetivos:

- Conocer los distintos componentes de una computadora e identificar en diferentes situaciones de uso componentes de hardware y software y cómo se relacionan éstos entre sí.
- Reconocer que las actividades que se realizan en Internet son posibles tanto gracias a una infraestructura que habilita la circulación de información como a la existencia de computadoras que la intercambian y procesan.
- Reconocer la importancia de los datos para relativizar el costo de los programas o aplicaciones gratuitas, en cuanto al uso de datos de los usuarios y usuarias.
- Reconocer distintos usos de los algoritmos de inteligencia artificial en la computación, su aplicación en situaciones cotidianas y sus implicancias éticas.

## 2. Descripción de la Experiencia

### 2.1. Modalidad de dictado y cursantes

El Curso “La ciudadanía en un mundo atravesado por computadoras” ha sido dictado en la sede de la Facultad de Ciencias Exactas, Ingeniería y Agrimensura, UNR, entre el 14/09/2023 al 6/12/23. La duración del curso fue de 12 semanas con modalidad mixta: 1 clase virtual semanal y 4 encuentros presenciales. La asistencia debía ser del 75% tanto en los encuentros virtuales como presenciales. Las clases virtuales sincrónicas se dictaron los días jueves de 18:30 a 20:30 hs, y las clases presenciales se dictaron en 4 sábados de 9 a 13 hs en la sede de la FCEIA. Se brindó a los cursantes, la disponibilidad de horarios de consultas semanales. El curso se aprobó mediante la entrega de Trabajo Prácticos. La carga horaria total de 60 horas, incluyó lecturas, prácticas y elaboración del trabajo final. Todo el curso se desarrolló con el soporte del aula virtual correspondiente en el Campus Virtual de la Facultad. El curso otorga puntaje por la Ordenanza Decreto Prov. D0302912 del Ministerio de Educación Santa Fe.

El enfoque didáctico de los distintos ejes temáticos del curso se realizó mediante la metodología de aprendizaje por indagación. Esta didáctica interpela a los/as cursantes a que sean ellos/as mismos/as los/as que encuentren soluciones a un problema a partir de un proceso de investigación. A través de esta exploración descubren conceptos o ideas subyacentes necesarias para resolver un desafío en busca de una solución. Este tipo de trabajo potencia las habilidades resolutivas, el trabajo en equipo y la cooperación, junto al pensamiento crítico. Para guiar este proceso, el docente realiza preguntas, otorga tiempo de resolución, se proponen hipótesis, se validan o se refutan, se busca bibliografía, se proponen respuestas, predicciones, y se argumenta, con una reflexión final que permite afianzar lo aprendido. Esta metodología es ampliamente utilizada en el nivel primario y en el curso fueron los mismos docentes los que descubrieron conceptos e ideas necesarias en la problematización de los conceptos de CC abordados. De esta forma, mediante las actividades adaptadas al nivel primario, se guió a los docentes en el descubrimiento de aquello que en un principio podía resultar difícil de pensar y de imaginar cómo funcionaba. Entonces, sólo para citar un ejemplo, cuando abordamos el problema de internet y su infraestructura, jugamos con los docentes llevando adelante la secuencia didáctica: “El viaje de la información por internet, hacia una ciudadanía digital informada” extraída del Repositorio Curricular de Programar. Dónde, mediante preguntas y escenarios de casos o situaciones puntuales descriptos en el pizarrón, fuimos resolviendo y entendiendo todos juntos

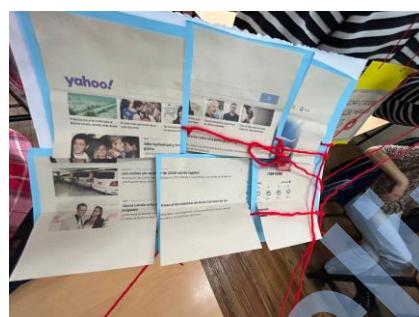
distintos problemas específicos de las redes, como por ejemplo: ¿Cómo se construye una red?, ¿Qué sucede cuando un router está fuera de servicio o saturado?, ¿Cómo viaja la información en internet? Partes de un mensaje y direcciones ip, etc. Las imágenes mostradas en las Fig 1, 2, y 3 registran la actividad y muestran cómo se llevó adelante.



**Fig. 1:** Con lana de color trazamos el recorrido de los mensajes donde los docentes jugaron a ser routers de internet. Adelante se ubicaba la PC (docente elegida) para enviar los mensajes (representados por hilos)



**Fig. 2:** Completando la visión de la Fig. 1, al final del salón, se ubicaba la PC que recibía los mensajes (fragmentos) y armaba las partes. Cada hilo representaba el recorrido de un mensaje en la red.



**Fig. 3:** Todos los fragmentos recibidos, en distinto orden, con sus respectivos hilos, pegados con cinta de papel, sobre una cartulina a modo de rompecabeza para “dibujar” en toda la composición completa, una página web.

El curso se ofreció a docentes del nivel primario, fundamentalmente del área de educación tecnológica, que ya hubieran hecho alguna capacitación previa en CC. Es decir, docentes que hubieran completado la “Especialización en didáctica de las CC”, el Curso “La Programación y su didáctica I” u otras formaciones similares. El número de docentes inscriptos al curso fue de 74. De los inscriptos 17 no se presentaron a ningún encuentro, 16 dejaron el curso luego de asistir a 1 o máximo 3 clases, teniendo una deserción temprana alta (28%). Si bien no registramos información sobre los principales motivos, podemos considerar que la alta demanda laboral de las/os docentes, situaciones personales y la demanda del curso en cuanto asistencia y dedicación, pueden ser algunos de los factores preponderantes. De los restantes 41, la mayoría completó y aprobó el curso. El 51% de los que lo aprobaron, realizan su actividad docente en el área de Educación Tecnológica o Informática/Computación.

A modo de presentación del grupo se solicitó a los cursantes que llenaran asincrónicamente un muro digital (<https://padlet.com/>) donde los docentes indicaron su formación, lugares donde trabajaban, intereses y expectativas frente al cursado. Frente a la pregunta ¿cuáles eran las expectativas que tenían del curso?, la mayoría señaló primero “adquirir conocimientos y capacitación”, conjuntamente con “aprender sobre nuevas herramientas y recursos didácticos”, y en menor medida se indicó la “actualización en el área” como respuesta. De estas apreciaciones y tras la experiencia del dictado, señalamos que la mayoría de los/as docentes buscó adquirir saberes y herramientas nuevos, y pocos/as buscaban actualizarse a partir de conocimientos previos. La mayoría de los/as docentes no conocía los temas abordados, o tenía preconceptos erróneos. Además, la realización conjunta de actividades les permite reproducirlas en el aula tal cual la aprendieron, pero también, les permite construir otras a partir de estas experiencias. Por eso, es tan importante brindarles recursos que puedan aplicar en el aula. Todo esto marca la importancia que tiene brindar trayectos formativos a docentes sobre estos temas, que resultan necesarios en la educación digital actual.

## 2.2. Adaptaciones necesarias del curso para el nivel primario

El curso fue diseñado por la Fundación Sadosky y está orientado a la educación obligatoria del nivel secundario, dado que la mayoría de las UUNN vienen trabajando en ese nivel. Dicho trayecto, contiene una buena selección de bibliografía para cada una de las unidades en las que se divide el curso, para que en los encuentros se trabaje sobre la discusión de los ejes temáticos de cada unidad junto al desarrollo de trabajos prácticos sobre los temas vistos. La propuesta que llevamos adelante en la FCEIA-UNR está destinada a los docentes del nivel primario, con lo cual se debieron realizar adaptaciones al diseño original del curso para que el mismo pudiese ser trabajado con los/as docentes del nivel primario y que ellos a su vez, pudiesen abordar los temas en el aula.

Una de las primeras medidas realizadas fue preseleccionar el material de lectura de cada unidad, distinguiendo aquellos aspectos que podíamos abordar y aquellos que podíamos dejar de lado. Cada lectura era trabajada con los docentes en los encuentros virtuales síncronos de manera grupal, explicando detalles que podían complejizar el entendimiento del tema, recorriendo en conjunto los contenidos del aula virtual, para que luego los/as cursantes pudiesen reelaborarlos en los trabajos que realizaban en grupo. Estos trabajos eran expuestos a todos los presentes, con la respectiva devolución de los docentes a cargo. Algunos de estos cierres se llevaron a cabo durante la misma clase y en otros casos se hacían en la siguiente, dándoles más tiempo para profundizar temas antes de exponerlos. Los grupos de trabajo se establecieron al inicio de la cursada y se mantuvieron hasta el final. Las/os maestras/os en general, no cuentan con mucho tiempo de lectura fuera de los horarios del cursado, por lo cual era importante trabajar los contenidos principales durante el tiempo ya asignado. Esta etapa de trabajo grupal resultó central para afianzar contenidos y detectar dificultades en temas que no eran comprendidos correctamente y poder corregirlos rápidamente.

En este nivel educativo es necesario trabajar partiendo de un marco teórico, pero con mayor cantidad de actividades prácticas que den lugar a espacios de discusión, materializando así los conceptos teóricos con trabajos prácticos. Esta incorporación de actividades concretas fueron las principales adaptaciones que se hicieron a todo el material, donde las mismas se sumaron tanto para el abordaje de un tema, como para el cierre reflexivo de los contenidos de los ejes temáticos. Se problematizó con actividades lúdicas, juegos interactivos y formas narrativas gráficas como: comics, graffitis, flyers, etc.. Estas últimas herramientas descriptivas permitieron plasmar la discusión y abrir el debate de aquellos aspectos de índole social y ciudadano, habilitando así el ejercicio pleno de la ciudadanía, sumando la conciencia social tecnológica a la ya tradicional que se construye tempranamente en el aula. La Tabla 1 muestra las principales actividades desarrolladas en el curso y las Fig. 4, 5 y 6 ilustran algunas de las producciones de los docentes.

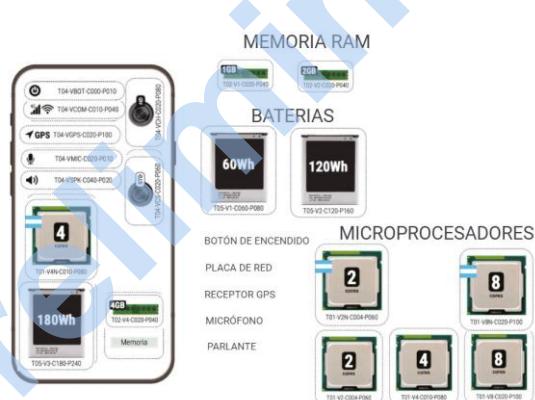
**Tabla 1:** Actividades prácticas adicionadas por eje temático para adecuar el curso a docentes del nivel primario.

Ejes Temáticos	Actividades Desarrolladas
<b>Arquitectura y Organización del Computador</b>	<ul style="list-style-type: none"> <li>Trabajamos con las siguientes actividades de los Manuales de Ciencia de la Computación para el Aula para el primer y segundo ciclo del primario.             <ul style="list-style-type: none"> <li>○ Computadoras por todos lados</li> <li>○ Las Partes de toda Computadora</li> <li>○ Una nueva máquina</li> </ul> </li> <li>Se elaboró una trivia digital usando slides de google para identificar claramente los dispositivos tecnológicos que incluyen una computadora y los que no la contienen.</li> <li>La actividad de confeccionar un mejor celular según los perfiles de usuarios se realizó en grupo empleando la plataforma Jamboard, donde se distribuyeron los distintos perfiles por grupo, se les entregó las componentes del celular ya recortadas digitalmente, por lo cual podían moverlas y elegirlas por separado. Luego, justificaron en la misma plataforma cada elección realizada.</li> <li>El cierre de este eje, giró en torno a las problemáticas sociales vinculadas a la obsolescencia programada y el impacto ecológico de los recursos tecnológicos, el uso de los recursos minerales y su escasez mundial sumado al conocido “derecho a reparar” los dispositivos y no tener que desechar. Éstos temas fueron abordados mediante la elaboración de un comic por grupo, en la plataforma canvas, el cual luego expusieron ante todos los cursantes.</li> </ul>
<b>Redes de computadora e Internet</b>	<ul style="list-style-type: none"> <li>Este tema se trabajó realizando en clase la secuencia didáctica: "El viaje de la información por internet, hacia una ciudadanía digital informada" extraída del Repositorio de Programar.</li> <li>Se utilizó la actividad "Búsqueda del tesoro: ¿Por dónde viaja la información en Internet?" del Repositorio Curricular de Programar, la cual se llevó a una versión de juego digital usando la plataforma Kahoot. Disponible en: <a href="https://create.kahoot.it/share/lcmac-internet/969396e6-cc4b-4266-932c-a857489bb337">https://create.kahoot.it/share/lcmac-internet/969396e6-cc4b-4266-932c-a857489bb337</a> la cual permitió ahondar sobre los aspectos técnicos relacionados a la infraestructura de internet y sociales, dada la dependencia actual de este servicio para realizar transacciones bancarias y compras, sólo por citar algunas de las discusiones problematizadas.</li> <li>El tema se cerró con reflexiones expresadas mediante graffitis que cada grupo elaboró según los documentos que se les asignó. Así se expresaron sobre los problemas referidos a La Nube, Las búsquedas en la web, Netflix y el streaming, BitCoins y el daño ambiental.</li> </ul>
<b>Inteligencia Artificial</b>	<ul style="list-style-type: none"> <li>Se utilizó la ficha didáctica: ¿Qué necesita la IA para ser IA? extraída del Repositorio Curricular de Programar. Donde se realizaron:             <ul style="list-style-type: none"> <li>Actividad 1: Distinguir que aplicaciones que utilizamos diariamente ya emplean IA para brindarnos algún servicio.</li> <li>Actividad 2: Trabajamos con el predictor de palabras de la aplicación de mensajería whatsapp.</li> </ul> </li> </ul>

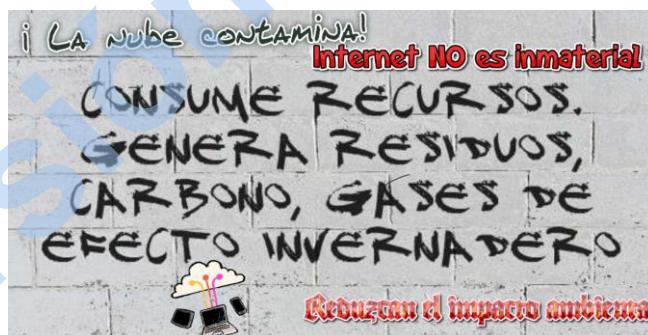
- Actividad 3 y 4: Fijamos los conceptos aprendidos, distinguimos la entrada de datos, la salida, y dónde está la inteligencia artificial. ¿Quienes ingresan los datos? y ¿quién produce la salida? Extrapolamos luego a otras plataformas o redes sociales, que funcionan de igual manera, pero recomendando películas, música, videos, etc. en lugar de palabras, destacando en esa oportunidad de que datos se nutren para recomendar.
- Se utilizó la plataforma <https://stretch3.github.io/> (versión extendida de Scratch que admite librerías de IA de aprendizaje automatizado) en conjunto con la plataforma <https://teachablemachine.withgoogle.com> para generar un recomendador de actividades de aseo personal, identificando un peine de una cepillo de dientes, y con ello distinguir si la persona se dirigía a lavarse los dientes o a peinarse. Esto permitió poner en valor las notebook del plan conectar igualdad, las cuales cuentan con cámaras webs integradas, con lo cual podrían pensarse y realizarse proyectos similares con los estudiantes en las escuelas.
- Se agregó una actividad para conocer la plataforma ChatGPT, y explicar mediante consultas relacionadas cómo funcionaban las IA generativas. Este trabajo incluyó la exploración de la plataforma, la creación de varios prompts, para posteriormente comparar las respuestas generadas, contrastar diferencias y detectar los patrones entre consultas similares. Este trabajo se basó en la Jornada 1 del curso “Inteligencia artificial, internet y microchips”, del INFOD.



**Fig. 4:** Cómic producido por uno de los equipos para concientizar sobre la obsolescencia programada



**Fig. 5:** Una resolución de la actividad armado del “mejor celular”, seleccionando sus partes según el perfil del usuario (en Jamboard).



**Fig. 6:** Graffiti realizado por uno de los equipos docentes, para concientizar sobre el impacto ambiental de Internet.

