

# Programación por Bloques en cátedras de Ingeniería en Informática

Hernán C. Ahumada<sup>1</sup>, Daniel A. Rivas<sup>1</sup>, Oscar E. Quinteros<sup>1</sup>, Cecilia E. Gallardo<sup>1</sup>, Nelson A. Contreras<sup>1</sup>, Marta del V. Miranda<sup>1</sup>, Javier Favore<sup>1</sup>, María V. Póliche<sup>1</sup>

<sup>1</sup>Fac. de Tecnología y Ciencias Aplicadas - Universidad Nacional de Catamarca  
Maximio Victoria 55, 4700 Catamarca, Argentina  
{hcahumada, darivas, oequinteros, ceciliagallardo, nacontreras, mvmiranda, jfavore, vpoliche}@tecno.unca.edu.ar

## Resumen

Los datos de rendimiento académico de la Facultad de Tecnología y Ciencias Aplicadas indican que la materia Programación I es la que mayor cantidad de cursadas requiere para alcanzar la condición de alumno regular en la carrera Ingeniería en Informática. Tal situación da cuenta de la dificultad de los alumnos para desarrollar las capacidades básicas del Pensamiento Computacional y de los fundamentos de la Programación. Para revertir, en el corto y mediano plazo, la problemática expuesta se implementaron prácticas de enseñanza alternativas incorporando como recurso didáctico algunas herramientas de programación por bloques.

**Palabras clave:** Pensamiento Computacional, Programación por bloques.

## Contexto

Esta línea de trabajo se viene llevando a cabo en el marco de un proyecto de investigación acreditado por SECyT - UNCA dentro del Programa de Incentivos mediante Resolución rectoral 375/2018, denominado “*Estudio del impacto de enseñar mediante técnicas de programación por bloques las nociones*

*básicas de programación a alumnos de las cátedras Fundamentos de Informática (1° año) y Programación I (2° año) de la carrera Ingeniería en Informática de la Facultad de Tecnología y Ciencias Aplicadas (FTyCa) de la Universidad Nacional de Catamarca (UNCa)*”, que se desarrolla en el Departamento de Informática de la Facultad de Tecnología y Ciencias Aplicadas (Universidad Nacional de Catamarca).

## Introducción

Wing (2006) expresa que “el Pensamiento Computacional es el proceso de pensamiento involucrado en la formulación de problemas y expresión de las soluciones en una forma que pueda ser efectivamente llevada a cabo por un agente de procesamiento de información”.

Para Olabe (2015), “el Pensamiento Computacional es una metodología basada en la implementación de los conceptos básicos de las ciencias de la computación para resolver problemas cotidianos, diseñar sistemas domésticos y realizar tareas rutinarias”.

La Sociedad Internacional para la Tecnología en Educación (ISTE) y la

Asociación de Docentes en Ciencias de la Computación (CSTA) (ISTE and CSTA, 2011), establecen una definición operativa del Pensamiento Computacional (PC) como un proceso de solución de problemas que incluye, entre otras, las siguientes características:

- Formular problemas de manera que permitan usar computadoras y otras herramientas para solucionarlos.
- Organizar datos de manera lógica y analizarlos.
- Representar datos mediante abstracciones, como modelos y simulaciones.
- Automatizar soluciones mediante pensamiento algorítmico (una serie de pasos ordenados).
- Identificar, analizar e implementar posibles soluciones con el objeto de encontrar la combinación de pasos y recursos más eficiente y efectiva.
- Generalizar y transferir ese proceso de solución de problemas a una gran diversidad de éstos.

Según Brennan y Resnick (2012) el Pensamiento Computacional involucra tres dimensiones en las que se puede evaluar si el estudiante adquiere la capacidad de aplicarlas en la resolución de problemas.

*Dimensión 1: Conceptos Computacionales.* Son aquellos conceptos que permiten implementar la estrategia de solución a problemas. Tales conceptos son: comandos, secuencias de comandos, procedimientos, repetición simple, alternativa condicional, repetición condicional, sensores, variables, parametrización e interactividad.

*Dimensión 2: Prácticas Computacionales.* Se refiere a la habilidad de aplicar los conceptos computacionales siguiendo criterios de buenas prácticas en Programación. Entre esas prácticas computacionales están: la descomposición en subproblemas, utilización de procedimientos, legibilidad del programa, reutilización.

*Dimensión 3: Perspectivas Computacionales.* Perspectiva de expresar la estrategia de solución, perspectiva de trabajar colaborativamente para encontrar la solución, perspectiva de preguntar si es posible dar solución a otros problemas desde un punto de vista computacional.

