

Pensamiento computacional, programación creativa y ciencias de la computación para la educación.

**Reflexiones y experiencias desde América
Latina**

*Francisco Silva Garcés
Ivan terceros
(Coordinadores)*



EDICIONES
CIESPAL



Red de investigación de
**Conocimiento
Hardware y
Software Libre**

2021

Pensamiento computacional, programación creativa y ciencias de la computación para la educación.

Reflexiones y experiencias desde América Latina

Francisco Silva Garcés

Ivan terceros

(Coordinadores)

Esta publicación es una iniciativa del Centro Internacional de Estudios Superiores de Comunicación para América Latina (CIESPAL) y la Red de Investigación de Conocimiento, Software y Hardware Libre (RICHSL). Para garantizar su calidad ha pasado por un proceso de revisión por pares ciegos.

CIESPAL

Centro Internacional de Estudios Superiores de Comunicación para América Latina

Av. Diego de Almagro N32-133 y Andrade Marín • Quito, Ecuador

Teléfonos: (593 2) 254 8011

www.ciespal.org

<https://ediciones.ciespal.org/>

Red de Investigación de Conocimiento, Software y Hardware Libre

+593 9 994491992

info@conocimientolibre.ec

Edición

Camilo Molina

Corrección de estilo

Dolores Chacón

Diagramación

Diego S. Acevedo A.

ISBN: 978-9978-55-203-2

Ediciones Ciespal, 2021

Los textos publicados son de exclusiva responsabilidad de sus autores.



Reconocimiento

CC BY

Esta licencia permite la redistribución, comercial y no comercial, siempre y cuando la obra no se modifique y se transmita en su totalidad, reconociendo su autoría.

Acciones en la UNNE para promover el pensamiento computacional mediante la enseñanza de la programación

Gladys N. Dapozo

Cristina L. Greiner

Raquel H. Petris

Introducción

Existe consenso generalizado a nivel global sobre las competencias requeridas para el siglo XXI, así como de la importancia de la formación en las áreas STEM (Science, Technology, Engineering and Mathematics). En particular, en América Latina y el Caribe existe el desafío de fomentar la educación STEAM dado que el desempeño de los jóvenes latinoamericanos en el área de ciencias y la inversión en innovación y desarrollo es menor que en otras regiones del mundo, además, el número de científicos e ingenieros es muy bajo en proporción a la economía y población de nuestros países (Ochoa, Valenzuela y Márquez, 2018). También es un hecho conocido que la demanda de las empresas de software y servicios informáticos es creciente e insatisfecha, mientras disminuye la cantidad de estudiantes y graduados en carreras de informática. Los países a nivel mundial desarrollan políticas para atender esta realidad.

Por otra parte, si bien es marcada la tendencia mundial en educación de fomentar el pensamiento computacional y la enseñanza de la programación desde niveles iniciales, una dificultad importante está vinculada con la formación de los docentes. Muchos de ellos no cuentan con las habilidades necesarias para enfrentar este desafío y, en muchos casos, tampoco están preparados con los nuevos enfoques de la didáctica de la programación (The Royal Society, 2012).

En el marco de esta problemática, en la Universidad Nacional del Nordeste (UNNE), universidad pública gratuita con amplia influencia en la región noreste del país, se desarrollan diversas acciones para promover el pensamiento computacional, mediante la enseñanza de la programación.

En este trabajo se describen estas acciones, que abarcan: la promoción de vocaciones para incrementar el interés de los jóvenes para realizar carreras TIC, la incorporación de nuevos métodos para facilitar la enseñanza-aprendizaje de la programación en la universidad, favoreciendo la retención y permanencia de los alumnos, y la formación de los docentes de los niveles preuniversitarios con las competencias tecnológicas y pedagógicas necesarias para abordar el desafío de la enseñanza de la programación en las escuelas.

Antecedentes y contexto general

Distintos reportes de países del primer mundo (Computer Science Teachers Association, 2010) (The Royal Society, 2012) describen las dificultades de la enseñanza de la computación, entre las que se mencionan: la preocupante falta de profesores capacitados para enseñar los temas propios de la disciplina, la falta de esquemas de educación continua para mantener actualizados a los profesores, y el escaso reconocimiento a las Ciencias de la Computación como una disciplina académica rigurosa, cuya enseñanza es imprescindible para mejorar las perspectivas profesionales de todos los estudiantes.

El informe de The Royal Society of London señala que estas situaciones generan un “círculo vicioso”, que se ilustra en la Figura 1.