### 2.3 Algunas dificultades encontradas

Como cierre de la Unidad 1: La importancia de los conocimientos de CC en la construcción completa de la ciudadanía, se solicitó a los cursantes que realizaran el Autotest 1 (<https://forms.gle/3tNoVFbYLXU14tX8>), para evaluar los conceptos aprendidos y opiniones sobre la relevancia de incorporar CC en la escolaridad obligatoria. De los cursantes, hubo 15 docentes que lo respondieron más de una vez: 9 personas lo contestaron 2 veces y 6 personas lo contestaron 3 veces. Se realizó una vez desarrollada la primera unidad, entre octubre y diciembre de 2023, y se repitió entre febrero y marzo de 2024. Los puntajes obtenidos en la primera respuesta y las últimas han sido fluctuantes, en su mayoría con puntaje creciente. Como se puede observar de las respuestas a la Pregunta 1, para el docente del nivel primario el término “Pensamiento Computacional” está muy vinculado a las CC y más de la mitad, lo usarían para identificar esta área fundamental para el desarrollo del mundo tecnológico. Si bien en el curso se focalizó este área con la denominación de las Ciencias de la Computación, explicando su historia, alcances y áreas, discutiendo también posturas como la presentada por Fernando

Schapachnik en “Despejando la ensalada epistemológica que la Informática trajo a la Educación” ([https://www.youtube.com/watch?v=XOVX28J\\_JzU](https://www.youtube.com/watch?v=XOVX28J_JzU)) respecto al término Pensamiento Computacional, los/as cursantes tienen a esta última denominación más internalizada. Una hipótesis de los autores es que al docente del nivel primario, le es más cercano el término de Pensamiento Computacional que el de CC y por eso lo eligen. Respecto a las respuestas de la Pregunta 2, está claro para este grupo que el área de CC es más amplio que la programación e identifica a la disciplina que abarca a la tecnología distintiva de este siglo. La Pregunta 3 tuvo un 70% de respuestas correctas y se basaba en la postura de una fuente (ICILS), se puede considerar que para contestar esta pregunta, no hayan tenido presente a esta fuente, ya que algunos realizaron el autotest varios meses después. La Pregunta 4 evidencia que todavía la diferencia entre uso de TICs y aprendizaje de CC para algunos docentes no es del todo clara y hay que seguir trabajando al respecto. También se podría reformular la pregunta, para establecer como opción que “consiste principalmente en promover el desarrollo de habilidades de uso de sistemas digitales en diferentes áreas del conocimiento”. Respecto a la Pregunta 5 hay un 30% que contestó de forma adecuada, un 50% considera a la Robótica como un área dentro de CC (lo cual podría contemplarse según muchas fuentes). Los que incluyeron en sus respuestas la opción TICs (34%) siguen manifestando dificultad para diferenciar el uso de aplicaciones, con lo que es CC.

En general se puede observar que este Autotest 1 les planteó problemas para realizarlo correctamente y varios docentes tuvieron que hacerlo varias veces para superar un 4 (2 preguntas bien). Se sugiere para próximas ediciones de este curso, revisar las preguntas de modo que queden más claras, evitar las que hagan hincapié en una fuente de información específica y agregar otras preguntas a fin de apuntalar más las diferencias entre TICs y CC, y las áreas que CC abarca.

El Autotest 2 (<https://forms.gle/qc8AfSjwzBEex7gR8>) se propuso como actividad luego de completar la segunda unidad del curso: ¿Eso también es una computadora? Este análisis se realiza sobre las respuestas de los 38 alumnos que completaron el cursado.

- Pregunta 1: Cualquier dispositivo que cuente con teclado y monitor es o tiene una computadora. El 82% contestó acertadamente eligiendo la opción Falso.
- Pregunta 2: La memoria RAM... con las opciones Almacena los programas que están en ejecución, Es el cerebro de la computadora, Ninguna es correcta, Almacena todos los datos (números, textos, imágenes, películas) que tenemos en la computadora, Almacena los datos (números, textos, imágenes, películas) que están siendo usados en el momento; siendo la primera y la última las opciones correctas. Aquí, el 63% de las y los cursantes respondió adecuadamente sólo las dos opciones correctas. El resto de las respuestas tienen 1 a 3 opciones elegidas, en las cuales se incluye al menos una de las opciones correctas. En forma individual, las dos opciones más elegidas fueron Almacena los programas que están en ejecución (100%) y Almacena los datos que están siendo usados en el momento (82%).
- Pregunta 3: La CPU ... con las opciones Hace aritmética básica, Recupera datos de la memoria para procesarlos y almacena el resultado del procesamiento, Es el cerebro de la computadora, Ninguna es correcta, Ejecuta las instrucciones de los programas; siendo Hace aritmética básica, Recupera datos de la memoria para procesarlos y almacena el resultado del procesamiento, y Ejecuta las instrucciones de los programas las opciones correctas. En esta pregunta, el 29% de los cursantes eligieron adecuadamente las 3 opciones correctas; 6 asistentes (16%) eligieron dos de las opciones correctas, sin incluir ninguna otra opción. En forma individual, las opciones más elegidas fueron Recupera datos de la memoria para procesarlos y almacena el resultado del procesamiento (82%), Ejecuta las instrucciones de los programas (79%) y Hace aritmética básica (66%). Como dato llamativo, la opción Es el cerebro de la computadora, fue elegida por casi el 45% de los asistentes como una de las opciones seleccionadas.
- Pregunta 4: La memoria RAM y otros dispositivos de almacenamiento como disco duro, pendrives o CDs, cumplen la misma función en una computadora. El 84% de las y los cursantes eligieron la opción correcta Falso.
- Pregunta 5: Una consola de videojuegos, como la playstation o la nintendo switch, ¿tiene una computadora? Aquí el 97% de los docentes eligieron la opción Sí que es la correcta.
- Pregunta 6: Se considera que la primera persona en escribir un programa fue... con las opciones Margaret Hamilton, Ada Lovelace, Alan Turing y Charles Babbage. El 95% de los asistentes eligieron la opción correcta: Ada Lovelace.

En general se observa que este Autotest les resultó más fácil, obteniendo un promedio de 8 en las respuestas recibidas. La pregunta 3 sobre las acciones de la CPU, es la que presentó mayor dificultad. El considerar a la CPU el cerebro de la computadora, por un 45% puede ser una metáfora que utilizan los maestros del nivel primario para indicar que es donde la computadora realiza el procesamiento de la información. En esta unidad se trabajaron cuestiones más concretas de arquitectura del computador que permitieron una mejor asimilación por parte de los y las docentes cursantes.

### 3. Resultados de la Experiencia

El curso se dictó satisfactoriamente, siguiendo el cronograma y contenidos establecidos según el programa, se ha dado soporte al curso mediante el aula virtual en el Campus de la Facultad, donde se presentaron todos los recursos educativos y las actividades prácticas. En los distintos encuentros se trabajó por indagación los distintos temas abordados, haciendo énfasis en los cierres reflexivos. Un total de 37 docentes lograron finalizar el curso con éxito.

### 3.1 Trabajos finales:

Los/as cursantes debían presentar un trabajo final grupal, elaborando un taller basado en la elección de uno de los temas tratados en el curso. Se asignó un docente para la tutoría de cada grupo, para acompañarlos con el desarrollo de cada propuesta. Se destinó una clase virtual para pensar el tema y la idea general del taller. Luego, en base a un documento guía, con la supervisión de los docentes del curso, cada grupo fue completando el desarrollo de su taller. En la última clase presencial se presentaron las ideas y avances de cada trabajo. Se presentaron un total de 10 propuestas de talleres para aulas del nivel primario y primer ciclo del secundario. Muchas de las propuestas fueron derivaciones de trabajos previamente realizados durante el curso, con improntas propias del grupo o temáticas del grado al cual estaban dirigidos. El 70% trabajó la arquitectura del ordenador, junto a la problemática de la obsolescencia programada y el reconocimiento de la arquitectura Von Neumann en los dispositivos tecnológicos. El 20% se refirió a redes e internet, trabajando su infraestructura y cómo se envían los mensajes por la red. Sólo el 10% (1 grupo) eligió el tema de inteligencia artificial para trabajar en el aula. La Figura 7 muestra las propuestas y un breve resumen de cada una.

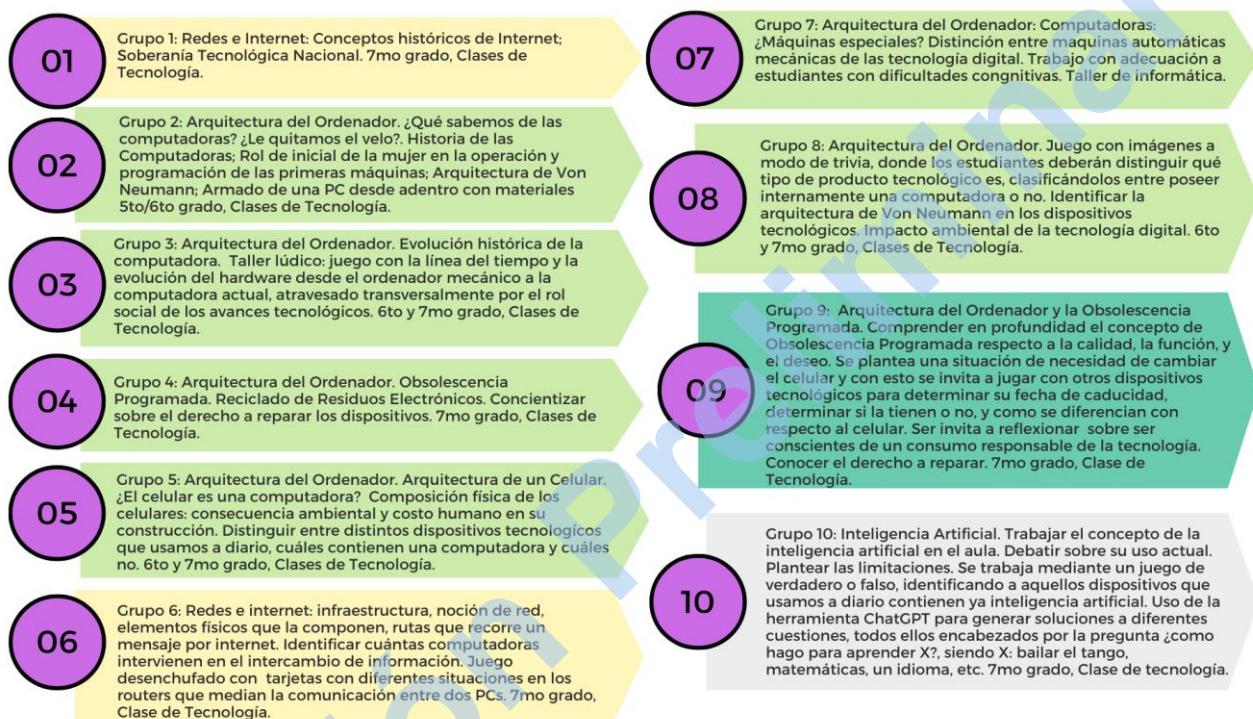


Fig. 7: Resumen de los trabajos finales del curso.

La evaluación de los talleres se realizó en base a una rúbrica ofrecida por los diseñadores del curso de la Fundación Sadosky. El promedio de las calificaciones fue 8.35 (escala 1-10) estando todos los trabajos muy bien realizados (accesibles en: <https://drive.google.com/drive/folders/1MVt0ybP3ahUfJ5qIIRYixlv3j5fALw4?usp=sharing>).

### 3.2 Encuesta final del cursado:

A fin de recabar percepciones y opiniones de los cursantes, sobre distintos aspectos del dictado del curso y sobre los temas abordados, se aplicó una encuesta diseñada por la Fundación Sadosky para este curso en particular. La misma se distribuyó vía el Campus Virtual al final del dictado del curso (diciembre 2023), obteniendo 32 respuestas de las/os cursantes. Todas las preguntas cuantitativas fueron consideradas utilizando una escala de Likert (1-5) donde en cada caso se indicó que significaban los extremos de la escala. A continuación se realiza un análisis cuanti-cualitativo de las respuestas recolectadas.

#### Preguntas generales sobre valoración y dificultades encontradas en el curso

El primer grupo de preguntas corresponden a sus percepciones respecto al desarrollo del curso y posibilidades de incorporación de sus contenidos en sus actividades áulicas. Los docentes refieren que pueden trasladar al aula un alto porcentaje de lo aprendido en el curso, obteniendo un promedio de 77%. En cuanto a las mayores dificultades para llevar a cabo las actividades sugeridas en este curso, de las opciones provistas, casi el 50% seleccionó la dedicación horaria, y en segundo lugar, un 25%, seleccionó el trabajo en grupo. Estas respuestas apuntaron a problemas ajenos al curso, vinculados a las actividades laborales de los asistentes. También, algunos cursantes (15%) expresaron que tuvieron problemas con la cantidad y la complejidad del material de lectura. Para la consulta ¿Qué tan satisfactorio fue para usted el curso? el puntaje promedio fue alto (4,7) lo que señala que a los cursantes les ha resultado muy satisfactorio el curso. Tanto para los

Conceptos abordados como para Actividades planteadas y Lecturas y recursos utilizados los asistentes los consideraron de complejidad media. Con respecto a la Originalidad de los materiales fueron considerados de bastante a muy originales, con un promedio de 4,10. Con respecto a la Dificultad del curso, el promedio de valores fue 3,03, lo cual indica que el curso fue considerado de complejidad adecuada. Para la consulta ¿Qué hubiera hecho diferente respecto al desarrollo del curso?, las respuestas en texto libre más presentadas, con un 50%, se refirieron a una gran conformidad con los contenidos y la posibilidad de su aplicación en el Aula. En porcentajes mucho más bajos, las respuestas fueron: Más clases presenciales (9%), Dar más app, videos y herramientas para utilizar en el aula (9%), y Ampliar el tema de Inteligencia Artificial (6%). En las consultas “Quisiera aprender más conceptos y contenidos de Ciencias de la Computación” y “Quisiera aprender más sobre la enseñanza de Ciencias de la Computación”, los asistentes estuvieron Muy de acuerdo con esta afirmación.

En las siguientes Tablas 2, 3, 4, 5 y 6 se analizan distintos módulos de preguntas sobre las unidades desarrolladas en el curso. Las consultas de opinión estaban referidas a expresar un número en la escala de Likert 1-5. Se muestran los promedios obtenidos en base a las respuestas de 32 cursantes:

**Tabla 2: Promedio de respuestas a las preguntas Sección “Las Ciencias de la Computación en la Escuela”**

Sección Las Ciencias de la Computación en la escuela - Preguntas	Promedio (1 Definitivamente no /5 Por supuesto que sí)
¿Piensa que su escuela debería incorporar la enseñanza de las Ciencias de la Computación?	4,88
Permite a mis estudiantes ser usuarios críticos de la tecnología que consumen	4,66
¿Piensa que es necesario contar con un espacio curricular propio para la enseñanza de las Ciencias de la Computación?	4,66
¿Se siente capacitado/a para enseñar Ciencias de la Computación?	3,19
¿Cree que en la mayoría de las escuelas donde trabaja están dadas las condiciones para incorporar la enseñanza de programación?	2,84

De los datos se desprende que, si bien el 96,88% de los docentes reconoce una necesidad imperiosa de incorporar las CC en sus escuelas, solo el 37,50% respondió sentirse capacitado para enseñar esta disciplina, y solo el 18,75% cree que en sus escuelas están dadas las condiciones para incorporar la enseñanza de programación.

**Tabla 3: Promedio de respuestas a las preguntas Sección “Las Ciencias de la Computación en la Escuela”**

Sección ¿Eso también es una computadora? - Preguntas	Promedio (1 Nada de acuerdo/5 Muy de acuerdo)
Si un dispositivo es una computadora entonces es necesario que tenga CPU	4,03
Si un dispositivo es una computadora entonces es necesario que tenga memoria RAM	4,06
Si un dispositivo recibe información de entrada y devuelve información como salida, entonces podemos afirmar que es una computadora.	3,44
Si un dispositivo es una computadora entonces es necesario que tenga teclado	1,88
Al momento de comprar una nueva computadora conviene elegir los componentes más potentes	2,69

Respecto a los datos recabados en la sección “¿Eso también es una computadora?”, resulta interesante notar que, si bien los docentes reconocen mayoritariamente la necesidad de que un dispositivo tenga un CPU y memoria RAM para ser una computadora, paralelamente también consideran que un dispositivo que reciba información de entrada, y devuelva información como salida, constituye una computadora.

**Tabla 4: Promedio de respuestas a las preguntas Sección “Redes e Internet”**

Sección Redes e Internet - Preguntas	Promedio (1 Nada de acuerdo/5 Muy de acuerdo)
Internet es una construcción colectiva, por lo que nadie es dueña/o de internet.	3,41
La mayoría de los mensajes que enviamos por internet en Sudamérica pasan por EEUU.	4,38
Siempre que usamos internet estamos consumiendo servicios de un grupo muy reducido de empresas.	3,97
Las computadoras que deciden por dónde circulan los mensajes en internet (routers o enruteadores) necesitan información de todas las computadoras de la red.	3,19
En internet, el mismo programa que se encarga de dirigir los mensajes hasta el destino correcto se encarga de que estos lleguen completos y sin errores.	3,34

Sobre la sección “Redes e Internet”, podemos notar una tendencia en los docentes cursantes a confundir algunas de las especificidades que conlleva el ruteo de los mensajes, y los protocolos de comunicación. Es posible que la falta de conocimientos previos en el área, y la orientación inicial del curso hacia el nivel secundario, requieran adaptaciones adicionales al nivel primario, incluso más allá de las introducidas en la presente cohorte.

**Tabla 5: Promedio de respuestas a las preguntas Sección “Inteligencia Artificial”**

Sección Inteligencia Artificial - Preguntas	Promedio (1 Nada de acuerdo/5 Muy de acuerdo)
Las decisiones o recomendaciones que realiza la Inteligencia Artificial no dependen de las personas	2,22
La IA puede usarse para tomar cualquier tipo de decisión	2,38
La aplicaciones que utilizan IA, no cometan errores	1,50
La IA, como está implementada en una computadora, no tiene opinión y por lo tanto no puede discriminar ni prejuzgar.	3,38
La IA resuelve los problemas mejor de que lo harían las personas	1,94

Resulta interesante notar que más allá del significativo tiempo dedicado durante el curso al análisis de los sesgos que pueden presentar los sistemas basados en IA, muchas veces debido a sus datos o diseño, se observa una dificultad de los cursantes para poder identificar posibles consecuencias de los mismos en un sistema implementado en una computadora.