En años recientes se han presentado varias herramientas para motivar y facilitar el aprendizaje inicial en programación a niños y jóvenes. La gran mayoría de ellas adoptan el enfoque de programación por bloques. Entre las más difundidas se encuentran Scratch (Resnick et. al, 2009) y Alice (Dann et. al, 2011).

En entornos de programación por bloques las instrucciones están representadas por bloques. Un programa se construye arrastrando y encastrando bloques en el orden apropiado para un determinado fin. En este modo de programar los conceptos abstractos tienen una representación visual. Además se tiene la ventaja de que el programa escrito está libre de errores de sintaxis tan frecuentes en otros lenguajes de programación. Esto permite a los usuarios focalizarse en la creación del programa. Es por ello que las herramientas de programación por bloques son cada vez más utilizadas en los cursos de iniciación a la programación.

En nuestro país, desde el año 2013 la

Fundación Sadosky lleva adelante la iniciativa Program.AR que promueve la inclusión del aprendizaje significativo de Ciencias de la Computación las escuelas de todo el territorio nacional. Para ello lleva adelante entre otras actividades, cursos de capacitación docente y desarrollo de material y recursos didácticos (Fundación Sadosky, 2013). En tal contexto, han desarrollado la aplicación *Pilas Bloques* que provee un entorno de programación por bloques. En *Pilas Bloques* se incluyen más de 40 actividades (desafíos) que abordan los principales conceptos de programación con niveles crecientes de dificultad.

Los desafíos incluidos en *Pilas Bloques* han sido diseñados para adoptar una metodología de enseñanza basada en problemas (Torp & Sage, 1998) que aplique una didáctica de la programación por indagación (Sanzo et. al, 2017). Para esta estrategia de enseñanza la indagación autodidacta es fundamental un rol activo del alumno, siendo el docente el que guía y asiste el proceso de aprendizaje del alumno.

En la carrera Ingeniería en Informática que se dicta en la Facultad de Tecnología y Ciencias Aplicadas de la Universidad nacional de Catamarca, la materia Programación I es la que mayor cantidad de cursadas requiere para alcanzar la condición de alumno regular. Lo cual constituye una evidencia de la dificultad para desarrollar las capacidades básicas asociadas al Pensamiento Computacional.

Problemática similar a la aquí planteada ha sido consignada por docentes investigadores de Universidades tanto de nuestro país como del exterior. Tal el caso de la Universidad Nacional de San Luis (Zúñiga, 2014), y de la Escuela de Ingeniería Informática de la chilena

Universidad de Valparaíso (Muñoz, 2015). En ambos trabajos se expone a las herramientas de programación por bloques como un recurso útil para enseñanza en los primeros cursos de carreras de grado en Informática.

## **Líneas de Investigación, Desarrollo e Innovación**

La línea de investigación aquí presentada trata de incorporar un novedoso enfoque para la enseñanza de nociones básicas de programación a estudiantes de Ingeniería en Informática.

El trabajo de investigación comprende el estudio de las herramientas y actividades apropiadas para introducir a conceptos de programación, el diseño de los instrumentos de recolección de datos para luego que, luego de realizada la intervención didáctica, hacer una evaluación del impacto de la innovación en la enseñanza a estudiantes de los primeros años de la carrera Ingeniería en Informática de la Facultad de Tecnología y Ciencias Aplicadas de la U.N.Ca.

## **Resultados y Objetivos**

### **Objetivo General**

Promover el desarrollo de habilidades cognitivas básicas y específicas de pensamiento computacional y de la programación a alumnos de las cátedras Fundamentos de Informática (1° año) y Programación I (2° año) de la carrera Ingeniería en Informática de la Facultad de Tecnología y Ciencias Aplicadas (FTyCa) de la Universidad Nacional de Catamarca (UNCa).

### **Objetivos Específicos**

- Facilitar el abordaje de conceptos relacionados con la Programación.
- Impulsar el desarrollo de habilidades del Pensamiento Computacional.
- Evaluar el nivel mejora en lo cognitivo, en lo actitudinal y motivacional de los alumnos en base a la intervención didáctica propuesta.

### **Resultados esperados**

Mediante el presente trabajo de investigación se busca la formación y consolidación de un equipo de docentes y alumnos de la Facultad de Tecnología y Ciencias Aplicadas que se especialice en la enseñanza de los fundamentos de la programación a través de una didáctica por indagación utilizando plataformas educativas de programación por bloques. A partir de la experiencia a desarrollar se pretende continuar en años sucesivos con este tipo de actividades con nuevos grupos de estudiantes en las cátedras mencionadas.