Para romper este círculo vicioso es necesario entender que la enseñanza de la computación es muy distinta a la mera enseñanza del manejo de software y herramientas tecnológicas concretas. Cuando se hace referencia a la alfabetización digital, esta debe integrar también nociones sobre los lenguajes de las computadoras. En este marco, tanto la programación como el pensamiento computacional resultan relevantes para el aprendizaje. Al comprender su semántica y su lógica en la resolución de problemas, los alumnos también se preparan para entender y cambiar la realidad (Ripani, 2017).

Figura 1: Círculo vicioso de la enseñanza de la computación. Fuente: *¿Shut down or restart? The way forward for computing in UK schools*



Incorporar en la enseñanza una visión acertada de las Ciencias de la Computación implica fomentar el desarrollo de habilidades abstractas de pensamiento computacional y no sólo el uso de herramientas informáticas (Sadosky, 2013).

Además, en este nuevo paradigma de enseñanza, se considera importante no solo formar sino también inspirar a los estudiantes, por

lo cual el enfoque didáctico debería poner énfasis en elementos clave tales como la motivación, la resolución de problemas del mundo real, el trabajo en equipo y la participación activa de los estudiantes (President's Council of Advisors on Science and Technology-PCAST, 2010).

Por otra parte, las universidades argentinas enfrentan problemáticas preocupantes con respecto a la retención y la permanencia de los alumnos ingresantes a las carreras científico-tecnológicas en general, y a las vinculadas con la disciplina Informática en particular. En general, la deserción no sólo responde a cuestiones socioeconómicas. Entran en juego la diversidad cultural, problemáticas de orden personal, la carencia de conocimientos, la falta de estrategias cognitivas y metacognitivas de abordaje de los nuevos temas, la planificación adecuada del tiempo de estudio. También es necesario tener en cuenta los problemas generados por las diferencias respecto de aprendizajes y conocimientos previos (López, Olmo, Reyes, & Fernández, 2013).

La carrera Licenciatura en Sistemas de Información de la Universidad Nacional del Nordeste (UNNE), con una trayectoria de más de 30 años, tiene como objetivo la formación de profesionales para el desarrollo de sistemas. Sin embargo, esta carrera no es ajena a las situaciones mencionadas anteriormente, es habitual el retraso o abandono de los alumnos en el primer año de la carrera, compatible con el bajo nivel de graduación y la prolongación de la duración de los estudios de grado por encima del tiempo esperable, detectado en el sistema universitario argentino (García, 2015).

Conscientes de esta situación, el equipo docente lleva adelante distintas estrategias para mejorar la iniciación en la programación, siendo el aprendizaje de este campo de estudio una de las razones del retraso de los estudiantes.

Por su parte, a nivel nacional, la iniciativa Program.AR, impulsada por la Fundación Sadovsky, tiene como objetivo llevar la enseñanza y el aprendizaje de las Ciencias de la Computación a la escuela Argentina. Incluye múltiples aspectos relacionados con la difusión y

popularización de la disciplina, la generación de contenidos escolares y la formación docente, entre otros.

Para la formación docente, Program.AR ha elaborado un curso que incluye contenidos, herramientas y actividades especialmente diseñadas para llevar la enseñanza de la programación a las escuelas. El enfoque pedagógico se basa en el aprendizaje por indagación (Dostál, 2015), una metodología de enseñanza-aprendizaje a través de la cual los estudiantes deben encontrar soluciones a un problema a partir de un proceso de investigación, usualmente poniendo énfasis en el trabajo cooperativo y en la extracción de ideas a través de la reflexión sobre las actividades realizadas para construir la solución. El enfoque por indagación facilita la participación activa de los estudiantes en la adquisición del conocimiento y ayuda a desarrollar el pensamiento crítico y la capacidad para resolver problemas (Sampson, Grooms, & Walker, 2010). Esta modalidad es clave en el enfoque de la didáctica Program.AR, y se explica detalladamente en (Martínez López, 2013).

A nivel de las políticas públicas, en Argentina, en el año 2018, el Consejo Federal de Educación ha declarado de importancia estratégica para el sistema educativo argentino la enseñanza y el aprendizaje de la programación desde niveles iniciales, y aprobó los Núcleos de Aprendizajes Prioritarios (NAP) de Educación Digital, Programación y Robótica. Se espera que todos los establecimientos educativos del país incorporen estos contenidos, en forma gradual, considerando las particularidades de sus contextos, realidades y políticas educativas, con lo cual cobra especial importancia contar con ofertas formativas específicas para la formación docente en Educación Digital, Programación y Robótica.

Promoción de Vocaciones TIC

Con el objetivo de acercar la práctica de la programación a los estudiantes de los últimos años del nivel medio de educación, se

dictaron talleres de programación de juegos y animaciones mediante el uso de la herramienta lúdica Alice.