**Tabla 6: Promedio de respuestas a las preguntas Sección “Impacto”**

Sección Impacto - Preguntas	Promedio (1 Nada de acuerdo/5 Muy de acuerdo)
Se renuevan los dispositivos muy seguido porque los desarrollos tecnológicos cambian todo el tiempo.	3,72
Los datos que almacena Google cuando utilizamos sus servicios gratuitos les dan más ganancia que lo que recauda por sus servicios pagos	4,13
Poder mirar películas por Netflix, guardar archivos en la nube, escuchar música en Spotify ahorra recursos naturales comparado a ir al cine o comprar DVDs o CDs, usar pendrives.	2,25
Que las compañías de celular nos regalen datos para el uso de WhatsApp es un problema	3,63
Está bien limitar qué cosas pueden realizar los sistemas basados en IA y cuáles deben ser sus alcances.	4,63

De este último grupo de preguntas de la encuesta, destacamos que, mayoritariamente, los docentes destacan la importancia de limitar los alcances de los sistemas computacionales basados en IA.

### 3.3 Encuentro final

En el último encuentro presencial de cierre del curso (ver foto en Fig. 8), se realizó un espacio para socializar las propuestas de Trabajos Prácticos, considerando que estos espacios son muy importantes y enriquecen los trabajos desarrollados con las opiniones y sugerencias de sus pares. También se trabajó con herramientas de IA debatiendo acerca de los grandes alcances que nos brindan hoy estos sistemas, pero también sus riesgos e implicancias sociales. Al finalizar, en un espacio de libre expresión, muchos docentes manifestaron lo interesante que les resultó el curso, la necesidad de aprender como docentes para encarar estos temas en el aula, y cuán importante es seguir capacitándose para estar más seguros en estos contenidos novedosos para ellos.



**Fig. 8:** Foto del grupo en el encuentro final.

## 4. Conclusiones

En los/as cursantes hubo mucho interés en las temáticas abordadas y en especial, en el trabajar distintas actividades que pueden llevar al aula del nivel primario, enfocando en el segundo ciclo de este nivel. A la propuesta de diseño inicial del curso que estaba pensado más para su desarrollo en el nivel secundario, le hemos realizado algunas adecuaciones necesarias para el nivel educativo, especialmente en lo que respecta a aumentar la cantidad de actividades prácticas y materializar algunas discusiones. De los/as inscriptos/as iniciales el 50% aprobó el curso (37 cursantes), este porcentaje es mayor si consideramos que 17 inscriptos no se presentaron a ningún encuentro luego, considerando 57 docentes que asistieron al menos a una clase, el porcentaje de aprobados es del 65%. Los/as docentes que han seguido el curso con participación activa en los encuentros sincrónicos no han presentado dificultad para aprobar el curso. Los factores por los que los/as cursantes han abandonado el curso son diversos y se deben entre otros, a distintos motivos personales y a problemas vinculados a la gran carga laboral que suelen tener los/as maestros/as. Los comentarios de los/as participantes en la jornada final de cierre, fueron muy positivos, tanto respecto a los temas abordados a los cuales ven como necesarios para capacitarse y llevarlos al aula, como en la modalidad de trabajo por indagación, con actividades que pueden trasladar a sus cursos. También, destacaron la forma mixta de dictado, con encuentros virtuales que les resultan más accesibles para participar, ya que muchos de los cursantes son de localidades aledañas a Rosario, y con encuentros presenciales, que permiten otro tipo de interacción con el grupo y generan comunidad. Respecto a los problemas de aprendizaje detectados, destacamos que hay que seguir trabajando en los distintos trayectos de este nivel, también seguir enfatizando las diferencias entre aprender y enseñar Ciencia de la Computación, respecto de TICs, donde la “comprensión” debe distinguirse más claramente del “uso” de las computadoras. El resto de los contenidos sobre: arquitectura del computador, redes e internet, e inteligencia artificial, para la mayoría ha sido novedoso y necesitan seguir afianzando estos conocimientos para trabajarlos con sus alumnos. El tema de Inteligencia Artificial es el que más los interpela y quieren tener conocimientos para entender su potencial, pero también sus riesgos e implicancias sociales. Como se ha mencionado, el curso necesitó adecuaciones para su dictado en el nivel primario y se considera que se podría extender la carga horaria de su dictado en una nueva edición, incluyendo más actividades prácticas en cada unidad.

## Referencias

- [1] CFE, ME, Argentina, Resolución 263-15, (2015) [https://cfe.educacion.gob.ar/resoluciones/res15/263-15\\_01.pdf](https://cfe.educacion.gob.ar/resoluciones/res15/263-15_01.pdf)
- [2] CFE, ME, “Núcleos de aprendizajes prioritarios: Educacion digital, programación y robótica”, (2018), <https://www.educ.ar/recursos/150123/nap-de-educacion-digital-programacion-y-robotica>
- [3] Gómez, M. J. (2020). Aspectos de adquisición de lenguaje en la enseñanza de programación [Tesis de doctorado, Universidad Nacional de Córdoba]. Repositorio Institucional. <https://www.researchgate.net/publication/345256073>
- [4] Ko, A. J.; Beitlers, A.; Wortzman, B. y otros (2022). Critically Conscious Computing: Methods for Secondary Education. <https://criticallyconsciouscomputing.org>
- [5] Ko, A. J.; Oleson, A.; Ryan, N. y otros (2020). It is time for more critical CS education. Communications of the ACM, 63(11), 31-33.
- [6] Baladron, M. (2019). El Plan "Argentina Conectada": Una política de Estado desde la infraestructura de comunicaciones. Ciencia, tecnología Y política, 2(2), 017. Disponible en: <https://doi.org/10.24215/26183188e017>
- [7] Czemerinski, H.; Dabbah J. et.al; compilado por Carmen Leonardi et.al (2018). Ciencias de la computación para el aula: 1er ciclo de primaria: libro para docentes. Fundación Sadosky.
- [8] Tanenbaum, A. y Wetherall, D. (2012). Redes de computadoras, Quinta edición. México: Pearson Educación.
- [9] Ación, L. et al. (2021). Desmitificando la Inteligencia Artificial. Artículo publicado en el libro “Inteligencia artificial, una mirada interdisciplinaria”, Academias Nacional de Ciencias de Morales y Políticas, 2021.
- [10] Pedace, K., Schleider, T., & Balmaceda, T. (2022). Inteligencia artificial y sesgos: El caso de la predicción del embarazo adolescente en Salta. Revista Iberoamericana de Ciencia, Tecnología y Sociedad.

# Modelización como propuesta pedagógica para enseñar contenidos de Ciencias de la Computación

M Emilia Echeveste<sup>1</sup>, Jonathan Alonso<sup>1</sup>, Claudio Di Paolo<sup>2</sup>,  
Nicolás Balmaceda<sup>3</sup> y Consuelo Moyano Budde<sup>3</sup>

<sup>1</sup>Universidad Nacional de Córdoba- CONICET

<sup>2</sup> Instituto Provincial de Educación Media (IPEM)

<sup>3</sup> Universidad Nacional de Córdoba  
CÓRDOBA, ARGENTINA

[emilia.echeveste@unc.edu.ar](mailto:emilia.echeveste@unc.edu.ar), [jonymalonso@gmail.com](mailto:jonymalonso@gmail.com), [cjdipaolo@gmail.com](mailto:cjdipaolo@gmail.com)  
[nicolas.balmaceda@mi.unc.edu.ar](mailto:nicolas.balmaceda@mi.unc.edu.ar), [consuelo.moyano@mi.unc.edu.ar](mailto:consuelo.moyano@mi.unc.edu.ar)

## Resumen

Desarrollamos una propuesta de clase gestada en el marco de un proyecto de investigación denominado “Análisis y diseño de propuestas colaborativas de Modelización mediadas por Tecnologías Digitales para fortalecer el desarrollo profesional de docentes de Matemática y Computación” financiado por la Secretaría de Ciencia y Tecnología del Ministerio de Producción, Ciencia e Innovación Tecnológica de la provincia de Córdoba (Mincyt) y la Secretaría de Ciencia y Tecnología de la Universidad Nacional de Córdoba (Secyt). Se tomó como base los principios de la modelización matemática y se pensó una actividad para implementar en el espacio curricular “Sistemas Digitales de Información” en una institución educativa secundaria con orientación en informática de la ciudad de Córdoba. Desde el punto de vista teórico de Bassanezi sobre modelización matemática se propuso reversionar sus postulados y orientarlos a contenidos de las Ciencias de la Computación (CC). De esta manera, se co-construyó entre un docente del área e investigadores/as de Educación, Matemática y Computación una propuesta para llevar al aula. Si bien no se ha podido profundizar el análisis y reflexión de la implementación, presentamos la propuesta y los resultados de las percepciones estudiantiles sobre la misma. Esto permitirá seguir generando aportes a la didáctica de las CC.

**Palabras clave:** Modelización; Didáctica de las Ciencias de la Computación; Escuela secundaria.

## 1. Introducción

A continuación desarrollaremos una propuesta de clase llevada a cabo en el mes junio de 2024 que se armó en el marco de un proyecto de investigación denominado “Análisis y diseño de propuestas colaborativas de Modelización mediadas por Tecnologías Digitales para fortalecer el desarrollo profesional de docentes de Matemática y Computación”, financiado por la Secretaría de Ciencia y Tecnología del Ministerio de Producción, Ciencia e Innovación Tecnológica de la provincia de Córdoba (Mincyt) y la Secretaría de Ciencia y Tecnología de la Universidad Nacional de Córdoba. Este proyecto sienta sus bases teóricas en la modelización matemática como abordaje pedagógico, definida por Blomhøj [1] 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 cuales construir importantes conceptos matemáticos. La Matemática y las Ciencias de la Computación (CC) se han vinculado desde sus orígenes tomando, en reiteradas ocasiones, referencias y bases situadas en la trayectoria investigativa de la educación matemática, lo que permite incorporar y articular sus aportes a las prácticas educativas. De esta manera, 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.

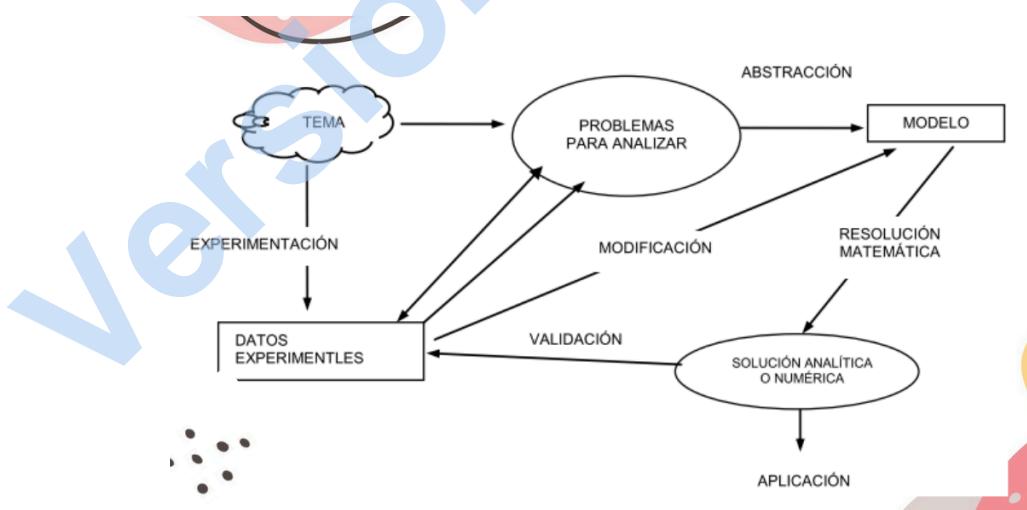
Como equipo nos posicionamos desde un enfoque de investigación llamado *investigación-acción participativa* que configura, analiza y resuelve un problema de investigación con actores que están involucrados en él, agrupando en la acción de investigar aquellos hechos que tengan implicancias y afecten, intentando generar transformaciones [3]. En esta oportunidad presentaremos una propuesta de clase implementada en el espacio curricular de “Sistemas Digitales de Información” de una institución educativa pública con orientación en informática de la ciudad de Córdoba. Recuperando los aportes teóricos de Bassanezi [2], los cuales se ampliarán en el apartado siguiente, co-construimos entre el equipo de investigación y un docente del área de informática, una actividad de modelización esta vez pensada con contenidos de las Ciencias de la Computación. A continuación desarrollaremos el enfoque teórico de la experiencia, recuperando los aportes que sirvieron de base para pensar la actividad de modelización en temas de Ciencias de la Computación. Luego desarrollamos la descripción de la experiencia que versa sobre la formulación de la actividad y la realización de la misma, la cual se generó luego de varios intercambios en reuniones de equipo. Dado que no podemos exponer análisis exhaustivos posteriores a la experiencia por ser una práctica realizada hace apenas un mes (con el receso invernal de por medio) sí desarrollaremos las primeras aproximaciones de la actividad. La misma finalizó la primera semana de julio en donde realizamos una encuesta al estudiantado para recabar información sobre el recibimiento de la propuesta. En los resultados presentamos el punto de vista de los y las estudiantes al mismo tiempo que las primeras reflexiones tanto del docente como del equipo de investigación.

## 2. Enfoque de la Experiencia

Como definición de modelización retomamos lo mencionado en el apartado anterior por Blomhøj [1] y recuperamos aportes de Bassanezi [2] para quien modelizar es obtener un modelo matemático, que implica la producción de conocimiento combinado con la abstracción y la formalización, vinculado a fenómenos y procesos empíricos vistos como situaciones problemáticas. Se pretende buscar una formalización artificial (modelo matemático) que contemple las relaciones que involucran tales argumentos.

Para pensar en conocimientos específicos de las Ciencias de la Computación, tomamos los aportes teóricos de Bassanezi [2] quien propone un ciclo que organiza el trabajo en el aula en función de fases o subprocessos que intervienen en un proceso de modelización en sentido matemático. Este autor considera los siguientes subprocessos: 1) elección de tema y formulación de problema(s), 2) experimentación, 3) abstracción, 4) resolución, 5) validación, 6) modificación y 7) aplicación, que se pueden observar en la Figura 1.

Figura 1. Ciclo de Modelización Matemática planteado por Bassanezi (2012)



Siguiendo ese mismo criterio, intentamos aplicar las mismas fases antes mencionadas, en este caso, buscando arribar a la formulación de problemas computacionales en donde se construyan “modelos y soluciones computacionales” como respuestas a problemáticas vinculadas a esta área específica. Cuando nos referimos a modelos, no se circumscribe solo a como se conoce tradicionalmente en las Ciencias de la Computación, ligados a estudiar el comportamiento de un sistema complejo por medio de la simulación por computadora. Sino más bien, siguiendo la línea de un proceso de modelización orientado a la resolución a un problema computacional, que nos permite aplicar un conocimiento o contenido de las Ciencias de la Computación.

Para pensar la propuesta se llevó a cabo una primera reunión a comienzo del mes de mayo de 2024, en el que se realizó una capacitación en temas de modelización en educación. Allí participó el equipo completo de investigación junto al docente que llevó a cabo la experiencia en el espacio curricular vinculado a las CC y un docente del espacio curricular de Matemática de otra institución escolar, quien realizó una experiencia similar a la relatada en esta oportunidad, pero con contenidos de matemática. Luego, para la planificación de la actividad, armamos un subgrupo compuesto por cuatro integrantes del equipo de investigación y el docente de “Sistemas Digitales de Información”. Realizamos cuatro reuniones de frecuencia semanal que comenzaron a inicios del mes de mayo de 2024. Allí pensamos conjuntamente la secuencia, qué contenido se podría abordar y, puntualmente, qué actividad se les propondría al estudiantado. Siguiendo el lineamiento de la modelización pensamos en una situación problemática que les generara interés, mediante la cual tuvieran que explorar, investigar y reconocer contenidos del espacio curricular en cuestión. Todo esto forma parte de la co-construcción.

El tema y el problema a resolver (Fase 1) fue seleccionado por el docente que implementó la propuesta. En este sentido, podemos decir que no se trata de una modelización abierta. En términos de Villarreal y Mina [3] se sugiere que, para que una modelización sea abierta, la Fase 1 de Bassanezi debería establecer la libertad para que el grupo estudiantil seleccione un tema y los medios a través de los cuales investigarlo, sin poder anticipar el contenido que se pondrá en juego. Según las autoras [3], existe poca investigación en torno a la modelización abierta en la escuela secundaria.

A continuación se presenta con mayor detalle la actividad, la cual se implementó en el aula a partir de la segunda semana de junio de 2024 y abarcó 6 encuentros de tres módulos de 40 minutos cada uno, los días jueves y viernes. La experiencia se desarrolló en una escuela pública de nivel secundario, ubicada en el centro de la ciudad de Córdoba. La institución alberga a una población estudiantil diversa, lo que contribuye a un ambiente educativo plural y enriquecedor. En esta experiencia participaron inicialmente 23 estudiantes, con una distribución equilibrada en cuanto al género con 12 mujeres y 11 varones con edades que varían de los 14 a los 18 años. Sin embargo, durante la implementación de la propuesta, la cantidad de participantes fue disminuyendo debido a diversas actividades que se llevaron a cabo en la escuela, afectando la continuidad y presencia de algunos estudiantes en el aula.

### 3. Descripción de la Experiencia

La experiencia que se presenta surge del trabajo colaborativo entre el equipo de investigación y el docente de una clase de 4º Año en la materia "Sistemas Digitales de Información". El objetivo fue construir una propuesta didáctica basada en la modelización como herramienta pedagógica para enseñar el tema "Técnicas de almacenamiento y sus volúmenes". El docente eligió este tema dado que aún no lo había abordado en su planificación, y consideró que trabajar con nuestro equipo de investigación ofrecía una buena ocasión para hacerlo. A partir de esta elección, se diseñó una situación problemática concreta que los estudiantes debían resolver dispuestos en grupos, relacionada con “la edición de material audiovisual producto de una fiesta de 15 años”. Según el docente, esta situación es especialmente relevante para el estudiantado, dado que, por su edad, están familiarizados con este tipo de eventos. Aquí se da inicio al proceso de modelización. Esto remite a la primera instancia del ciclo de modelización de Bassanezi: (1) *elección de tema y formulación de problema*.