Se espera observar que a través de la pauta pedagógica y recursos didácticos utilizados, los alumnos logren aprendizajes significativos de los conceptos elementales de programación fortaleciendo las capacidades relacionadas a las 3 dimensiones del Pensamiento Computacional (conceptos, prácticas y perspectivas computacionales) de modo pueda incorporarlas de manera natural y hacer uso de estas capacidades en todos los aspectos de su formación profesional.

### **Formación de Recursos Humanos**

Los objetivos y actividades del presente proyecto, llevarán a que se realice capacitación permanente de todos los integrantes del Proyecto de

Investigación en temáticas de Pensamiento Computacional y Didáctica de la Programación, aportando a la formación complementaria en las carreras de posgrado que cursan algunos integrantes del proyecto:

- Maestría en Informática de la Universidad de Santo Tomás de Aquino (Tucumán) - Lic. Valeria Póliche,
- Maestría en Ingeniería de Software. - Lic. Oscar Quinteros bajo la co-dirección del Dr. Hernán Ahumada,
- Carrera de Especialización en Docencia Universitaria de Disciplinas Tecnológicas de la Facultad de Ciencias Agrarias de la Universidad Nacional de Catamarca - Lic. Daniel Rivas.
- Incorporación de cuatro docentes no categorizados a los efectos que se inicien en Proyectos de Investigación acreditados: Lic. Oscar Quinteros, Esp. Lic. Nelson Contreras, Esp. Lic. Marta Miranda e Ing. Javier Favore.

Los integrantes de este Proyecto tienen previsto participar de eventos científicos con el doble propósito de capacitarse en temáticas específicas concernientes al proyecto y para dar a conocer los resultados de las investigaciones realizadas. Eventos Científicos Nacionales a participar: WICC (Workshop de Investigadores en Ciencias de la Computación), Congreso Argentino de Tecnología en Educación y Educación en Tecnología (TE&ET), Simposio Argentino de Educación en Informática (SAEI), Jornadas Argentinas de Didáctica de la Programación (JADIPRO).

## Referencias

ISTE and CSTA (2011). Computer Science Teachers Association and the International Society for Technology in Education. "Pensamiento Computacional, Caja de Herramientas". Eduteka. Disponible en: <http://www.eduteka.org/modulos/9/272/2062/1>.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada (pp. 1-25).

Dann, W., Cooper S. & Pausch, R. (2011). Learning to Program with Alice. Prentice Hall

Fundación Sadosky (2012). "Informe CC 2016. Una propuesta para refundar la enseñanza de la computación en las escuelas argentinas", Buenos Aires, <http://www.fundacionsadosky.org.ar/wp-content/uploads/2014/06/cc-2016.pdf>.

- Levis, D. (2007). Enseñar y Aprender con informática/ Enseñar y aprender informática. Medios Informáticos en la escuela Argentina. In R. Cabello & D. Levis (Eds.), Medios Informáticos en la Educación a principios del siglo XXI (pp. 21-50). Buenos Aires-Argentina: Prometeo.  
-Martínez López, P. E. (2013). Las bases conceptuales de la programación. Una nueva forma de aprender a programar. Creative Commons.

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 10mo Simposio de la Sociedad de la Información (SSI'12), dentro de las 41ras Jornadas Argentinas de Informática (JAIIO '12), 1-23. ISSN 1850-2830.

Muñoz, R., Barcelos, T. S., Villarroel, R., Barría, M., Becerra, C., Noel, R., & Frango Silveira, I. (2015). Uso de Scratch y Lego Mindstorms como apoyo a la docencia en Fundamentos de programación. In Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática (pp. 248-254). Universitat Oberta La Salle.

Sawady O'Connor, F.A. & Factorovich, P. (2013). Cuaderno Teórico de Scratch. Fundación Sadosky

Torp, L., & Sage, S. (1998). El Aprendizaje Basado en Problemas. (E. Litwin, Ed.). Buenos Aires: Amorrortu.

Wing, J.M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Zúñiga, M. E., Rosas, M. V., Fernández, J., & Guerrero, R. A. (2014). El desarrollo del pensamiento computacional para la resolución de problemas en la enseñanza inicial de la programación. In XVI Workshop de Investigadores en Ciencias de la Computación