Desde el año 2013 al 2016, la Facultad de Ciencias Exactas de la UNNE firmó un convenio con la Fundación Sadosky para llevar adelante el programa Vocaciones en TIC. El convenio establecía la conformación de equipos de docentes y alumnos avanzados de la carrera universitaria Licenciatura en Sistemas de Información, que fueron capacitados por especialistas de la Fundación.

El programa consistía en visitas de estos equipos a las escuelas de nivel preuniversitario, particularmente de los 2 últimos años, con el objetivo de despertar interés en los jóvenes para estudiar carreras vinculadas con las TIC, a través de talleres de capacitación para elaborar juegos y animaciones con el software Alice. Al finalizar el taller se brindaba una charla sobre la importancia de la formación profesional en Informática, las características de la profesión y las posibilidades laborales, dado que en general los alumnos desconocen las posibilidades de la formación en TIC. También en cada visita se instaba a los alumnos a participar con sus producciones en una competencia organizada en la universidad al final del período de visitas, previéndose en este evento premios para distintas categorías. Al finalizar cada visita se solicitaba a los alumnos la respuesta a un cuestionario, orientado a conocer características de los alumnos en relación a las tecnologías, y el posible interés en continuar sus estudios en una carrera informática.

De las respuestas de los alumnos participantes en los talleres de programación con Alice, se infiere que los alumnos que expresaron interés en seguir una carrera vinculada con la Informática se destacaban por poseer habilidades tecnológicas en relación con la computadora, adquiridas principalmente en forma autónoma, observándose que estas habilidades eran menores en el caso de las mujeres. Respecto de los factores que influyen al momento de elegir una carrera para los alumnos que indicaron Muy Probable que continuaran una carrera de Informática, los más relevantes a la hora de inclinarse por la Informática fueron: “Conocimiento sobre lo que es la carrera” (81%), “Experiencia

que tengo con computadoras” (81%), “Interés en los juegos” (74%), “Interés en el hardware” (74%) e “Interés en la Programación” (72%). Esta vinculación fortalece la idea de que motivar tempranamente a los alumnos del nivel medio en el conocimiento práctico de las tecnologías informáticas incrementará su interés por seguir carreras vinculadas con la informática. Estos resultados fueron publicados en (Dapozo, Greiner, Pedrozo Petrazzini, & Chiapello, 2014).

Nuevas metodologías para enseñar programación en la universidad

En la carrera Licenciatura en Sistemas de Información de la Facultad de Ciencias Exactas de la UNNE se desarrollaron actividades orientadas a estimular el pensamiento computacional mediante la incorporación de herramientas lúdicas de programación (Lightbot, Scratch, PilasBloques). El propósito era motivar a los alumnos y ofrecer una base de ideas y técnicas que permitan apoyar una mejor comprensión de los conceptos básicos de programación que forman parte de los contenidos de la carrera. Las actividades fueron tomadas del curso Introducción a la Programación y su Didáctica, que la Fundación Sadosky propone para la formación de docentes a través de la iniciativa Program.Ar.

En la propuesta formativa se insistió en los conceptos de abstracción, referido a pensar una estrategia antes de intentar la resolución de los ejercicios, la descomposición del problema en partes, traducida a la creación de bloques o procedimientos en la solución diseñada, y en la legibilidad de la solución, mediante la definición de nombres representativos de los procedimientos creados. También se diseñó e implementó un método de resolución de problemas, que se consideró favorecedor para lograr la transición del entorno lúdico al entorno formal de la programación.

En 2018 y 2019 se perfeccionó el método de resolución de problemas, y se diseñó una estrategia de enseñanza con el propósito de mejorar la

comprensión de los conceptos básicos de programación, basada en la programación por bloques, consistente en:

- a. Definición de un método de resolución de problemas
- b. Desarrollo de un conjunto de actividades lúdicas, basadas en la programación por bloques, aplicando el método de resolución de problemas.
- c. Aplicación del método de resolución de problemas en el desarrollo de los contenidos propios del curso, referidos a algoritmos y estructuras de datos, utilizando para la práctica un lenguaje de programación imperativa, el lenguaje de programación C, de uso habitual en este tipo de instrucción.

El método de resolución de problemas, basado en (Martínez López, 2013), combina herramientas conceptuales y herramientas del lenguaje, promoviendo el pensamiento computacional que se requiere para la programación. Consiste de los siguientes pasos: 1) Idear una estrategia de solución, y explicitarla; 2) Expresar la estrategia mediante una división en bloques o módulos; 3) Declarar y nombrar adecuadamente cada bloque de modo que exprese la tarea que realiza; 4) Definir cada uno de los bloques que expresan tareas, mediante instrucciones primitivas.