Para guiar a los estudiantes en el desarrollo de esta situación, se diseñó una guía de actividades que abordaba aspectos clave como la elección de hardware adecuado, la selección de software de edición, la compatibilidad del software con diferentes dispositivos y la elección de un medio de entrega adecuado para el material final. La guía instaba a los estudiantes a investigar, tomar decisiones informadas y justificar cada una de sus elecciones, fomentando un enfoque crítico y reflexivo. Aquí, en términos de modelización, se llevó a cabo una (2) *experimentación*, (3) *abstracción*, y (4) *resolución*, en la medida que fueron explorando y resolviendo las preguntas formuladas en la guía. Cabe aclarar que, para Bassanezi, la abstracción se define como un proceso clave en el cual se extraen y seleccionan las características esenciales de una situación del mundo real para representarlas de manera simplificada a través de un modelo matemático.

- ¿Qué tipo de Hardware necesitarías? ¿Qué componentes de tu computadora consideras que son más relevantes para realizar este trabajo de almacenar, editar y distribuir el material? Justifica tu respuesta. /Buzón de sugerencia: tener en cuenta marcas, modelos, capacidades y características/
- En base a tu respuesta anterior, realizá un presupuesto teniendo en cuenta el menor costo posible que te permita armar/ ensamblar la máquina que necesitas. Para ello, te recomendamos que ingreses a diferentes páginas web argentinas, en lo posible cordobesas.

- ¿Qué programas de edición de imágenes y videos conocés? Explorá y elegí al menos tres programas que puedan cumplir esa función y determinar comparativamente sus requerimientos técnicos y qué capacidad de espacio te ocuparía en tu disco.
- De acuerdo a tu respuesta anterior, ¿ese software de edición se ejecuta en cualquier sistema operativo? )En una tablet, ¿podrías descargar estos programas? ¿Y en un celular? ¿Por qué?
- ¿Qué medio elegirías para entregar tu trabajo? (DVD, Blu-ray, USB, enlace digital, etc.). Elige uno y justifica tu respuesta. ¿Hay relación entre el formato elegido y la capacidad de almacenamiento? Tené en cuenta el formato en el que grabaste las imágenes y los videos.

A medida que fueron respondiendo a estas preguntas vivieron un proceso de modelización, ya que realizaron actividades de validación (5) y modificación (6) a medida que la misma exploración y recolección de datos les impedía dar respuesta a las preguntas de la guía. Por ejemplo, un grupo al elegir el formato de entrega de las fotos y los videos editados, consideró como primera opción armar una página web y dejar allí el material fotográfico, pero se encontraban con el problema de necesitar periódicamente un administrador del sitio web que realice la actualización y el mantenimiento, por lo tanto, cambiaron de opción

En tanto que avanzaba la realización de las actividades, el docente percibió que a sus estudiantes les costaba entender qué significaba realizar un trabajo del modo propuesto, es decir, trabajando clase a clase como un proyecto continuo y colaborativo. Esta dificultad se evidenció en la organización del trabajo grupal, la toma de decisiones informadas y la integración de las distintas tareas en un producto final coherente. Ante esta situación, el docente decidió diseñar una rúbrica de evaluación que sirviera como guía clara para el grupo de estudiantes. La rúbrica incluyó siete categorías que se pueden observar en la Figura :

Figura 2. Rúbrica de desempeño grupal diseñada por el docente

# Rúbrica de desempeño grupal

## Quinceañera: edición de fotos y video.

Elección de Hardware, Sistema Operativo y Software

Tareas	Regular 1	Bueno 2	Muy Bueno 3	Excelente 4	Puntuación (de 1 a 4)
<b>Selecciona el puntaje con un ✓</b>					
Interpretación de consigna	No reconoce al Informe escrito como material de entrega	Entienden que hay que hacer un informe, desconoce como hacerlo	Reconoce la presentación del informe, tienen idea pero necesitan ayuda	Trabajan independientemente en la construcción del informe.	<input type="checkbox"/>
Investigación y organización de la información	Poca investigación y desorganizada	Buena investigación, falta mejorar la organización	Muy buena investigación y organización	Correcta investigación y su organización	<input type="checkbox"/>
Hardware: Comparativa de presupuestos	Presupuesto incompleto sin comparación	Presupuesto logrado con una incompleta comparación	Presupuesto elegido con comparaciones bien presentadas	Menor presupuesto seleccionado entre 3 sitios web consultados comparables	<input type="checkbox"/>
Hardware: Elección de la computadora más adecuada	No reconocen las variables de importancia para la toma de decisiones	Deciden una computadora a partir de reconocer algunas variables	Arman una computadora a partir de los presupuestos.	Arman la computadora adecuada, a menor costo, a partir de los presupuestos.	<input type="checkbox"/>
Identificación de la compatibilidad entre Sw y S.O.	Reconoce un solo S.O. y sólo algunas funciones y Sw que corren en él.	Reconoce más de un S.O. y sólo algunas funciones comparables. Distingue compatibilidades	Asocia S.O. en distintas presentaciones de computadoras y reconoce compatibilidades entre Sw y S.O.	Entiende e identifica S.O. en distintas presentaciones de computadoras y reconoce compatibilidades y formatos	<input type="checkbox"/>

Identificación de la compatibilidad entre Hw y S.O.	Reconoce un solo S.O. y sólo algunas funciones	Reconoce más de un S.O. y sólo algunas funciones comparables	Asocia S.O. en distintas presentaciones de computadoras y reconoce compatibilidades	Entiende e identifica, S.O. en distintas presentaciones de computadoras y reconoce compatibilidades	<input type="checkbox"/>
	Presenta una redacción que no respeta el formato	Presenta, en formato de informe, una incompleta descripción vincular	Presenta un informe bien desarrollado. Defiende bien su presentación.	Presenta y defiende en forma correcta su Informe	<input type="checkbox"/>
Presentación y entrega de informes. Defensa de su trabajo.	Presenta una redacción que no respeta el formato	Presenta, en formato de informe, una incompleta descripción vincular	Presenta un informe bien desarrollado. Defiende bien su presentación.	Presenta y defiende en forma correcta su Informe	<input type="checkbox"/>
					<input type="checkbox"/>

#### VALORACIÓN FINAL

PUNTOS	CALIFICACIÓN	PUNTOS	CALIFICACIÓN
27/28	<b>EXCELENTE</b>	14/20	<b>BUENO</b>
21/26	<b>MUY BUENO</b>	< 14	<b>REGULAR (Rehacer)</b>

Esta herramienta no solo facilitó la evaluación de los trabajos sino que, también, ayudó a los grupos de estudiantes a comprender mejor las expectativas y objetivos del proyecto.

## 4. Primeros análisis de la experiencia

La implementación de la propuesta didáctica en el aula fue un proceso dinámico y colaborativo que involucró tanto a los estudiantes como al docente. Desde el inicio, se planteó una situación problemática basada en un contexto real relacionado con la edición de material audiovisual producto de una fiesta de 15 años. Esta situación fue diseñada para ser lo más cercana posible a las experiencias cotidianas del estudiantado, facilitando así la conexión entre los contenidos teóricos y su aplicación práctica. Algunos indicios de esto se podían notar a través de comentarios en el aula, como "mirá, esto es para vos", entre estudiantes que estaban próximas a vivenciar un cumpleaños de 15.

La guía de actividades proporcionó al estudiantado un marco estructurado para abordar el problema. Este enfoque de trabajo resultó novedoso para ellos, ya que no estaban acostumbrados a una metodología que requería tanta autonomía y responsabilidad en la toma de decisiones. A lo largo del proceso, los estudiantes trabajaron en grupos, lo que fomentó la colaboración y el intercambio de ideas, aunque también supuso un desafío adicional al exigirles adaptarse a una dinámica de trabajo más activa y participativa.

El docente jugó un papel fundamental en la experiencia, actuando como orientador y guía. Su participación permitió que las actividades fueran adaptadas a las necesidades y al ritmo de aprendizaje estudiantil. Además, la rúbrica de desempeño grupal diseñada por el docente ofreció a los y las estudiantes una comprensión clara de los criterios de evaluación.

## 5. Resultados

A través de la encuesta realizada a los y las estudiantes se pueden reconocer dos grandes aportes de la propuesta a su clase de “Sistemas Digitales de Información”. Por un lado, el **trabajar en equipo** presenta mayor recurrencia en sus respuestas sobre aquello que más les gustó. Los convoca la idea de que sea *en grupo*, que haya “intercambio de ideas”, tal como lo menciona una de los estudiantes; o, incluso, “poder hablar y compartir ideas con los compañeros de grupo” (otra de las respuestas obtenidas). También les pareció interesante hablar con gente nueva, en este caso, uno de los compañeros investigadores del equipo. Esto muestra la riqueza no solo de la interacción entre pares sino también de otros sujetos externos a la institución. En este sentido, la actividad de modelización fomenta acciones como entrevistar a

personas o especialistas del área a los fines de conseguir respuestas al problema planteado. De hecho, en uno de los grupos se consideró consultar con la abuela de un integrante que se dedicaba a la fotografía.

Por otro lado, pero ligado a lo anterior, se rescata la posibilidad de *hacer una actividad diferente a la que estaban acostumbrados*, más precisamente, la posibilidad de "buscar información" o investigar. Por ejemplo, en una de las encuestas, un estudiante aludió a que "*hubo mucha investigación*". La modelización, entendida en términos de Bazanessi, presenta actividades como la experimentación, la validación del modelo y el dinamismo mismo del ciclo, que permiten la búsqueda de información y la exploración, actividades que se reconocen positivamente entre quienes la realizan. Por ejemplo, uno de los estudiantes menciona que consiguió "*La experiencia de saber cosas de computación*", lo que refuerza la propuesta a los fines de poner en juego saberes de las Ciencias de la Computación. Algunos de estos saberes que forman parte de la currícula fueron: identificar componentes de hardware y software, analizar el uso de sistemas operativos y aplicaciones que puedan correr en ellos, compatibilidades, tamaño de datos y soportes donde almacenarlos.

Todo el grupo de estudiantes reconoce que no ha tenido una propuesta similar anteriormente. En especial, hacen referencia a actividades como interactuar con internet, "ponerme a buscar, a ver y comparar los precios"; lo consideran como un proyecto que "duró bastante tiempo y necesitaba mucha elaboración". Si bien no aparecen grandes puntos negativos en sus devoluciones, algunos estudiantes sugieren ajustes vinculados a tener una mayor cantidad de tiempo, tanto en la explicación de la actividad como en la realización de la misma.

## 4. Reflexiones finales

El desarrollo y análisis de propuestas que recopilen información de contextos educativos reales, en donde se ponen en juego prácticas educativas de modelización, favorece un proceso de aprendizaje en el que se establece la construcción de conceptos de las Ciencias de la Computación. Tal como se mencionó en el apartado resultados, esta propuesta no solo le permitió a este grupo de estudiantes "la experiencia de saber cosas de computación" sino que se considera como la posibilidad de trabajar en el aula realizando actividades que no habían realizado anteriormente. Además, vinculan esta propuesta a una posible actividad laboral que podrían realizar, logrando así un sentido y aplicabilidad a lo aprendido. Esto remite a la fase 7 del ciclo de Bassanezi que es la *aplicación* ya que les permite a los y las estudiantes ver la utilidad y el impacto de lo aprendido de computación en situaciones reales. En este sentido, no solo validan su aprendizaje teórico, sino que también adquieren habilidades para aplicarlos de manera práctica y contextualizada.

Esta propuesta es un aporte a los desarrollos didácticos de las Ciencias de Computación, tomando en cuenta los desarrollos de la educación matemática que lleva décadas de estudios en esta temática.

## 5. Agradecimientos

A la institución educativa que nos abrió las puertas para poder realizar nuestra propuesta. Al resto del equipo de investigación, que, si bien no figura en la autoría del escrito porque realizaron otra intervención educativa, siempre acompañan y sostienen los proyectos.

## Referencias

- [1] 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>.
- [2] Bassanezi, R. C. (2012). Temas y modelos (1.ed.). Campinas, Brasil: UFABC.
- [3] Villarreal, M. E., & Mina, M. (2020). Actividades experimentales con tecnologías en escenarios de modelización matemática. Boletim de Educação Matemática, 34(67), 786-824.
- [4] 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.

# Pilas y Bloques, aliado como andamiaje hacia experiencia construcciónista con robótica educativa.

Ana Lía Zamora, Belén Rodriguez

<sup>1</sup> Escuela de Comercio N°4 Batalla de Tucumán, San Miguel de Tucumán, Argentina  
[analiaz10@gmail.com](mailto:analiaz10@gmail.com), [gabinetepedagogicoc4@gmail.com](mailto:gabinetepedagogicoc4@gmail.com)

## Resumen

La experiencia educativa consistió en presentar a Pilas y Bloques como aliado en el inicio del camino de aprendizaje hacia otros softwares que presentan el lenguaje programación en bloques hasta llegar a la Robótica con el propósito de ir de una propuesta de enseñanza simple a una más compleja. Esta propuesta pedagógica, consistió en brindarles al profesor y a los estudiantes, en el marco del proyecto de Feria de Ciencias, talleres teórico-prácticos, iniciado con Pilas y Bloques como andamiaje pedagógico para lograr la introducción gradual al lenguaje de programación, hasta lograr el prototipo final con el kit de Lego, en este caso: una máquina programada con sensor, que separa según el color del objeto al pasar por la cinta transportadora, con la respectiva programación para su funcionamiento. Este recorrido puede compararse con la alfabetización inicial, donde aprender a leer y escribir el alfabeto es el primer paso para dominar un nuevo lenguaje, en este caso, el lenguaje y la lógica de la programación. Fue Pilas y Bloques un aliado fundamental en este proceso de alfabetización digital.

**Palabras clave:** Alfabetización digital; Saberes emergentes; Programación con bloques; Robótica educativa; Andamiaje pedagógico.

## 1. Introducción

Implementar programación como una herramienta transversal, fue un desafío para nuestra institución educativa, debido a que no contamos actualmente con un espacio curricular específico que esté vinculado al aprendizaje de las Ciencias de la Computación, puesto que somos una Escuela de Comercio con orientación en Economía y Administración y orientación en Ciencias Naturales, pero estamos convencidos que nuestra misión como establecimiento educativo es formar ciudadanos que estén preparados para insertarse en la sociedad digital, por lo que se tomó la decisión de qué saberes emergentes como la programación y la robótica se aborden de forma transversal en distintas materias.

En este sentido, el propósito y el sentido educativo del desarrollo de esta experiencia, es propiciar en los profesores de espacios curriculares como Tecnología, a través del proyecto para feria de ciencias “ECO LEGO”: una Innovación Tecnológica para un ambiente sostenible, o en la materia informática, el uso de un nuevo lenguaje de programación. Se trabajó con el software Pilas y Bloques, que “*es una aplicación desarrollada por Program.AR de la Fundación Sadosky para enseñar y aprender a programar por medio de bloques de forma simple y divertida*” [1], se presentaron desafíos con diversos niveles de dificultad para que los adolescentes puedan conocer el mundo de la programación y a su vez abrir camino al paso siguiente, que se materializó en el espacio físico con el kit Lego Mindstorms.

Proyectando de esta manera, continuar con distintas materias para que sea la programación quien realice la transversalidad, “...*hay abundante evidencia científica que indica que los niños/as y adolescentes que aprenden Programación, mejoran su desempeño en otras áreas disciplinares, como matemática y lenguas extranjeras*”[2]

Así mismo, tomamos los lineamientos del marco legal vigente en nuestro país, la Ley de Educación Nacional N° 26.206, la presente en el Cap II, Art. 11 señala entre los fines y objetivos de la política educativa nacional la de desarrollar las competencias necesarias para el manejo de los nuevos lenguajes producidos por las tecnologías de la información y la comunicación, además promover el aprendizaje de saberes científicos fundamentales para comprender y participar reflexivamente en la sociedad contemporánea. En respuesta al Capítulo III Artículo 88, la Res. CFE 343/18 aprueba los Núcleos de Aprendizaje Prioritarios de Educación Digital, Programación y Robótica para facilitar el acceso y dominio de las tecnologías de la información y la comunicación en los contenidos curriculares indispensables para la inclusión en la sociedad del conocimiento.

“El mundo vive una transformación que tiene pocos antecedentes en la historia, marcado por la evolución y los avances existentes en las Tecnologías de la Información y la Comunicación (TICs), un conjunto de nuevas tecnologías (inteligencia artificial, internet de las cosas, análisis de big data, impresión 3D, sensores inteligentes, etc.) que parece estar cambiando radicalmente la forma en que producimos, consumimos, comercializamos y, por supuesto, la manera en que trabajamos.” [3]

Lo que nos lleva como educadores comprometidos a generar propuestas de innovación educativa y tecnológica que encuentren entre sus objetivos el de garantizar la alfabetización digital para el aprendizaje de competencias y saberes necesarios para la integración en la cultura digital y la sociedad del futuro.

## 2. Descripción de la Experiencia

### 2.1. Propuesta Didáctica

Este proyecto educativo se basó en el Aprendizaje Basado en Proyecto, que enfatiza el aprendizaje activo y la interdisciplinariedad. Aborda los contenidos del Diseño Curricular Jurisdiccional del área de biología vinculados a la educación ambiental, los procesos de contaminación. Causas y consecuencias ambientales, del espacio curricular de tecnología los ejes relacionados a los procesos tecnológicos, los medios técnicos y la reflexión sobre la tecnología como proceso sociocultural: diversidad, cambios y continuidades.