En una primera etapa del curso de programación, se propone a los estudiantes la resolución de problemas, planteados como desafíos que a modo de juego deben resolver con programación en bloque, utilizando Scratch o PilasBloques, aplicando el método de resolución de problemas descripto en el párrafo anterior.

Luego de la etapa de actividades lúdicas continúan las actividades formales del curso, que incluyen la representación de las soluciones en el lenguaje de programación C. Para la resolución de las actividades prácticas se utilizó el mismo método aplicado en la resolución de las actividades lúdicas. Para la introducción de cada uno de los conceptos básicos, propios de un curso de programación inicial (modularización,

estructuras de control, uso de parámetros), se referenció previamente la actividad lúdica que incluye el concepto. Los alumnos participantes expresaron su satisfacción con la realización de las actividades lúdicas.

En el ciclo lectivo 2019 participaron de una encuesta 240 estudiantes, de los cuales el 80% respondió que la estrategia de enseñanza les resultó útil para entender los conceptos básicos de programación. A estos estudiantes se les solicitó que indiquen cuáles de las herramientas del lenguaje (Secuencia, Repetición, Selección, Procedimiento) fueron mejor asimiladas mediante las actividades realizadas con la herramienta PilasBloques. El 74% indicó Secuencia y Repetición, un 54% marcó Selección y un 44% indicó Procedimiento. Del contraste entre la información obtenida de las encuestas y los resultados de la primera evaluación de los contenidos propios del curso, se observó que, de aquellos alumnos que apreciaron la estrategia, el 53% aprobó el examen. De estos, el 45% obtuvo una calificación de 8 o superior, en una escala de 1 a 10. Los resultados de esta experiencia se publicaron en (Dapozzo, Greiner, Petris, Godoy Gugliemone y Espíndola, 2019).

De los resultados obtenidos en las distintas experiencias se infiere que la introducción a la programación a través de actividades lúdicas y con la aplicación de un método de resolución de problemas resultó positiva para la comprensión de los conceptos propios de la programación en el nivel introductorio universitario, facilitando la incorporación de los conceptos y técnicas formales del área Algoritmos y Estructuras de Datos.

Formación de docentes de niveles educativos preuniversitario

Capacitación en didáctica de la Programación

Con el objetivo de formar a los formadores, se dictaron cursos de didáctica de la programación a los docentes de los niveles preuniversitarios de distintas disciplinas.

La capacitación docente se realizó en el marco de la Iniciativa Program.Ar de la Fundación Sadosky, a través de una convocatoria destinada a universidades nacionales que ofrecieran los equipos docentes y la infraestructura necesaria para llevar adelante el dictado del curso La Programación y su didáctica (Factorovich y Sawady O'Connor, 2017), destinado a los docentes de los niveles primario, secundario y terciario, de cualquier área de conocimiento, priorizando a docentes del área TIC y de Matemáticas. El curso tenía las siguientes características:

Objetivos: Que los docentes comprendan la relevancia de enseñar Ciencias de la Computación, que conceptualicen la noción de programa, que incentiven a sus propios alumnos a ser creadores de programas y no sólo usuarios de aplicaciones hechas por terceros, que planifiquen la solución a un problema de programación, dividiendo en subproblemas.

Contenidos: Importancia de enseñar Ciencias de la Computación. Comandos (acciones) y valores (datos). Procedimientos. Noción de programa y autómata. División en tareas más pequeñas. Identificación de patrones. Resolución de problemas. Repeticiones simples y condicionales. Alternativas condicionales. Parámetros.

Duración del curso: 100 horas reloj, 70 de ellas presenciales, distribuidas en 14 clases.

Para la incorporación de los conceptos se aplicó la estrategia pedagógica de aprendizaje por indagación (Dostál, 2015), aprendizaje basado en problemas (Torp & Sage, 2007) y se siguieron las metodologías de enseñanza de programación propuestas en (Martínez López, Bonelli, & Sawady O'Connor, 2012). Se utilizaron las herramientas Lightbot, Scratch, PilasBloques, Alice, Gobstone, Robots didácticos N6 Max.

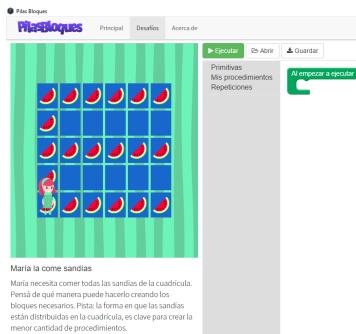
El equipo docente para el dictado (instructores) estaba integrado por profesores de las asignaturas iniciales de programación de la Facultad de Ciencia Exactas y Naturales y Agrimensura de la UNNE (FaCENA-UNNE). Los instructores fueron capacitados por especialistas de la Fundación Sadosky, quienes enfatizaron sobre la importancia del aprendizaje por descubrimiento. Los especialistas

brindaron permanente apoyo a los instructores durante el dictado del curso Programación y su didáctica, y monitoreaban el avance del curso a través de los informes que, por cada clase, remitían los instructores.