También se trabajó con saberes emergentes que se abordan en los NAP de Educación Digital, Programación y Robótica.

En lo que respecta a la evaluación, fue formativa para acompañar el proceso de aprendizaje de los alumnos, nos permitió observar, retroalimentar y ajustar las estrategias de enseñanza y aprendizaje en tiempo real para que los estudiantes adquieran habilidades de manera gradual aplicando conocimientos teóricos en contextos prácticos.

En todo momento se realizó un seguimiento individualizado y se valoró el trabajo colaborativo, lo que nos permitió identificar las fortalezas y las áreas de mejora de cada estudiante y del trabajo en equipo. Fue la retroalimentación constante la que le ofreció a los estudiantes la oportunidad de mejorar, de reformular la hipótesis y perfeccionar el prototipo antes de llegar al resultado final. Este proceso garantizó no solo la adquisición de habilidades técnicas de programación y robótica sino de competencias como el pensamiento crítico y la creatividad, claves para la resolución complejos y auténticos como se enfrentaba en este proyecto para resolver un problema de su comunidad.

### 2.2. Contexto

La Experiencia Educativa se desarrolla en la Escuela de Comercio N°4 - Batalla de Tucumán, una institución educativa pública de gestión estatal, ubicada en la Provincia de Tucumán, en la Capital de San Miguel de Tucumán, específicamente en el Barrio Oeste 2, con los estudiantes del Nivel Secundario correspondientes al Ciclo Básico- 2º año división “5” que estuvieron involucrados en el proyecto para Feria de Ciencias: “ECO LEGO”: una Innovación Tecnológica para un ambiente sostenible.

El establecimiento cuenta con turno mañana, turno tarde y turno noche, tiene una población de 750 estudiantes, entre 12 y 18 años, como así también, destinatarios adultos en el turno noche.

Esta propuesta pedagógica, diseñada para ser presentada en la Feria de Ciencias, surgió a partir del interés de los estudiantes por la importancia de habitar un entorno sostenible y del conocimiento previo que han construido durante su trayectoria en la institución educativa en torno al compromiso con los problemas medioambientales. A partir de esto, se planteó el desafío de explorar cómo la programación y la robótica pueden ser tecnologías clave para favorecer la resolución de dichos problemas.

### 2.3. La Experiencia

Se transitó por varias etapas de indagación, comenzando con la identificación del problema de los residuos plásticos y su impacto ambiental y se planteó una hipótesis sobre la efectividad de la robótica como herramienta educativa para fomentar la sostenibilidad.

A través de talleres teóricos y prácticos, los estudiantes aprendieron a construir el robot utilizando el kit LEGO que recibió la institución educativa en el marco del Programa Aprender Conectados, armaron el prototipo, aprendieron a manejar el sensor de color, los motores de movimiento, el comando receptor y trasmisor central (ladrillo EV3) y a utilizar el diversos lenguajes de programación en bloques con distintos software: como Scratch y Pilas y Bloques , que les permitió indagar desde allí el efecto y funcionamiento de instrucciones en el entorno virtual y luego en el físico, para poner en funcionamiento el robot.

Los resultados obtenidos mostraron: no solo un aumento significativo en el conocimiento y la conciencia ambiental de los estudiantes, una mejora en sus habilidades tecnológicas, en lectoescritura, oralidad y matemática sino también el desarrollo de competencias como la exploración, la comunicación, la colaboración entre pares, el trabajo en equipo y la creatividad para la resolución del problema de la clasificación de residuos.

Los Materiales que se utilizaron en esta propuesta educativa son:

- Kit Lego Mindstorms EV3- Netbooks. -Smart TV -Video Motivacional “porque es importante saber programar” Program.AR [5]- Manuales de armado-Videos tutoriales-Diario de Apuntes.
- Software: Pilas y Bloques, Scratch y Lego Education.
- Se trabajó en los distintos espacios de nuestro establecimiento como: el Salón de Usos Múltiples (SUM), biblioteca, aula y taller de laboratorio para cada momento y propósito según el uso de las herramientas. Manipulación y cuidados del kit lego.

### Metodología:

El proyecto educativo se desarrolló utilizando una variedad de metodologías activas que promueven el aprendizaje significativo y la transferencia de conocimiento a contextos reales. Estas metodologías están alineadas con los enfoques pedagógicos actuales que buscan desarrollar competencias como la creatividad, el trabajo colaborativo, la comunicación, el pensamiento crítico. A continuación se presentan las metodologías utilizadas:

Se trabajó a partir del Aprendizaje Basado en Proyecto es una metodología de diseño abierto y flexible y la planificación implementa un conjunto de tareas basadas en la resolución de preguntas o problemas (desafíos), mediante un proceso de investigación o creación por parte de los estudiantes que trabajan de manera relativamente autónoma, con un alto nivel de implicación y cooperación, culminando con un producto final, y la difusión del mismo.

En este contexto, Perrenoud (2000) afirma que “*el trabajo por proyectos permite construir saberes, competencias, descubrir nuevos mundos relacionados con el contrato social, desarrollando una inteligencia colectiva, autónoma con capacidad de elegir y trabajar cooperativamente.*” [4]

Por su parte, Rebeca Anijovich y Silvia Mora, sostienen que: “*Un proyecto tiene como intención: la resolución de un problema, la producción de un objeto, la profundización de algún conocimiento, Para que un proyecto resulte significativo, en términos de aprendizaje, es necesario que los alumnos se propongan una meta, planifiquen las acciones para cumplirla, lleven adelante una diversidad de actividades, prueben y elijan caminos alternativos, recursos variados y tomen decisiones para cada uno de estos trayectos. En síntesis, es preciso que los estudiantes se involucren en un proceso de planeamiento, investigación, práctica y toma de decisiones*”. [5]

Para cumplir con el objetivo, se implementó la dinámica de taller con instancias teóricas y prácticas, lo que habilitó el debate abierto para animar a los alumnos a hablar sobre temas relacionados con este proyecto en el proceso de implementación se contó con propuestas de Gamificación en donde se utilizó el interés y la motivación intrínseca generada por los juegos para mejorar el aprendizaje y el compromiso.

Se plantearon desafíos a los estudiantes para que puedan completar tareas, participar en actividades y lograr objetivos específicos.

Transitaron por distintos niveles que representaban etapas para avanzar a niveles superiores, lo que simboliza su progreso y logros.

Los desafíos se plantearon para que la resolución sea individual o en equipo y recibieron retroalimentación instantánea sobre su desempeño que los ayudó en todo momento a mantener la motivación y a corregir errores rápidamente. La experiencia aportó a la exploración a partir de Visual Thinking se brindó una programación con uso de bloques de instrucciones y desde Design thinking se asume un enfoque colaborativo y creativo que incluye la empatía, la definición del problema, la generación de ideas, la prototipación y la experimentación. Esta metodología se utiliza en una amplia gama de contextos, desde el diseño de productos y servicios.

En conjunto, estas metodología aportaron un marco integral para la implementación del proyecto que se desarrolló en las siguientes instancias:

Etapa 1: Sensibilización ambiental, lo que permitió reconocer un problema que afecta a la comunidad. Para ello, los estudiantes realizaron una indagación exhaustiva en la que identificaron, reconocieron, y comprendieron el problema de los residuos plásticos, su acumulación y las consecuencias tanto a nivel global como local para el medio ambiente y para la salud pública.



Se realizó una revisión de las prácticas actuales en lo que hace a la gestión de residuos, tales como la recolección y la separación, particularmente en su barrio densamente poblado, donde predominan viviendas en monoblocks, lo que representa un desafío específico el tema de la recolección de residuos.

En este marco, se planteó la hipótesis sobre ¿Cómo podría la robótica ser una herramienta efectiva para fomentar la sostenibilidad en el manejo de los residuos? Partiendo del supuesto que la programación y la robótica se pueden convertir en motor de cambio, fomentando no sólo la participación activa de los estudiantes, sino también su capacidad para liderar iniciativas

que beneficien a su entorno. Posteriormente, se consideró que el desarrollo de un prototipo robótico para la clasificación automatizada de residuos podría aumentar la eficiencia del reciclaje y una gestión más sostenible en el barrio Oeste II.

Se procedió al reconocimiento del kit Lego y sus distintas partes, como así también, a visualizar el manual de indicaciones para la construcción de distintos prototipos.

Etapa 2: Introducción a la programación, fue la siguiente fase para poder llegar al objetivo, se abordó a partir de la exposición interactiva, se enseñó el desarrollo del pensamiento computacional, la lógica de la programación, el uso y funcionamiento de Softwares con programación en bloque, se realizaron desafíos graduales para programar el sensor y detectar el color.

Estos talleres se realizaron en el SUM del Establecimiento, les mostramos a los alumnos en el SmartTv el video: ¿Por qué todos tienen que aprender a programar? Participá de la Hora del Código . [6]

Posteriormente se reflexionó con los estudiantes sobre el video y sobre conceptos teóricos partiendo de la idea que “*Un programa es una forma de expresar una solución a un problema de forma tal que una máquina pueda ejecutarlo. En este sentido, los programas no son más que un conjunto de símbolos –letras, números, etc.– que describen la solución que pensó alguien. Para poner efectivamente en funcionamiento un programa, tiene que haber una máquina que lo ejecute. La Programación, como área dentro de las Ciencias de la Computación, trabaja sobre los programas y la forma de construirlos y no sobre las máquinas que los ejecutan.*

*Una cuestión fundamental es que, además de ser ejecutados por una máquina, los programas tienen que poder ser leídos y entendidos por personas: para poder corregir, adaptar o modificar un programa, es imprescindible que pueda leerse y comprenderse con facilidad.”* . [7]

Luego aplicamos conceptos y prácticas de distintas secuencias didácticas, del manual para docente de fundación Sadosky v2017[8] , las mismas fueron:

Secuencia didáctica N° 1: Proponer a los alumnos que identifiquen actividades que realizan cotidianamente, las que involucran una serie de pasos o acciones e indicar nombre de cada acción. Dado que el Sum queda en la planta de arriba de la escuela, dimos lugar a que se expliquen en identificar estos pasos y acciones, desde que salieron del aula hasta llegar a donde estaban sentados. (salir del aula, caminar cantidad de pasos hasta la escalera, girar a la derecha o izquierda, subir un pie, luego el otro, y así sucesivamente). Otro ejemplo que usamos fué el del clima en esos días para la elección de abrigarse si hacía frío, etc. También resultó un a guia el material de curso virtual a cargo del Ministerio de Educación de Tucumán como “pensamiento computacional y la vida cotidiana”

Secuencia didáctica n° 2: Se realizó una introducción a la interfaz de Pilas y Bloques, versión online y offline, en este caso se trabajó sin conexión a internet debido que en ese momento no contamos con wifi en el establecimiento. Se hizo hincapié en sus componentes principales como ser el área de bloques, sus categorías, el área de ejecución del programa y la identificación del personaje y se plantearon ejercicios organizados de acuerdo a los conceptos del pensamiento computacional: Autómatas, comandos, procedimientos y repetición, alternativa condicional, repetición condicional, sensores numéricos, parametrización de soluciones.



Así mismo con la plataforma de scratch se realizó un recorrido por el entorno, focalizando en cada una de las funciones de la plataforma: el escenario, el área donde se colocan y organizan los bloques de código, la barra de herramientas de bloques, sus categorías (movimiento, apariencia, sonido, control, sensores, operadores y variables) y el área de personajes, para que puedan crear su primer proyecto.



Con respecto al software Lego Education, se explicó con más detalle las funciones de los componentes del kit, como ser los motores, sensores y el controlador central que recibe las instrucciones del programa, para que los estudiantes también se familiaricen con la interfaz gráfica de programación, que es similar a Scratch pero con bloques específicos para controlar los elementos robóticos.

Para ello, cada uno de los alumnos siguió los pasos para instalar en su propia netbook el programa Lego Education. Al proceder en el uso del mismo, se observó que los alumnos estaban familiarizados totalmente con el entorno por los softwares usados anteriormente, trabajaron en las instrucciones que el prototipo requería para su respectivo movimiento del motor, de la cinta transportadora, detección del color mientras pase un elemento por delante de este, emitir el sonido de la pronunciación del color detectado, y mover el motor hacia la izquierda o derecha según el color detectado, a través de las instrucciones programadas.

**Secuencia didáctica n°6:** Nos permite aplicar la similitud de los desafíos realizados en pilas y bloques, en lo que representa el uso de la repetición, aquí materializaron su aplicación con el movimiento constante de la cinta transportadora mientras estaba encendido, y a su vez el uso de una función creada por el usuario. Para la toma de color del sensor usamos tapitas de colores como prototipos de objetos diversos de plástico que podrían ser reciclables en la vida real.



**Etapa 3: Construcción de robots con el Kit Lego Mindstorms EV3.** Con ayuda del manual para armado del modelo ev3 máquina con cinta transportadora. Para poder cumplir con el propósito, se realizaron variaciones con el modelo propuesto, se transitó por una instancia de exploración del montaje del robot, para posibilitar que los alumnos prueben sus soluciones propuestas con la programación.



Etapa 4: Programación de robots para tareas de sostenibilidad, el prototipo diseñado posibilita el recorrido del objeto de color, por el cual al pasar por el sensor de color, detecta el mismo (en este caso amarillo y rojo) y según sea, el motor que contiene el separador en forma de paleta que lo mueve hacia la izquierda, derecha o se queda estatica, para dejarlo pasar al final. Siendo de esta forma el sensor óptico quien actuaba ante los colores programados para separar y recolectar.

Etapa 5: Evaluación y retroalimentación: a partir del software Lego Education y la prueba en el prototipo se realizaron diversas pruebas, identificación del error como ser la detección de colores debido a la sensibilidad de los sensores, problemas de sincronización entre el sensor y los movimientos de la máquina.

Los estudiantes realizaron la corrección de errores, con ajustes en el código de programación y en la estructura del prototipo. Esta fase se caracterizó por una constante prueba, error y ajuste, en el que cada corrección se evaluaba hasta que lograron el funcionamiento deseado.



Una vez finalizada la etapa de pruebas y ajustes, los alumnos voluntarios para exponer que debían ser 4 para representar a su curso, se prepararon para participar en las Ferias de Ciencia de la Provincia de Tucumán. En esta instancia, se les guió en cómo comunicar eficazmente los objetivos de su proyecto, el proceso de construcción del prototipo y los desafíos que enfrentaron durante el desarrollo.

En esta instancia, los estudiantes tuvieron la oportunidad de compartir su experiencia con otras escuelas de la zona y recibir retroalimentación de parte de jueces.

2 DE AGOSTO DE 2024

La Escuela Secundaria Ricardo Gutiérrez recibió a la instancia interescolar de la Feria de Ciencias.  
Alumnos de 2°5°: Balin Valentín, Suarez Alessandro Cruz Lucia Carolina, Suarez Fiorella, representando a sus compañeros con el Proyecto "ECO LEGO": una Innovación Tecnológica para un ambiente sostenible.  
Felicitaciones trabajaron y aprendieron un montón! 🎉🎉  
<https://www.educaciontuc.gov.ar/.../la-escuela.../>



Publicación en la Red Social Facebook de la Esc. de Comercio N°4- Batalla de Tucumán

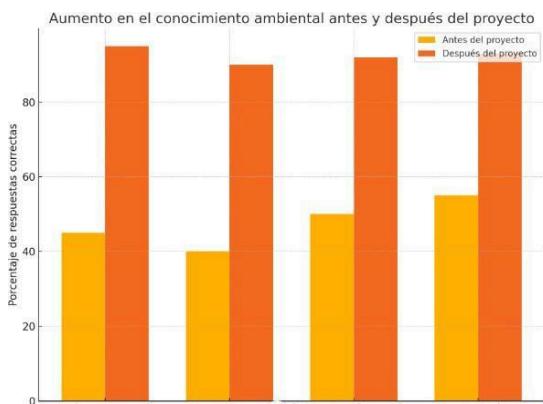
Luego del recorrido estamos convencidos que asegurar aprendizajes significativos y socialmente relevantes, generadores de capacidades, se vuelve un imperativo si la educación pretende dar respuesta a los retos propios de la complejidad de la vida cotidiana, social, política -regional, nacional y mundial- y, al mismo tiempo, conciliar esta ineludible responsabilidad social con la de generar condiciones para que los estudiantes puedan construir un proyecto personal de vida.

### 3. Resultados

El Proyecto Educativo, tuvo un impacto significativo en el aprendizaje de los estudiantes. Se observó un incremento considerable en el conocimiento ambiental de los estudiantes. A través de la investigación y sensibilización sobre el problema de los residuos plásticos y su impacto en la comunidad del Barrio Oeste II de la Provincia de Tucumán, los estudiantes desarrollaron una mayor comprensión de la sostenibilidad y en lo que respecta al desarrollo de prototipo se observa que se consolidaron sus competencias en programación y robótica.

A continuación se detallan gráficos y tablas que visibilizan los resultados obtenidos:

**Figura 1: Aumento en el conocimiento ambiental antes y después del proyecto**



**Figura 2: Presenta el desarrollo de las habilidades de programación y la construcción de robots:**

Habilidad	Antes del proyecto	Después del proyecto
Construcción de robots	0	15 (Totalidad de los alumnos)
Programación básica	3	15 (Totalidad de los alumnos)
Adaptación del software al prototipo	0	10

A lo largo del desarrollo del proyecto educativo, los estudiantes evidenciaron una notable mejora en sus habilidades de programación y de construcción de robots. A medida que avanzaban en la construcción del prototipo, realizaron pruebas sucesivas que les permitieron adaptar y ajustar tanto el software como el prototipo físico, hasta llegar al resultado esperado.