Cada clase consistía en presentar una actividad, en algunos casos de tipo *un-plugged* (sin computadora), planteando una situación problemática o desafío, con el objetivo de que los docentes participantes, en el marco del aprendizaje por indagación, fueran descubriendo los conceptos requeridos, e intenten resolver el desafío planteado utilizando las herramientas específicas, bajo la supervisión y guía de los instructores. Las distintas soluciones se ponían en común con todo el grupo y los instructores animaban el debate para analizar en forma constructiva la estrategia de solución presentada, el uso de la descomposición del problema en partes y la adecuada denominación de esas partes, para lograr la legibilidad de la solución, aspecto fuertemente incentivado.

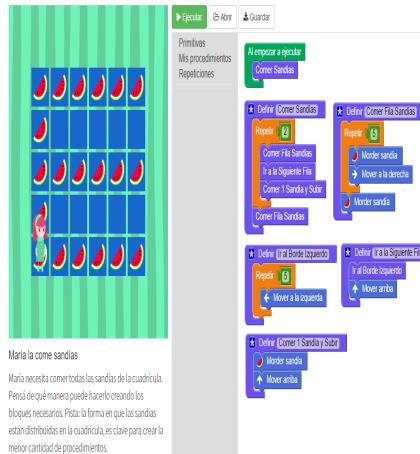
Cabe destacar que estas actividades del curso fueron diseñadas con un enfoque lúdico para incrementar la motivación de los docentes participantes, y cada una de ellas estaba diseñada para incorporar conceptos y técnicas de programación predeterminadas. En la Figura 2 se muestra, a modo de ejemplo, la actividad denominada “María la come sandía”. Presenta un desafío que consiste en hacer que María coma todas las sandías de la cuadrícula, utilizando los recursos conceptuales dados. Se insiste a los docentes participantes que primero analicen el problema y luego definan una estrategia de solución, descomponiendo el problema en partes (bloques) y nombrando adecuadamente las partes. La solución se codifica en el panel de la derecha de la herramienta, mediante bloques de instrucciones. La Figura 3 muestra una solución particular de este desafío, en la que se utilizan procedimientos (bloques) y estructuras de control (Repetir).

Figura 2. Ejemplo de actividad con PilasBloques. Fuente: (Factorovich, Sawady O'Connor, 2017).



Fuente: (Factorovich, Sawady O'Connor, 2017).

Figura 3. Solución sugerida.



Fuente: elaboración propia.

Siguiendo el enfoque didáctico propuesto, se trabajó para que los docentes participantes incorporen un método para la resolución de los desafíos, haciendo énfasis en la definición y justificación de una estrategia para llegar a la solución, y en la legibilidad de la misma, antes que la codificación eficiente que resuelve el problema.

Para verificar el cumplimiento de los objetivos del curso, los docentes participantes debían realizar actividades de enseñanza de programación con sus propios alumnos de los establecimientos educativos en los que se desempeñan, basadas en los contenidos y herramientas brindados en el curso, utilizando el enfoque didáctico propuesto. Para ello, previamente, debían elaborar una planificación y presentarla en un formulario con los datos del docente, de la escuela y de los estudiantes (nivel, grado) y un esquema previsto para cada clase: Objetivo de aprendizaje, actividades, tipo de actividad (sin computadora o con el uso de herramientas), y cierre, enfatizando

conceptos y generando la oportunidad de retroalimentación de parte de los estudiantes. También se solicitaba explicitar cómo se daría el proceso de aprendizaje por descubrimiento.

Para aprobar el curso los docentes participantes debían presentar la planificación de actividades destinadas a sus alumnos e implementarlas bajo la supervisión de los instructores. Para realizar estas prácticas se acordaban días y horarios con las instituciones educativas.

Un total de 142 docentes, pertenecientes a los niveles educativos primario, secundario y terciario, cumplieron las condiciones de aprobación en los cursos dictados desde el 2015 al 2017. Mayoritariamente los docentes provenían del nivel secundario, tal como se muestra en la tabla 1.

Tabla 1. Cantidad de docentes por año de dictado y nivel educativo.

Año/ Nivel	Primario	Secundario	Terciario	Total
2015	6	43	11	60
2016	17	18	3	38
2017	17	18	9	44
2018	5	14	3	22
Total	45	93	26	164

Cabe resaltar también que, como resultado de la capacitación, los docentes consideran mayoritariamente que cambiaron su percepción respecto de la enseñanza de la programación y el 7% restante considera que confirmó sus ideas sobre la enseñanza de la programación. Reconocen que el curso les permitió actualizar los métodos de enseñanza tradicionales hacia nuevas herramientas y métodos basadas en herramientas lúdicas y un enfoque didáctico más entretenido y, por tanto, más motivador (Dapozzo, Petris, Greiner, Company, & Espíndola, 2018).

Generación de nuevas ofertas educativas de formación docente

Con el objetivo de capacitar a los docentes de los niveles preuniversitarios en los temas vinculados con los NAP de programación y robótica, aprobados por el Consejo Federal de Educación, organismo de concertación y coordinación de la política educativa nacional, se diseñó e implementó la Diplomatura Universitaria en Programación y Robótica Educativa, buscando acompañar las políticas públicas orientadas a incorporar en la educación básica conceptos y herramientas en el manejo de tecnologías de programación y robótica.

Esta oferta permitió, además, satisfacer la demanda de los docentes de la región, quienes muestran especial interés en una formación sobre pensamiento computacional para lograr motivar a los niños y jóvenes.

La oferta tiene los siguientes objetivos institucionales:

- Estar en concordancia con las políticas públicas orientadas a incorporar en la educación básica conceptos y herramientas para lograr ciudadanos activos y críticos de este mundo tecnológicamente intensivo, mediante la formación de los docentes de nivel preuniversitario en el manejo de tecnologías de programación y robótica, adecuadas al nivel en que se desempeñan.
- Dar continuidad a las actividades que se realizan en la Facultad de Ciencias Exactas de la UNNE, y que se iniciaron con el dictado del curso de didáctica de la programación, promovido por la Fundación Sadosky.

La propuesta tiene como objetivos específicos desarrollar capacidades y competencias pertinentes, para que los docentes puedan:

- Comprender la relevancia de la incorporación en las escuelas de los temas propios de las Ciencias de la Computación como medio para contribuir con la formación de ciudadanos que entiendan el mundo tecnológico de lo cotidiano.

- Entender y aplicar técnicas básicas de programación y de robótica para desenvolverse adecuadamente en los nuevos contextos educativos.
- Apropiarse de un método de resolución de problemas para facilitar la enseñanza de la programación, método que puede ser extendido a otros ámbitos.
- Aplicar e integrar nuevos enfoques didácticos que promuevan el aprendizaje activo y colaborativo.
- Proponer y diseñar actividades de enseñanza innovadoras que ofrezcan retos y desafíos motivadores para los estudiantes.
- Fortalecer en sus estudiantes la capacidad de abstracción, más allá del uso de herramientas, brindando la posibilidad de aplicar los conocimientos en diferentes dispositivos y sistemas.

La Diplomatura Universitaria en Programación y Robótica Educativa contempla una carga horaria de 200 horas reloj. Los contenidos se distribuyen en 5 módulos presenciales de 40 horas reloj cada uno, que se muestran en la Tabla 2, con actividades de apoyo a la presencialidad a través de la plataforma de UNNE Virtual.

Tabla 2. Módulos Diplomatura Universitaria en Programación y Robótica Educativa

Módulo	Denominación	Carga horaria
1	Programación visual basada en bloques	40 hs.
2	Programación de placas Arduino	40 hs.
3	Fundamentos de Programación	40 hs.
4	Programación de microcontroladores	40 hs.
5	Enfoques didácticos para la programación	40 hs.

Cada módulo presenta una metodología de enseñanza propia, pero siempre propiciando la creatividad y el trabajo colaborativo.

En el módulo 1 se desarrollan en primer lugar actividades prediseñadas para incorporar los conceptos básicos de programación y un enfoque de resolución de problemas que se aplicará a lo largo del trayecto formativo. Las actividades se presentan como desafíos que se deben resolver, con un enfoque lúdico que propicia la participación. La segunda parte del módulo se orienta a la programación de actividades propias utilizando el lenguaje de programación Scratch.

En el módulo 2 se utilizan las metodologías de aprendizaje “aprender haciendo” (*learning by doing*) y “aprendizaje basado en proyecto” (*Project-based learning*). Se diseñan y programan circuitos, de complejidad incremental, utilizando placas Arduino y sus diferentes componentes, con el propósito de crear proyectos de robótica. Se imparten conocimientos elementales de electrónica para dar soporte al armado de los circuitos. A lo largo del curso los alumnos elaboran un proyecto que integra los contenidos impartidos, y que deben exponer al final del curso.

En el módulo 3 se resuelven actividades prácticas de complejidad creciente en las que se ponen en juego la capacidad de abstracción y de resolución de problemas, respetando las restricciones que imponen los lenguajes de programación.