**Figura 3: Evaluaciones de la percepción de los estudiantes sobre la importancia de la sostenibilidad.**

Habilidad	Antes del proyecto	Después del proyecto
Impacto de los residuos plásticos en el ambiente	3	(Totalidad de los alumnos)
Responsabilidad personal en la sostenibilidad	3	(Totalidad de los alumnos)
Conocimiento de acciones sostenibles diarias	5	(Totalidad de los alumnos)

## 4. Conclusiones

A través de diversas disciplinas, herramientas digitales y estrategias metodológicas , pudimos participar y trabajar motivados para que los estudiantes adquirieran nuevas habilidades, en el marco de la "alfabetización digital". Iniciamos en un entorno virtual y, finalmente, trasladamos la experiencia al espacio físico, con el propósito de lograr la resolución de un problema concreto.

Destacamos la importancia de contar con una herramienta educativa virtual de origen nacional, que nos permitió establecer un primer contacto con la programación en bloques e interactuar de manera efectiva con este entorno. Fue a partir de la gamificación a través de los desafíos de Pilas y Bloques, que se logró el desarrollo del pensamiento computacional y la lógica de la programación, pudieron transitar por pruebas, errores y autocorrecciones, lo que les posibilitó a los alumnos conectarse con el buen sentido de los errores, que equivocarse los hacía volver a intentar y aprender, siempre guiados para reconocer que no está mal equivocarse en estas etapas de diseño.

Aprendieron que el proceso de "prueba y error" es fundamental y que la retroalimentación les permite realizar nuevas pruebas, lo que fortaleció el entendimiento de los procedimientos los preparó para pasar al hardware, como un siguiente paso en la construcción, donde la lógica de funcionamiento es comandada por el software, a través de instrucciones en forma de lenguaje. Incluso cuando el prototipo presentaba errores al funcionar, les resultaba más fácil identificar y detectar la causa y origen de dichos fallos.

Se realizaron pruebas en la que los alumnos probaron su código al conectar en el kit de hardware y si se presentaba un error era el motivo de colaboración entre pares, y sus rostros eran de alegría al descubrir una solución y ver cuando el movimiento en el mundo físico provenía del mundo virtual.

Esta es una experiencia transitó el camino inverso, partió primero del entorno virtual, con la enseñanza y el aprendizaje gradual de diversos softwares de programación en bloques y luego la construcción material del hardware con la posibilidad de armar, desarmar, reconstruir una vez probado sus códigos iniciales, ganando la seguridad de recrear, reinventar, hasta lograr el objetivo deseado.

Cerramos nuestras conclusiones tomando los aportes de Papert (2020):

*"El constructivismo Papertiano se centra fundamentalmente, en el arte de aprender a aprender utilizando tecnología. No obstante, la importancia radica en la significación de hacer cosas para aprender...."*

*Constructivismo + tecnología = constructivismo*

*Papert sostiene que entre más variado y sofisticado sea el material de que dispone el educando, más robusto y duradero será su conocimiento construido.".[9]*

## 5. Agradecimientos

Queremos expresar nuestro profundo agradecimiento a los estudiantes de 2 Año "5" División, por su activa y entusiasta participación en cada una de las etapas del proyecto. Su dedicación y curiosidad fueron esenciales para alcanzar los objetivos propuestos, demostrando no solo habilidades técnicas de programación y robótica, sino también un gran trabajo en equipo y su capacidad para aplicar los conocimientos adquiridos a problemáticas reales de su entorno socio-cultural.

Agradecemos también al Sr. Director, José Juárez, y a la Sra. Vicedirectora, Liliana Lucena, por su continuo apoyo y visión para fomentar un entorno educativo que promueve la innovación, el trabajo interdisciplinario y el uso de la tecnología como herramienta educativa.

Asimismo, un especial reconocimiento al profesor de tecnología, José Madrazo, quien con su experticia permitió que los alumnos pudieran acceder a diversas herramientas tecnológicas, gestionando su uso de manera óptima y creativa.

Destacamos el compromiso de nuestro establecimiento educativo, Escuela de Comercio N°4 Batalla de Tucumán, por promover una enseñanza integral, innovadora para mejorar las prácticas pedagógicas y lograr aprendizaje profundo, transferibles y significativos.

Agradecemos a todo el equipo de la Fundación Sadosky por brindarnos la posibilidad de contar con herramientas pedagógicas armadas tanto de software como el acceso a manuales para capacitación y estudio, de forma gratuita y fácil acceso, cuya transposición didáctica elaborada por especialistas, nos permite y facilita algo accesible, para aprender y enseñar, permitiendo que nuestro trabajo se concentre más en comunicarlas de manera eficiente y trabajo en equipo a través de distintas formas, adaptables a la situación de cada comunidad educativa y trabajar más en gestionar los recursos de la mejor manera posible para su acceso y uso..

## Referencias

### Recursos en línea

- [1] Pilas Bloques es una aplicación para enseñar y aprender a programar.[Entrada web] Recuperado de:: <https://pilasbloques.program.ar/acerca-de-pilas-bloques/>
- [2] Resolución CFE N° 263/15 Buenos Aires, 12 de agosto de 2015. [Online]. Available: [http://www.bnm.me.gov.ar/giga1/normas/RCFE\\_263-15.pdf](http://www.bnm.me.gov.ar/giga1/normas/RCFE_263-15.pdf), pp.2
- [3] Publicación BID: Instituto para la integración de América Latina y el Caribe (INTAL). Sector de Integración y Comercio (INT). Nota técnica n° idb-tn-1672. Travesía 4.0 Hacia la transformación industrial Argentina. Ramiro Albrieu- Ana Ines Basco- Caterina Brest Lopez- Balisario de Azevedo-Fernando Peirano- Martín Rapeti- Gabriel Vienni. Argentina Junio 2019. pp15.[Entrada web] Recuperado de: <https://publications.iadb.org/es/travesia-40-hacia-la-transformacion-industrial-argentina>
- [4] PERRENOUD, PHILIPPE. Aprender en la escuela a través de proyectos: ¿Por qué? ¿Cómo? (2000), Facultad de Psicología y de Ciencias de la Educación, Universidad de Ginebra.
- [5] ANIJOVICH, REBECA; MORA, SILVA. (2010). Estrategias de enseñanza: otra mirada al quehacer en el aula. - 1a ed.- Buenos Aires: Aique Grupo Editor. Coordinadores: Vicario S. Claudia Marina- Escorcia S. German “Constructores de conocimiento: Papert y su visión” 2020, Introducción- Bibliografía: pp. 18
- [6] ¿Por qué todos tienen que aprender a programar? Participá de la Hora del Código. [Entrada web] Recuperado de: <https://youtu.be/HrBh2165KjE?si=GXct5Dnchs1xBmMV>
- [7] Ciencias de la computación para el aula. Manual para docentes. 1º ciclo secundaria. Fundación Sadosky. Investigación y desarrollo en Tic. Program.ar. Pablo E. “Fidel” Martínez López, Federico Alois, Daniel A. Ciolek, Federico Martínez, Denise Pari, Pablo Tobia.. [https://program.ar/descargas/cc\\_para\\_el\\_aula-1er\\_ciclo\\_secundaria.pdf](https://program.ar/descargas/cc_para_el_aula-1er_ciclo_secundaria.pdf). pp15
- [8] Cuaderno para el docente. Actividades para aprender a programar. Segundo Ciclo de la Educación Primaria y Primero de la Secundaria. Volumen 1. Febrero 2017. pp14-20: <https://program.ar/descargas/manual-docente-descarga-web-v2017.pdf>
- [9] Coordinadores: Vicario S. Claudia Marina- Escorcia S. German “Constructores de conocimiento: Papert y su visión” 2020, Seymour Papert. Su teoría: pp. 48-49.

### Capítulo de libro

- FURMAN, M.; LARSEN, M.E. Y PRIMON, M. (2020) “Aprendizaje Basado en Problemas: ¿Cómo llevarlo al aula? Documento No 10. Proyecto Las preguntas educativas: ¿Qué sabemos de educación? Buenos Aires: CIAESA.
- PERRENOUD, PHILIPPE (2004) Diez nuevas competencias para enseñar. Invitación al viaje. Barcelona Grao.

### Marco Legal

- Ley de Educación Nacional 26,206 [Entrada web] Recuperado de: <https://www.argentina.gob.ar/sites/default/files/ley-de-educ-nac-58ac89392ea4c.pdf>
- Ley de Educación Provincial de Tucumán N° 3891 [Entrada web] Recuperado de: [http://www.bnm.me.gov.ar/redes\\_federales/snie/pais/tierra\\_del\\_fuego/producciones/docs/ley\\_8391.pdf](http://www.bnm.me.gov.ar/redes_federales/snie/pais/tierra_del_fuego/producciones/docs/ley_8391.pdf)
- Núcleos de Aprendizajes Prioritarios para Educación Digital, Programación y Robótica. Resolución CFE N° 343/18 (2018). Consejo Federal de Educación. [Entrada web] Recuperado de: [http://www.bnm.me.gov.ar/giga1/normas/RCFE\\_343-18.pdf](http://www.bnm.me.gov.ar/giga1/normas/RCFE_343-18.pdf)
- Colección Marcos Pedagógicos Aprender Conectados (2017): “Orientaciones pedagógicas” Ministerio de Educación de la Nación. [Entrada web] Recuperado de: [https://www.argentina.gob.ar/sites/default/files/orientaciones\\_pedagogicas\\_0.pdf](https://www.argentina.gob.ar/sites/default/files/orientaciones_pedagogicas_0.pdf)
- Colección Marcos Pedagógicos Aprender Conectados (2017): “Competencias de educación digital” Ministerio de Educación de la Nación [Entrada web] Recuperado de: [https://www.argentina.gob.ar/sites/default/files/competencias\\_de\\_educacion\\_digital\\_0.pdf](https://www.argentina.gob.ar/sites/default/files/competencias_de_educacion_digital_0.pdf)
- Colección Marcos Pedagógicos Aprender Conectados (2017): “Programación y Robótica: Objetivos de aprendizaje para la educación obligatoria” Ministerio de Educación de la Nación [Entrada web] Recuperado de: <https://www.educ.ar/recursos/132339/programacion-y-robotica-objetivos-de-aprendizaje-para-la-educacion-obligatoria>

- Material de curso de capacitación virtual autoasistido. Coordinación de Educación Digital. Ministerio de Educación. Gobierno de Tucumán. “Aprendiendo a programar con Pilas Bloques” clase n° 1 pp10, clase n°2 pp 4-5, pp12.
- Fernández Solo de Saldiva I. ( 2016) Juego serio: gamificación y aprendizaje. Comunicación y Pedagogía N°303 Recuperado de: <http://www.centrocp.com/juego-serio-gamificacion-aprendizaje/>
- Lego. Recuperado de <https://education.lego.com/es-es/>
- LEGO Mindstorms EV3: Una Revisión del Mejor Robot Educativo Recuperado de: <https://educacionrobotica.com/>
- Ciudad Autónoma de Buenos Aires. Argentina. Recuperado de: <http://program.ar/descargas/manual-docente-descarga-web-v2017.pdf>
- Publicación de la red social Facebook de la Escuela de Comercio N°4 - Batalla de Tucumán: [https://www.facebook.com/EscuelaDeComercioN4BatallaDeTucuman/posts/pfbid06NoQhvtZPUVWD9Grp9Qt\\_i828xWuc7iTRO971B197FheP2QoJdrfCVMrja6Ef6LWv1](https://www.facebook.com/EscuelaDeComercioN4BatallaDeTucuman/posts/pfbid06NoQhvtZPUVWD9Grp9Qt_i828xWuc7iTRO971B197FheP2QoJdrfCVMrja6Ef6LWv1)
- ScratchJr - Aprende. (2017). Recuperado de: <https://www.scratchjr.org/learn/interface>

Versión Preliminar

# Programación en Grupo usando Python+ABP

## Una Experiencia Enriquecedora en el Aula Universitaria

Natalia Colussi<sup>1</sup>, Pamela Viale<sup>2</sup>

<sup>1</sup> Departamento de Ciencias de la Computación, Licenciatura en Ciencias de la Computación

<sup>2</sup> Instituto Tecnológico de Diseño e Innovación, Tecnicatura en Inteligencia Artificial

Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario

{colussi,viale}@fceia.unr.edu.ar

### Resumen

En el presente artículo se analiza uno de los aspectos del proceso de aprendizaje de la programación mediante la conformación de grupos de trabajo. Mediante la aplicación del Aprendizaje Basado en Proyectos y Problemas (ABP) sobre un proyecto de programación grupal en Python, estudiaremos cómo funciona el trabajo grupal entre los estudiantes de la cátedra de Programación II en su segundo dictado (redictado) del primer año universitario de la carrera Licenciatura en Ciencias de la Computación. Los redictados en primer año recuperan y retienen a los estudiantes con dificultades para asimilar conceptos específicos del área y para sobrellevar los procesos tradicionales de evaluación universitarios. Se planteó un abordaje integral de la problemática, aplicando un ABP guiado y supervisado por el equipo docente, con objetivos y controles semanales. Empleando el lenguaje Python junto al módulo *streamlit* para generar automáticamente un *front-end*, los grupos desarrollaron las funcionalidades que responden a las consultas solicitadas (*back-end*). Se observará cómo la modalidad de trabajo grupal influye positivamente en el fortalecimiento del aprendizaje de la programación y permite desarrollar las llamadas *habilidades blandas (soft-skills)*. Los estudiantes retienen y afianzan saberes, generan comunidad de aprendizaje entre pares, y logran regularizar y aprobar la materia. El análisis de las encuestas a los estudiantes revela cómo se produce este proceso.

**Palabras clave:** Aprendizaje Basado en Proyectos y Problemas; Programación en Python; Didáctica en Ciencias de la Computación; Programando en Grupo.

### 1. Introducción

La enseñanza de la programación en el ciclo inicial universitario presenta desde hace ya unos años diversos desafíos. Por un lado, las carreras de informática han incrementado sus inscripciones y las cátedras de primer año se han superpoblado de alumnos. Por otro lado, la demanda de la industria y las actividades sociales que exigen o necesitan de cada vez más tecnología digital, generan una mayor área de vacancia laboral, y con ello una oportunidad para que muchas personas busquen desarrollarse en el área de la informática.

En este contexto, la Licenciatura en Ciencias de la Computación (LCC) en la Facultad de Ciencias Exactas, Ingeniería y Agrimensura (FCEIA), de la Universidad Nacional de Rosario (UNR) refleja en sus aulas esta realidad, explícitamente con un aumento de estudiantes en la inscripción a la carrera que se ha duplicado y triplicado en los últimos cinco años. Asimismo, los estudiantes secundarios provienen de diversas orientaciones -técnica, bachiller, ciencias económicas, etc.- generando una diversidad en cuanto a saberes y conocimientos previos. Sin embargo, los recursos docentes para dictar las materias disciplinarias en las carreras informáticas son escasos y los espacios áulicos de laboratorios con disponibilidad tecnológica actualizada no se han incrementado al ritmo de la matrícula. Todos estos factores contribuyen en diferente medida a que una gran mayoría de los estudiantes no logre aprobar las materias, lo cual al inicio de la carrera puede llevar a su abandono.

En este contexto, desde la UNR se ha planteado desde hace ya más de una década la posibilidad del redictado para las materias de primer año en las distintas carreras, como medida de retención de los estudiantes en el ámbito universitario, promocionando áreas estratégicas de desarrollo nacional. Considerando estas cuestiones, resulta relevante plantear innovaciones pedagógicas que permitan hacer frente a la realidad presentada, buscando garantizar una educación de calidad que permita a los y las estudiantes el acceso a los conocimientos disciplinares y a los saberes necesarios en la asignatura que se recusa, que en este caso, se trata de Programación II.

En este sentido y con el objetivo de ofrecer a los y las estudiantes una propuesta diferente a la transitada en el primer dictado de la cátedra, en el 2022 se diseñó una nueva propuesta didáctica que respondiera a las múltiples problemáticas subyacentes de los cursantes del redictado, relacionadas particularmente con la dificultad para comprender los conceptos disciplinares impartidos, los procesos de evaluación, y la adaptación al medio universitario. En función de la experiencia de las autoras con el redictado de Programación I de la LCC [1,2,3,4, 11] se pensó la estrategia didáctica de la cátedra del redictado de Programación II, tomando la experiencia en el uso de ABP [8,13] y la experiencia de los autores Frydenberg y Mentzer [7] en esta línea didáctica, donde se destacan los beneficios del ABP con Python para el desarrollo del pensamiento crítico, la creatividad, la aprehensión de saberes, y la colaboración entre los pares. También destacamos que el trabajo didáctico utilizado se nutre del enfoque de resolución de problemas de Polya [12] adaptada al diseño de programas por Thompson [14], teniendo como pilar fundamental al enfoque del Pensamiento Computacional de Wing [16], la estrategia pedagógica de Indagación [5] para apuntalar el aprendizaje autónomo, y la adquisición de las llamadas soft-skills mediante la incorporación de saberes profesionales tempranamente [9].

En el presente proyecto se analizan datos obtenidos durante los tres últimos años (2022-2024) durante los cuales se ha llevado adelante esta propuesta didáctica renovada. Con ello se pretende conocer cuáles han sido las percepciones y opiniones de los y las estudiantes que han participado de las propuestas de recursado, a fin de evaluar las mismas e informar futuros diseños pedagógicos.