Para llevar adelante las actividades del módulo 4 se aplica el enfoque didáctico de Aprendizaje Basado en Proyectos (ABP), con el objetivo de lograr las competencias claves, a través de un flujo de trabajo probado en diversos contextos y condiciones diferentes. Se propone trabajar con microcontroladores y realizar prácticas de IoT, utilizando plataformas *open source* disponibles en la nube. Se aplican lenguajes de programación para el desarrollo de aplicaciones orientados a expresar ideas, consumir datos de terceros e intercambio de información.

En el módulo 5 se diseña un proyecto adecuado al nivel educativo, al contexto y destinatarios de la propuesta pedagógica en la que cada docente desarrollará sus actividades. El mismo debe contener los fundamentos de la aplicación del proyecto, las metas y objetivos a alcanzar, el diseño de actividades áulicas para la enseñanza de

la programación y la robótica, utilizando los enfoques didácticos abordados en este módulo. Se trata de potenciar las experiencias prácticas de los docentes en sus espacios para introducir los conceptos de las ciencias de la computación, las relaciones entre las competencias y los criterios de evaluación en los distintos niveles del sistema de educación, estableciendo directrices orientativas para facilitar el desarrollo de estrategias metodológicas que permitan trabajar por competencias en el aula.

Cada módulo finaliza con una evaluación de pares, por medio de rúbricas especialmente diseñadas para cada instancia. Las opiniones y sugerencias que se realizan entre pares son sumamente enriquecedoras y constructivas, afianzando los proyectos y mejorando las propuestas.

El dictado para la primera cohorte de esta oferta formativa se desarrolló entre marzo y agosto de 2019. Luego de un proceso de selección de más de 300 interesados, se seleccionaron docentes de las áreas de Tecnología y Matemática, conformando una matrícula de 75 docentes. En la tabla 3 se muestra la composición de los docentes por género y nivel educativo. El nivel secundario es mayoritario con un 63% de participación, siendo las docentes las más interesadas en este tipo de propuesta con un 60%.

Tabla 3. Docentes por nivel educativo y género

Nivel Educativo	Varones	Mujeres	Total	% Nivel
Docentes inactivos	5	6	11	15%
Primario	8	5	13	17%
Secundario	17	30	47	63%
Terciario		4	4	5%
Total	30	45	75	
% Género	40%	60%		

Al finalizar las actividades se realizó una encuesta a los docentes. De esta se destaca que el 93% considera viable introducir los conceptos

de programación y robótica en sus establecimientos educativos y que un 38 % ha puesto en práctica con sus alumnos conceptos y técnicas de programación y robótica, en tanto que otro 38% indica que lo logró a medias. Los resultados de esta experiencia se publicaron en (Dapoz y otros, 2019).

Conclusiones

En consonancia con la tendencia mundial de promover el pensamiento computacional como competencia clave para el desempeño de las personas en un mundo inmerso en las tecnologías de la información y la comunicación, en Argentina se realizan importantes esfuerzos a nivel de las políticas educativas para introducir en las escuelas la alfabetización digital, y como parte de esta, la programación de computadoras.

En este marco, en la UNNE se han llevado a cabo distintas acciones para promover el pensamiento computacional, a través de la enseñanza de la programación, en los distintos niveles del sistema educativo: primario, secundario, terciario y universitario.

Estas acciones han contribuido a que los jóvenes de las escuelas secundarias conozcan la programación, mediante la codificación de juegos y animaciones, conocimiento orientado a promover vocaciones TIC para incrementar la cantidad de profesionales del área.

La realización de estas actividades que incluyeron metodologías de enseñanzas novedosas dio lugar a cambios significativos dentro de la propia institución. Los docentes de los cursos iniciales de programación en la carrera de grado de Informática incorporaron como metodología de enseñanza la programación por bloques y el enfoque lúdico, con el objetivo de consolidar los conceptos básicos de programación y motivar a los alumnos, favoreciendo la permanencia y avance de los estudiantes en la universidad.

Por otra parte, generar cambios de fondo en las escuelas y hacer efectiva la incorporación del pensamiento computacional como

una competencia clave en la formación de los niños y jóvenes, exige la capacitación de los docentes. En este sentido, las actividades de formación docente, realizadas en la región de influencia de la UNNE, han impactado positivamente en la formación de los mismos. Los docentes participantes entendieron la importancia y la necesidad de esta capacitación, respondieron con mucho entusiasmo y compromiso con las propuestas pedagógicas, y las trasladaron al aula. Pudieron comprobar que las actividades de programación y de robótica incrementan notoriamente la motivación de los alumnos. Siendo esta el motor principal para un aprendizaje efectivo.

Por otra parte, la experiencia adquirida por el equipo docente llevó a la formalización de una oferta educativa de nivel universitario destinada especialmente a la formación de docentes de todos los niveles educativos preuniversitarios en programación y robótica.