## 2. Descripción de la Experiencia

### 2.1. Propuesta Didáctica

En las tres cohortes estudiadas, la metodología de trabajo ha sido similar. Los proyectos se desarrollaron de manera grupal a lo largo del cuatrimestre, una vez finalizado el dictado teórico de los conceptos de programación que se ejercitan con el lenguaje Python. Se establecen de antemano plazos y objetivos a cumplir, para favorecer la organización del trabajo y lograr concluir cada uno de los proyectos propuestos. Se busca de esta manera, integrar a los estudiantes mediante la actividad grupal, fortaleciendo saberes a partir del trabajo en conjunto y la discusión con pares, trabajando la motivación a partir de la realización de un proyecto que resulte atractivo visualmente y disciplinar, ya que permite evaluar e integrar saberes desarrollados en la primera etapa del cuatrimestre. También, parte del entusiasmo, se da a partir de la promoción (aprobación) de los temas vistos en Python al aprobar el proyecto. La implantación del ABP en el aula universitaria de primer año, se trabaja con instancias de supervisión y retroalimentación por parte del equipo docente, brindando guía y seguimiento a la realización de los proyectos de forma semanal, por el plazo de 6 semanas que dura la realización del mismo. A modo de cierre, se realiza una puesta en común, donde se ofrece una mirada crítica y reflexiva sobre el trabajo propio y el de los demás grupos.

Durante todo el proceso, se evalúan contenidos propios de la asignatura correspondiente a los conceptos como: a) el uso de estructuras de datos simples: listas, tuplas, y diccionarios; b) diseño, algoritmia, y modularizado del problema; c) la aplicación de la metodología específica de construcción de programas denominada: “La Receta Infalible” para programar [6]; y d) el desarrollo de soft-skills. En este trabajo, nos interesa conocer la opinión de los estudiantes frente a esta propuesta novedosa, ya que sus comentarios permiten avanzar en la revisión del diseño pedagógico del recursado.

### 2.2. Contexto

Las experiencias que aquí se presentan se han llevado adelante en el aula del redictado de Programación II de la carrera Licenciatura en Ciencias de la Computación perteneciente a la Facultad de Ciencias Exactas, Ingeniería y Agrimensura dependiente de la Universidad Nacional de Rosario por el lapso de tres años, 2022, 2023, y 2024. En el Redictado asisten en un 50% estudiantes que la cursan por primera vez, ya que provienen de la cursada del Redictado de Programación I que es correlativa con Programación II. El otro 50% corresponde a estudiantes que vuelven a cursar y provienen del primer dictado de la materia. La cantidad de estudiantes y docentes ha ido variando con el paso del tiempo. En la Tabla 1 se muestra un resumen de esta información y cuántos proyectos de programación se realizaron.

La materia Programación II corresponde al primer año de la carrera y es una de las materias disciplinarias de la misma. Los estudiantes para cursarla deben tener regularizado y/o aprobado Programación I, la primera materia específica de computación que cursan en la LCC. En esta última, se estudia el paradigma de funcional con el lenguaje Racket y el entorno de programación Dr. Racket desarrollado por Felleisen et.al. [6], donde se adquieren los saberes básicos de programación, vinculados a la división de problemas, modularización, elementos básicos: expresiones, tipos básicos, condicionales, definición y aplicación de funciones, recursión, interacción, y tipos compuestos: estructuras y listas.

Año	Cantidad	Masc./Fem	Edad (prom)	Equipo Docente	#Proyectos
2022	21	17/4	20 años	1 doc. teoría y 1 doc. de práctica.	9
2023	18	13/5	21 años	1 doc. teoría y 2 doc. de práctica.	9
2024	35	29/6	20 años	1 doc. teoría y 2 doc. de práctica.	18

**Tabla 1:** Resumen de cursadas Redictado de Programación II.

Estos conocimientos se suponen logrados una vez que comienzan a cursar Programación II y se avanzan entonces sobre los conceptos de la programación desde el paradigma imperativo, empleando los lenguajes Python y C. Se incluyen entonces temas como: tipo de datos y expresiones, entrada y salida, asignación, condicionales simples y múltiples, ciclos, recursión, manejos de archivos y datos estructurados como las tuplas, las listas y los diccionarios, junto a la manipulación de archivos. Sumando también a la definición de funciones, aplicación, parametrización y modularización de las soluciones algorítmicas, manejo de memoria dinámica y punteros.

### 2.3. La Experiencia

La experiencia se desarrolló a lo largo de las últimas semanas de clases. El diseño e implementación de la propuesta pedagógica del redictado de Programación II, consistió en la realización de un proyecto de programación grupal en Python, donde se diseña de una forma sencilla y práctica una aplicación web de consulta sobre un conjunto fijo de datos. Para ello se combinan módulos específicos para automatizar la generación de un *front-end* como Streamlit<sup>1</sup>, y módulos para procesar y graficar los datos, como pandas, matplotlib, etc., y todo el *back-end* de la aplicación se programa en Python.

Para la puesta en marcha del proyecto, los estudiantes reunidos en grupos de tres integrantes, recolectaron datos reales de repositorios públicos, siendo los dos primeros años, 2022 y 2023, los datos de empresa de alocación de viviendas conocida como AirBnb, y en el 2024 se introdujeron como opción nuevos datasets públicos de nuestro país, donde en este último caso, se trabajaron con datos sobre los combustibles y energías en Argentina, espacios de entretenimiento y esparcimiento de la ciudad de Buenos Aires, datos sobre los registros y las poblaciones aborigenes del país, y por último dato sobre las escuelas, en todos los niveles, de la provincia de Buenos Aires.

Una vez que los grupos tienen asignados los datasets sobre los cuales van a trabajar, cada grupo procede a la elaboración de un conjunto de 6 preguntas a realizar sobre los datos. Las preguntas se caracterizan por ser algunas de análisis dinámico (van a modificarse según la entrada del usuario) y otras serán estáticas (no dependerán de la entrada de datos). Las salidas a posteriori del procesamiento de los datos deben incluir tablas, gráficos de barra o de torta, y ubicaciones en un mapa. Todo el manejo de E/S se realiza mediante widgets que ofrece el módulo Streamlit y se programa desde Python incorporando el módulo mencionado e invocando a la funcionalidad que se necesita. Luego de este proceso de generación de preguntas, más el control, revisión y depuración de las mismas por parte de los docentes asignados a la supervisión de cada grupo, se comienza a trabajar sobre los aspectos de programación específicos para resolver cada proyecto.

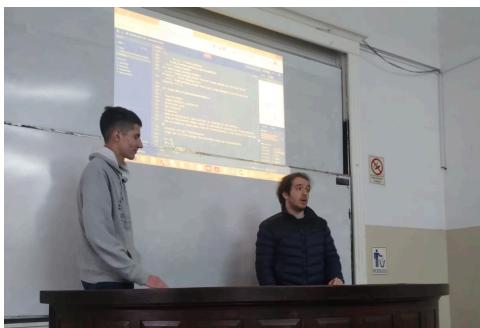
Semanalmente, se trabaja con objetivos que se asignan de manera general a todos los grupos, se publican en un documento en el aula virtual, y todos los estudiantes acceden a ellos al iniciar la semana. Los docentes que supervisan los grupos verifican que los estudiantes puedan cumplirlos y avanzar hacia la concreción del proyecto. Se asignan un control semanal para cada grupo, con un lapso de tiempo de unos 15 min a cada grupo, y un total de 5 o 6 grupos por docente.

<sup>1</sup> <https://streamlit.io/>

Los controles se llevan a cabo uno de los días de clase durante el horario de laboratorio. Si algún grupo requiere más tiempo de consulta y/o control, entonces se lo cita a consulta para resolver los problemas que tengan. Una vez cumplido el plazo y junto a la entrega del proyecto se realiza la exposición del mismo donde los estudiantes comparten los logros con el resto de la cátedra. En el año 2022 se hizo mediante una exposición sólo con los docentes, en el 2023 se realizó mediante un video expositivo, y en el 2024 se logró exponer en un aula con la asistencia del resto de los compañeros de cursada. En la Figura 1 y 2 se muestran estas instancias en el aula del laboratorio.



**Figura 1:** Control de proyectos en el aula del laboratorio



**Figura 2:** Exposición de Proyectos en el aula de clases

## 2.4. Documentación de la Experiencia

Con el objetivo de analizar la experiencia realizada y conocer las opiniones de los participantes, se diseñó y administró una encuesta a los estudiantes del redictado con preguntas abiertas y cerradas. Se relevaron 70 encuestas pertenecientes a las últimas tres cohortes desde el 2022 al 2024. Las encuestas fueron diseñadas para ser respondidas de manera individual al finalizar el proyecto de programación en coincidencia con la finalización del curso. Puntualmente, para este trabajo se presentará el análisis de las siguientes 5 preguntas:

- **Pregunta 1:** *¿Qué coincidencia y qué diferencias encontraron entre ustedes a la hora de desarrollar el/los programas?*
- **Pregunta 2:** *¿Cómo llevaron adelante el trabajo grupal?*
- **Pregunta 3:** *¿Cómo se desarrolló el proceso de corrección entre los miembros del grupo?*
- **Pregunta 4:** *Té pedimos que describas con una palabra, un adjetivo, o una frase, cómo te sentiste realizando el proyecto.*
- **Pregunta 5:** *¿En qué forma consideras que la estrategia de trabajo por proyecto ayudó a tu aprendizaje de la programación?*

Para procesar las respuestas se trabajó con la identificación de patrones emergentes en los datos, desde un enfoque mixto, cuantitativo y cualitativo, de forma tal que permita relevar las voces de los participantes con relación a:

- aspectos clave o fundamentales del funcionamiento del grupo sobre la realización de un proyecto común de programación,
- aspectos enriquecedores disciplinariamente de la experiencia para cada integrante del grupo, desde lo individual, formas de aprendizaje que los estudiantes logran vislumbrar tras la realización del proyecto.

Para el procesamiento y análisis de las respuestas se emplearon herramientas de inteligencia artificial, como ChatGPT (OpenAI), donde a partir de la consulta, suministrando la pregunta de la encuesta, el contexto y los objetivos de investigación, junto a algunas respuestas (sanitizadas), el asistente facilitó algunos patrones en los datos y categorías. Las sugerencias dadas por esta herramienta fueron validadas, recodificadas, y refinadas por las autoras, contrastando las planillas de respuestas, la revisión y clasificación manual de los datos previamente realizada. Se contabilizó entonces la frecuencia de aparición de cada categoría. Los resultados de dicho análisis se presentan en la sección siguiente, junto con un análisis de contenidos para cada respuesta, los cuales permitieron seleccionar extractos representativos que ejemplifican las diferentes categorías.

Existen otras herramientas muy conocidas, como *Atlas.ti*, que ayudan en el procesamiento automático de datos. Sin embargo, las autoras de este artículo no disponen de una licencia de dicha herramienta, razón por la cual se optó por el uso de otras opciones gratuitas. De todas formas, la herramienta nombrada anteriormente hace uso de OpenAI, según lo publicado en su página web.

### 3. Resultados

**Pregunta 1:** *¿Qué coincidencia y qué diferencias encontraron entre ustedes a la hora de desarrollar el/los programas?:* Para esta pregunta se procesaron 108 respuestas, de las cuales se desprendieron las siguientes categorías y porcentajes que se muestran en la Tabla 2.

Categoría	% TOTAL	Subcategoría	% TOTAL
Coincidencias entre los miembros del grupo	<b>52%</b>	Respecto a la forma “estilo” de programar y la lógica del programa; diseño del front-end de proyecto, formas de construir las funciones, etc.	<b>25%</b>
		Facilidad de entendimiento, comunicación y cómo llevar adelante el proyecto, formas de cooperar entre los miembros del grupo.	<b>27%</b>
Diferencias entre los miembros del grupo	<b>48%</b>	En la implementación técnica: la elección de la estructura de datos principal y el “estilo” de la programación, y estrategias de solución, junto a la documentación del programa con el uso de la “La Receta Infallible”.	<b>34%</b>
		En la organización y comunicación	<b>14%</b>

**Tabla 2:** Categorías halladas en las respuestas de las preguntas 1 analizadas en las encuestas.

En cuanto a las **coincidencias dentro de los miembros del grupo**, son parejos los porcentajes que señalan los estudiantes respecto al estilo o la forma de programar entre ellos y corresponde a que aprenden a programar en el curso, muchos no tienen experiencia previa en el mundo de la programación, ingresan a la carrera sin haber programado nunca, por lo tanto, van adquiriendo una forma de programar similar y ajustada a las indicaciones que se les da durante el cursado, vinculadas a la modularización del problema, definir funciones que sean testeables, manejar la E/S desde el main, etc. A su vez, dado que el programa es una forma de comunicar una solución, el entendimiento dentro del grupo también es posible como primer punto de partida para aunar a los integrantes. Respecto a la categoría de **diferencias entre los miembros del grupo**, señalamos que las diferencias técnicas también reflejan la heterogeneidad del curso, en el mismo conviven estudiantes que cursan por primera vez la materia provenientes del redictado de Programación I, y estudiantes que vuelven a cursar la materia. Esto último, puede provocar enfoques dispares a la hora de elegir, por ejemplo, la/s estructura/s de datos principal/es con la cual se van a procesar los datos del dataset dentro del proyecto. Pero en las respuestas se puede observar que estas diferencias, en realidad, generan discusiones enriquecedoras, ya que ponen de manifiesto distintas soluciones que luego son debatidas entre los miembros del grupo y eligen aquella que les resulta más conveniente para resolver el proyecto. Cabe señalar que estos debates y las elecciones y decisiones que de ellos se desprendan son supervisadas por los docentes tutores de cada grupo, lo cual termina de validar la decisión final o reorientarla si fuese necesario. La Figura 3 muestra extractos de las encuestas referidos a esta pregunta y cómo los estudiantes señalaron las coincidencias y diferencias en el desarrollo del programa.

<p>"Por lo general coincidíamos en el diseño y disposición de la pantalla. Teníamos diferencias en las formas y caminos que utilizábamos para resolver los problemas. Esto nos ayudó a atacar los problemas desde diferentes ángulos y poner lo mejor de ambas soluciones" (respuesta de un estudiante proyectos 2022)</p>	<p>"Cómo coincidíamos la forma en llevar adelante el proyecto, en la interpretación de la información y las consignas a realizar. Quizás como una diferencia podría ser el enfoque de cada uno en la realización de una misma función, es decir la misma función de igual funcionalidad pero distinta construcción" (respuesta de un estudiante proyectos 2023)</p>
<p>"Lo que más nos chocó en un principio fue las distintas ideas que teníamos cada uno y las distintas formas que teníamos de pensar sobre las mismas preguntas. Tuvimos que aprender a buscar siempre una idea en conjunto y de resolver problemas trabajando los 3 juntos" (respuesta de un estudiante proyectos 2024)</p>	
<p>"Definiría al grupo como uno con poca comunicación. Hacíamos líneas de código y avisábamos por el grupo de whatsapp y listo. Nos corregíamos las líneas de código escritas sin avisar y sin el por qué. Esto si bien no generó conflicto, me hizo sentir frustrado. Me cuesta comunicarme con la gente por lo que no comunique como me sentía" (respuesta de un estudiante proyecto 2024)</p>	

**Figura 3.** Extractos de las respuesta 1 de los distintos encuestados

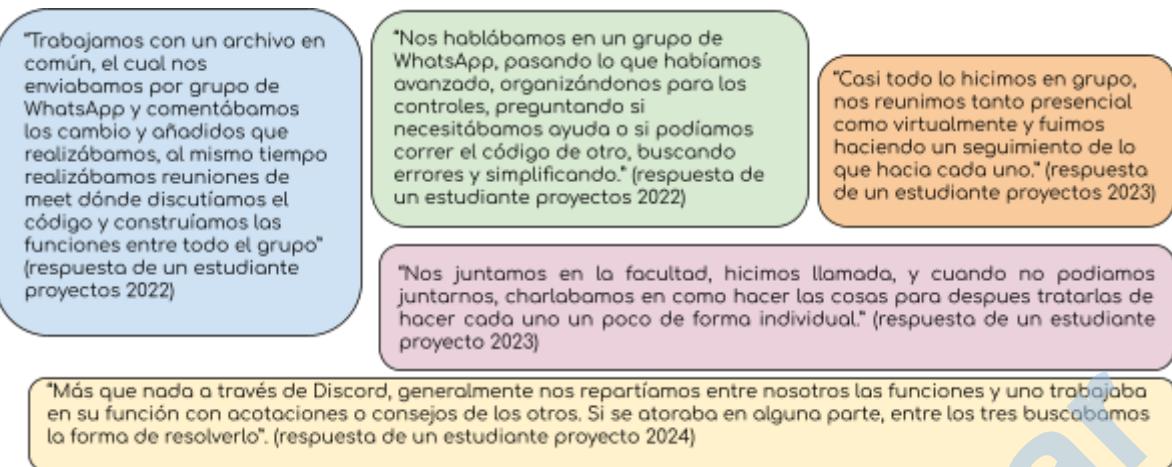
#### **Pregunta 2:** ¿Cómo llevaron adelante el trabajo grupal?

Para esta pregunta se procesaron 102 respuestas, de las cuales se desprendieron las siguientes categorías y porcentajes que se muestran en la Tabla 3.

Categoría	% TOTAL	Subcategoría	%RELATIVO
De forma Asíncrona	<b>40%</b>	Comunicación por mensajería instantánea como la plataforma whatsapp con un trabajo más independiente pero con atención a mantener el trabajo grupal compartiendo los avances y la comunicación.	<b>57%</b>
		Comunicación y transferencia de archivos por medio de plataformas colaborativas, como repositorios de archivos como Github, Drive, One, etc. con un debate de ideas entre los miembros del grupo para luego dividirse el trabajo.	<b>33%</b>
		Comunicación virtual o presencial del trabajo para coordinar el proyecto.	<b>10%</b>
De forma Síncrona	<b>60%</b>	Comunicación síncrona por meet, discord, zoom, live share de visual studio, etc. Trabajo colaborativo, debate de ideas, plantear un tema y tratar de resolverlo juntos, etc.	<b>60%</b>
		Juntarse presencial, en facultad, después de clases, también en casas particulares. Uso de la comunicación por whatsapp y herramientas colaborativas como google Drive, Github, etc.	<b>40%</b>

**Tabla 3:** Categorías halladas en las respuestas de las preguntas 2 analizadas en las encuestas.

En cuanto a las formas de realizar el proyecto, los estudiantes revelan que han trabajado mayoritariamente de **forma sincronizada**, tanto sea llevando adelante todos juntos el proyecto en todo momento, con algunas técnicas de pair programming o juntándose para programar en plataformas como Discord para consultarse y resolver entre todos las dudas que podían llegar a plantearse en el momento en un verdadero trabajo colaborativo. En la **sincronicidad, en un cálculo relativo**, se mencionan en mayor porcentaje a la plataforma de comunicación conexión virtual como Discord, zoom, live share, etc., mientras que en una menor medida aparecen las formas de comunicación presencial o de mensajería instantánea como WhatsApp. Para la **forma de trabajo asíncronas**, la mensajería instantánea es la opción más descripta para comunicarse, transferir archivos, e intercambiar dudas y avances sobre el trabajo. En segundo lugar, aparecen las herramientas más específicas para compartir archivos y trabajar de forma colaborativa. Atribuimos esta diferencia principalmente porque el celular es un dispositivo que siempre está disponible a la hora de brindar una respuesta rápida a un problema, al trabajar con tareas divididas o secuenciadas, quien prosigue el trabajo iniciado por otro necesita consultas rápidas de lo generado por el compañero para continuarlo o para componerlo con la “parte” que debe realizar. Aunque, si la respuesta se tornará más extensa, entonces se recurre a las plataformas virtuales que aparecen en menor porcentaje de uso en este caso de asincronicidad en la comunicación. La Figura 4 muestra extractos de las encuestas referidos a esta pregunta y cómo los estudiantes expresaron la forma en la que llevaron adelante la tarea grupal de programar.



**Figura 4.** Extractos de las respuesta 2 de los distintos encuestados

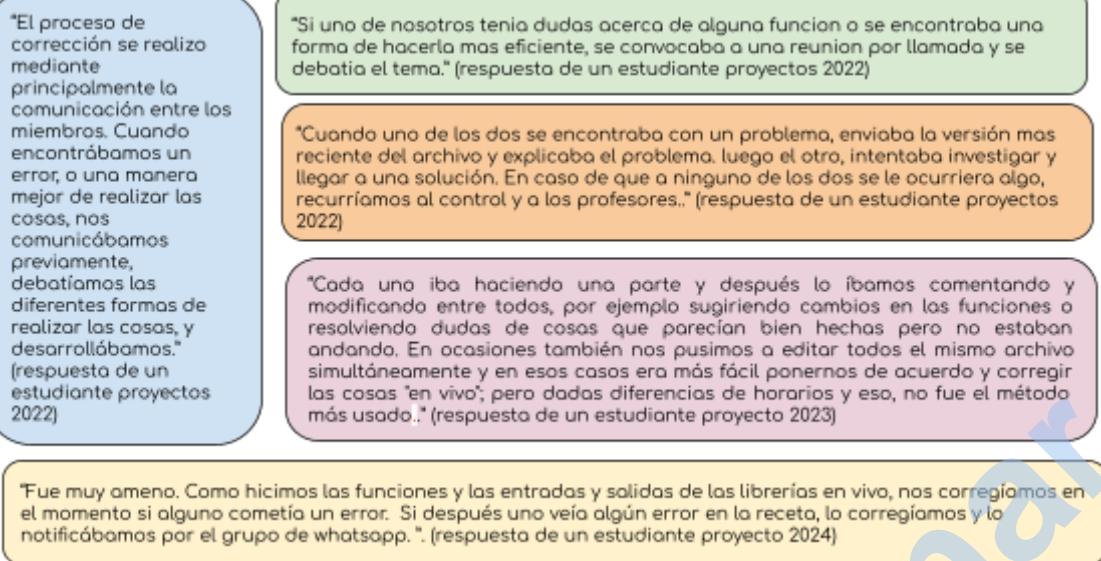
**Pregunta 3:** ¿Cómo se desarrolló el proceso de corrección entre los miembros del grupo?

Para esta pregunta se procesaron 66 respuestas, de las cuales se desprendieron las siguientes categorías y porcentajes que se muestran en la Tabla 4.

Categoría	Descripción	TOTAL
Corrección de forma Asíncrona	Intercambio de archivos de forma asíncrona. Revisión a destiempo, para luego hacer una puesta en común.	39%
Corrección de forma Síncrona	Realizar reuniones, utilizar aplicaciones de mensajería (como WhatsApp y Discord) y llamadas para corregir problemas y debatir mejoras.	49%
Combinación de formas tanto síncronas como asíncronas	Realizar controles y revisiones del código de independiente, para luego corregir todos juntos.	12%

**Tabla 4:** Categorías halladas en las respuestas de las preguntas 3 analizadas en las encuestas.

En cuanto a las formas de corrección, resulta mayormente mencionado **el modo síncrono**. Los estudiantes logran corregir y entender de mejor manera las modificaciones que deben realizar al código, tanto para que deje de producir errores o se corresponda con el comportamiento esperado, como para hacerlo más eficiente, al juntarse y resolverlo entre todos. Esta pregunta y su respuesta están en consonancia con la pregunta 2, donde se señala también la forma síncrona como la elegida para desarrollar el proyecto. **El modo asíncrono** también tiene un porcentaje importante de elección, atribuido a que los avances son en general a destiempo una vez que se dividieron el trabajo, se produce y genera código que se deja luego a un compañero para que este lo revise, o también aparece esta forma de trabajo, en la composición de las partes del programa, quien utiliza el código de otro encuentra problemas que debe corregir y procede a depurarlo. En estos últimos casos, también señalamos que aparecen **combinaciones de formas de corregir**, ya que muchos señalan que recurrían en esta instancia a llamadas virtuales a sus compañeros para resolver el problema en el momento entre todos. Destacamos también que, muchas de las expresiones que recuperamos de las respuestas señalan que esta etapa se realizó de forma respetuosa y amena, con debate pero sin enojo o descalificaciones hacia los compañeros. Esto es fundamental para trabajar en el grupo, y se pudo observar que estas consideraciones hechas en clase por los docentes fueron tenidas en cuenta. La Figura 5 muestra extractos de las encuestas referidos a esta pregunta y cómo los estudiantes expresaron la forma en la que corrigieron el código que producían.



**Figura 5.** Extractos de las respuesta 3 de los distintos encuestados

**Pregunta 4:** Te pedimos que describas con una palabra, un adjetivo, o una frase, cómo te sentiste realizando el proyecto.

Para esta pregunta se procesaron 82 respuestas y se clasificaron las ocurrencias de los adjetivos que usaron los estudiantes. Esto nos permitió generar una nube de palabras que se muestra en la Figura 6. Las respuestas más usadas fueron: "cómodo", en primer lugar, luego: "acompañado", "divertido", "satisfecho", "entretenido", "conforme", "entusiasmado", y "a gusto". Este juego de palabras demuestra lo positiva que resulta la experiencia para quienes la realizan, y denotan un clima de trabajo a la hora de dedicar tiempo a esta actividad académica, la cual retroalimentan el deseo de realizarlas, el compromiso y concreción del proyecto en sí mismo.

**Figura 6:** Nube de palabras con las sensaciones de los estudiantes en relación al trabajo realizado



**Pregunta 5:** ¿En qué forma consideras que la estrategia de trabajo por proyecto ayudó a tu aprendizaje de la programación?

Para esta pregunta se procesaron 94 respuestas, de las cuales se desprendieron las siguientes categorías y porcentajes que se muestran en la Tabla 5.

Categoría	Descripción	% TOTAL
Brinda una práctica real, profesional.	Experimentar situaciones más cerca de la realidad laboral de una persona que se dedica profesionalmente a la programación, ayuda a entender el manejo de proyectos grandes, a comunicar con un programa.	24%
Trabajo en equipo y diferentes enfoques al problema	La importancia de trabajar en conjunto, aprender de otros, y repartir tareas. Los diferentes enfoques para resolver problemas es beneficioso para el aprendizaje, ayuda a manejar la frustración.	55%
Indagación y aprendizaje autodidacta	Investigar y aprender por uno mismo, ganando autonomía.	15%
Estimulante el desafío	El reto en sí mismo permite desarrollar la imaginación, el pensamiento, la paciencia, la adaptación, etc.	6%

**Tabla 5:** Categorías halladas en las respuestas de las preguntas 5 analizadas en las encuestas.

En este caso, los estudiantes señalan que el **trabajo grupal**, y podríamos afirmar también que para muchos la conformación de un verdadero equipo, resulta ser el principal factor de ayuda para aprender a programar o mejorar su forma de programar. El empuje que genera el grupo y el compartir de los saberes, en la diversidad de estudiantes, de posibles soluciones, los debates, etc., es lo que termina enriqueciendo el trabajo final y nutre a cada uno de sus miembros. El aprendizaje entre pares es muy efectivo en estos casos. También se menciona en segundo lugar el poder **practicar la realización de un programa para un problema o desafío similar a uno del mundo real**, este tipo de desafío genera conocimientos que complementan a los aprendidos con los problemas de la práctica, tienen otra dimensión, otra necesidad computacional, otro tipo de integración de saberes más complejos, que permiten afirmar los conceptos fundamentales dados. En tercer lugar, aparece la **indagación**, el tener que aprender desde la ayuda que pueda brindar una librería nueva, probar y ensayar soluciones, investigar cómo usar una plataforma de programación específica, etc. y lograr concretar este objetivo personal o colectivo como grupo, habilita a explorar otras librerías luego, “se sabe cómo hacerlo”, y se puede construir conocimiento a partir de haber logrado esta primera instancia. Por último, aparece la **opción del reto, el desafío** en sí mismo como estímulo del aprendizaje, “poder lograrlo”, poniendo en juego otras funciones mentales como la abstracción, la creatividad, tener que “imaginar” cómo escribir una función específica, fortalece y asegura el aprendizaje. La Figura 7 muestra extractos de las encuestas referidos a esta pregunta y cómo los estudiantes expresaron la forma en la que el ABP les permitió aprender a programar.

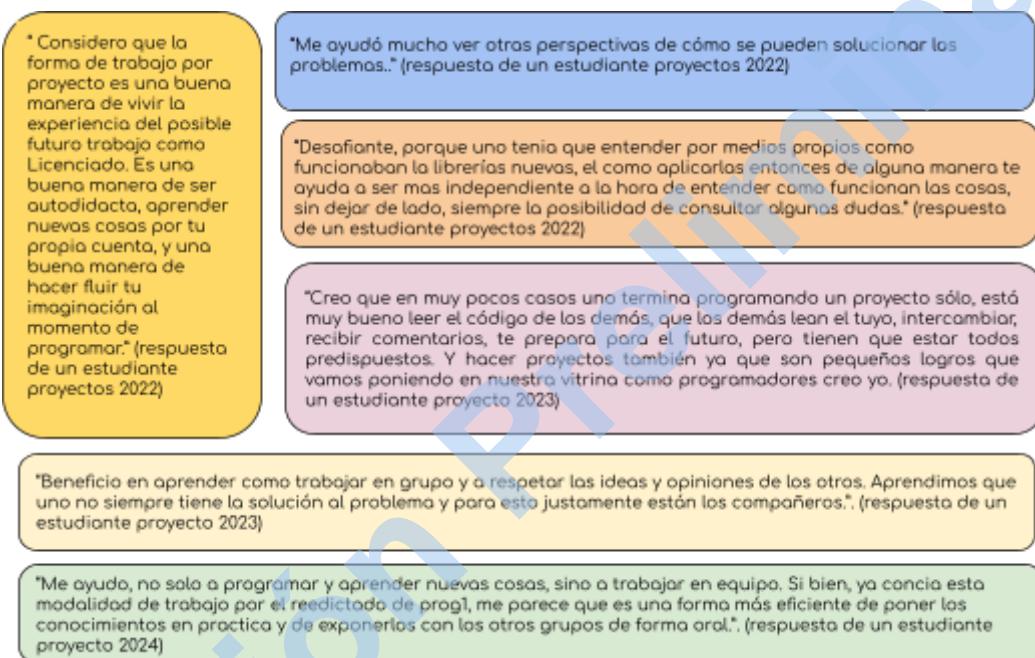


Figura 7. Extractos de las respuesta 5 de los distintos encuestados

## Conclusiones

Los resultados muestran que los estudiantes logran reforzar y ganar saberes con relación a la programación y su estilo a partir del trabajo grupal. Este tipo de experiencia mejora no solo las habilidades técnicas, sino que fomenta el desarrollo personal, a través de la comunicación efectiva, preparándolos para los desafíos del mundo real en el ámbito de la programación y el desarrollo del software. Se logra conectar con el mundo laboral y la necesidad del “aprendizaje continuo”, aprender algo nuevo y aplicarlo, explicar, aprender, debatir ideas, y consensuar con otras propuestas y soluciones; teniendo en cuenta que el desarrollo del software debe producirse a partir del trabajo de muchas personas, que interactúan e intervienen en la producción del programa.

El trabajo grupal en programación no solo permite a los estudiantes desarrollar habilidades técnicas, sino que también enriquece su experiencia individual y colectiva, fomentando la colaboración, el aprendizaje mutuo y la creación de soluciones más eficaces. La integración del Aprendizaje Basado en Problemas (ABP) en el curso del redictado de Programación II, utilizando Python, ha demostrado ser entonces una experiencia enriquecedora para estudiantes y docentes. Este enfoque educativo ha permitido a los alumnos embarcarse en un proceso de aprendizaje significativo, donde enfrentan y resuelven problemas complejos. La adopción de métodos de evaluación alternativos ha contribuido a un enfoque más integral y formativo. En conjunto, estos elementos evidencian el potencial del ABP para transformar la enseñanza en el ámbito de la programación y genera para las autoras un área de trabajo e investigación para mejorar el proceso de enseñanza y aprendizaje en el aula universitaria.

## Referencias

1. Colussi, N., Viale, P.: Actividades de Programación Grupales para Primer año de la Licenciatura en Ciencias de la Computación - Experiencias Didácticas en el Aula. Versión Extendida. En Jornadas de CyT de la UNR, (JorCyT), (2019).
2. Colussi, N., Viale, P., Monjelat, N.: Vidriera de proyectos: una modalidad de evaluación posible en tiempos de virtualidad. En Actas de II WITE - TRANSFORMACIÓN DIGITAL. DESAFÍOS DE LA EDUCACIÓN SUPERIOR, 1–10, (2021)
3. Colussi, N., Monjelat, N., & Viale, P., El ABP en el redictado de materias de programación: una experiencia en la educación superior. Actas de las Segundas Jornadas de Didáctica de las Ciencias de la Computación (JADiCC 2022). Universidad Nacional del Nordeste. (2022)
4. Colussi, N., & Monjelat, N. Estrategias didácticas para el aprendizaje y la enseñanza del pensamiento computacional en el nivel académico universitario. Libro de actas - XXVIII Congreso Argentino de Ciencias de la Computación - CACIC 2022. Universidad Nacional de la Rioja. (2022)
5. Dostál, J.: Inquiry-based instruction: Concept, essence, importance and contribution. PhD thesis, Palacký University, Olomouc, Czech Republic, 2015.
6. Felleisen, M., Findler, R., Flatt, M., Krishnamurthi, S. (2001). How to Design Programs: An Introduction to Programming and Computing. MIT Press, USA
7. Frydenberg, Mark, and Kevin Mentzer. "From Engagement to Empowerment: Project-Based Learning in Python Coding Courses." EDISG Conference, Information Systems & Computing Academic Professionals, 2020.
8. García Martín, J., Perez Martínez, J.: Aprendizaje basado en proyectos: método para el diseño de actividades. Revista Tecnología, Ciencia y Educación, 5, 37-63 (2018).
9. Keogh S., Bradnum J., Anderson E. Improving professionalism in first year computer science students: Teaching what can't be taught. En Proceedings of the 3rd Conference on Computing Education Practice (CEP '19), (2019). 1-4.
10. Martinez Lopez, P.: Las Bases de la Programación. Publicado electrónicamente por la Universidad Virtual de Quilmes, La Plata, (2013).
11. Monjelat, N.; Colussi, N.; Viale, P.: Introducción a la Programación en carreras terciarias y universitarias de Computación: estado del arte en el contexto argentino. Jornadas de Ciencia y Técnica de la UNR, (2021).
12. Polya, G.: How To Solve It: A New Aspect of Mathematical Method. Princeton: University Press, (1973).
13. Sánchez, P., Blanco, C.: Implantación de una metodología de aprendizaje basada en proyectos para una asignatura de Ingeniería del Software. En *Actas XVIII JENUI 2012*, Ciudad Real, Universidad Nacional de Cantabria, España (2012).
14. Thompson S.: Where do I begin? A problem solving approach in teaching functional programming. LNCS (vol 1292). Springer, Berlin, Heidelberg, (1997).
15. Wachenchauzer Rosita and et.al. Algoritmos y Programación I: Aprendiendo a programar usando Python como herramienta. Apuntes de cátedra. Online: <https://drive.google.com/file/d/0B0KKEIBDHL7tdEQ3bFZ2M3VrZzA/view?resourcekey=0-uMkBoDSnoZjQWxDvcBwQ4g>
16. Wing, J.M.: Computational thinking. Commun. ACM , 49(3), 33-35, (2006).