Los resultados obtenidos dan cuenta de un avance significativo en el objetivo de promover el pensamiento computacional como competencia clave para el desempeño en la sociedad de la información. La temática está instalada en las instituciones y las políticas educativas están orientadas hacia ese objetivo. El panorama es alentador, dado que en particular los niños y jóvenes se sienten especialmente motivados cuando se les presentan actividades que ponen en juego el pensamiento computacional para la resolución de problemas.

Referencias bibliográficas

- Computer Science Teachers Association. (2010). *Running On Empty. The failure to teach K-12 Computer Science in the Digital Age.* USA.
- Dapozzo, G. N., Greiner, C. G., Pedrozo Petrazzini, G. O., & Chiapello, J. A. (2014). Investigación para fortalecer actividades de promoción y retención de alumnos en carreras de Informática. *XX Congreso Argentino de Computación.* San Justo Buenos Aires: UNLaM.

- Dapozzo, G. N., Greiner, C. L., Petris, R. H., Godoy Gugliemone, M. V., & Espíndola, M. C. (2019). Enseñanza de programación en la universidad. Estrategias basadas en programación por bloques. En S. Pech, M. Prieto, J. García, & E. Orozco, *Innovation an Practice in Education* (págs. 89-98). Ciudad Real. España: CIATA.org.
- Dapozzo, G. N., Medina, Y., Petris, R., Vallejos, S., Espíndola, M. C., Sambrana, I., Lencina, B. (2019). Oferta educativa en programación y robótica para docentes. *Simposio Argentino de Educación en Informática (SAEI)* (págs. 145-157). Salta. Argentina: SADIO.
- Dapozzo, G. N., Petris, R. H., Greiner, C. L., Company, A. M., & Espíndola, M. C. (2018). Formación docente para incorporar la programación en las escuelas. *Actas de las XXIV Jornadas sobre Enseñanza Universitaria de la Informática* (págs. 31-38). Barcelona. España: AENUI.
- Dostál, J. (2015). *Inquiry-based instruction: Concept, essence, importance and contribution*. Univerzita Palackého v Olomouci.
- Factorovich, P. M. y Sawady O'Connor, F.A. (2017). *Actividades para aprender a Program.AR*. Obtenido de: <http://program.ar/descargas/manual-docente-descarga-web-v2017.pdf>
- García, A. (2015). Políticas institucionales para mejorar la retención y la graduación en las universidades nacionales argentinas. *Debate Universitario*, 4(7), 7-24.
- López, M. F., Olmo, P., Reyes, K., & Fernández, E. F. (2013). La programación lúdica como estrategia de articulación entre niveles. *VIII Congreso de Tecnologías en Educación y Educación en Tecnologías*. Santiago del Estero: UNSE.
- Martínez López, P. E. (2013). *Las Bases Conceptuales de la Programación. Una nueva forma de aprender a programar*. Quilmes: UNQui.
- Martínez López, P. E., Bonelli, E. A., & Sawady O'Connor, F. A. (2012). El nombre verdadero de la programación. Una concepción de la enseñanza de la programación para la sociedad de la información. *Anales del Simposio de la Sociedad de la Información SSI 2012* (págs. 1-23). La Plata. Buenos Aires.: SADIO.

- Ochoa, L. A., Valenzuela, A., Estela, D., Márquez, F. (2018). La indagación como estrategia para la educación STEAM. Organización de Estados Americanos. Disponible en: <https://tinyurl.com/y9ptbgs1>
- President's Council of Advisors on Science and Technology-PCAST. (2010). *Report to the President. Prepare and Inspire: K-12 Education in Science, Technology, Engineering, and Math (STEM) for America's Future*. EEUU: PCAST.
- Ripani, M. F. (2017). *Competencias de Educación Digital*. Buenos Aires. Argentina.: Ministerio de Educación de la Nación. Obtenido de <http://planied.educ.ar/wp-content/uploads/2017/09/Competencias-o5.pdf>.
- Sadosky, F. (2013). *Una propuesta para refundar la enseñanza de la computación en las escuelas argentinas*. Buenos Aires. Argentina.: Fundación Sadosky.
- Sampson, V., Grooms, J., & Walker, J. P. (2010). Argument-driven inquiry as a way to help students learn how to participate in scientific argumentation and craft written arguments: An exploratory study. *Science Education*, 217-257.
- The Royal Society. (12 de 01 de 2012). *Shut down or restart? The way forward for computing in UK schools*. Obtenido de <https://royalsociety.org/-/media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf>
- Torp, L., & Sage, S. (2007). *El aprendizaje basado en problemas*. Buenos Aires. Argentina: AMORRORTU.