

## ¿Cuánta lógica se necesita? Una aproximación del Pensamiento Computacional en los primeros años de carreras STEM

Javier Fornari<sup>1</sup>, Claudia Deco<sup>1,2</sup>, Luciana Burzacca<sup>1</sup>, Cristina Bender<sup>2</sup>

<sup>1</sup> Facultad de Tecnologías e Innovación para el Desarrollo,  
Universidad Nacional de Rafaela, Rafaela, Argentina.  
{javier.fornari, luciana.burzacca}@unraf.edu.ar

<sup>2</sup> Facultad de Ciencias Exactas, Ingeniería y Agrimensura,  
Universidad Nacional de Rosario, Rosario, Argentina.  
{deco, bender}@fceia.unr.edu.ar

### Resumen

En las carreras vinculadas a las áreas de ciencia, tecnología, ingeniería y matemáticas (STEM) se presenta el problema de la alta deserción en los primeros años. Por esto, las universidades deben buscar diferentes estrategias para que los alumnos puedan motivarse y tener mejores herramientas para continuar su programa de formación. En este trabajo se presentan experiencias desarrolladas en la enseñanza de la primera materia de programación a estudiantes de Ingeniería en Computación y Licenciaturas en Agroinformática y en Bioinformática de la Universidad de Rafaela. Estas experiencias apuntan a incorporar el desarrollo del Pensamiento Computacional en estos cursos, considerando que este proceso de pensamiento involucrado puede ayudar a los estudiantes a resolver mejor los problemas relacionados a sus carreras. En primer lugar, se evaluaron las destrezas del Pensamiento Computacional alcanzadas por los alumnos utilizando la didáctica convencional mediante el test de Román-González. Los resultados obtenidos fueron un puntapié para diseñar e implementar una actividad especial articulada con las cátedras de Física, Química y Matemáticas. Además, se propone el dictado de un Taller de Pensamiento Computacional con modificaciones en la didáctica de las materias de programación.

**Palabras clave:** Programación, Pensamiento Computacional, Carreras STEM.

## 1 Introducción

En estos últimos años quedan muchos puestos de trabajo sin cubrir en las áreas de ciencia, tecnología, ingeniería y matemáticas, usualmente denominadas STEM (Science, Technology, Engineering & Mathematics), por la falta de recursos humanos con la formación y los conocimientos necesarios. Las carreras afines a estas áreas no son de las más populares y existe una gran deserción en los primeros años [1].

El aprendizaje del Pensamiento Computacional beneficia no sólo a futuros informáticos, sino a toda la sociedad y por lo tanto, su enseñanza debería estar presente en todos los niveles educativos siendo conocimientos importantes a desarrollar en todo profesional, independientemente del área en la que vaya a desempeñarse [2]. Por esto, es de interés analizar, evaluar y mejorar los procesos de enseñanza y aprendizaje de las competencias involucradas en el Pensamiento Computacional, para que los futuros profesionales estén más preparados para asumir los retos que demanda la sociedad contemporánea. En particular, se trata de evaluar la didáctica aplicada en las asignaturas de programación de las carreras de Ingeniería en Computación, Licenciatura en Agroinformática y Licenciatura en Bioinformática de la Facultad de Tecnologías e Innovación para el Desarrollo, de la Universidad Nacional de Rafaela. Además, se puede evaluar también el grado de entendimiento por parte de nuestros alumnos, para proponer cambios donde se detecten problemas de aprehensión de conceptos. Estos cambios apuntan a desarrollar una didáctica para la enseñanza de la programación en los primeros años de la universidad así como desarrollar metodologías activas y lúdicas para la utilización y entrega de los materiales en experiencias educativas presenciales y en línea.

En este trabajo, en la Sección 2 se presentan algunos conceptos y fundamentos teóricos así como un breve estado del arte; en la Sección 3 se presenta la experiencia realizada y una discusión de los resultados obtenidos; finalmente se presentan conclusiones y futuros trabajos.

## 2 Conceptos Básicos

En 2006 Wing define el Pensamiento Computacional como el proceso de pensamiento envuelto en formular un problema y sus soluciones de manera que las soluciones sean representadas de una forma en que pueden ser llevadas a un agente de procesamiento de información [2]. Además, en [3] se plantea que el aprendizaje del Pensamiento Computacional beneficia no sólo a los futuros informáticos, sino a toda la sociedad y por lo tanto, su enseñanza debería estar presente en todos los niveles educativos siendo conocimientos importantes a desarrollar en todo profesional, independientemente del área en la que vaya a desempeñarse. Por esto, el estudio del Pensamiento Computacional ha tenido influencia en las investigaciones relacionadas al entendimiento y desarrollo de los procesos de enseñanza y aprendizaje, provocando que se aborde el tema desde el punto de vista de la formación en los distintos niveles educativos [4,5].

Según ISTE (International Society for Technology in Education) el Pensamiento Computacional es un proceso para la solución de problemas que incluye: la formulación de problemas de forma que se pueda usar una computadora y otras herramientas para ayudar a resolverlos; el análisis y la organización de los datos de forma lógica; la representación de los datos de forma abstracta como modelos y simulaciones; la automatización de la solución con pensamiento algorítmico; la identificación, análisis e implementación de posibles soluciones; y la generalización y transferencia de este proceso de solución a otros problemas.




Brennan y Resnik [6] proponen una definición de Pensamiento Computacional que involucra tres dimensiones: Conceptos computacionales comunes a la mayoría de los lenguajes de programación (secuencias, bucles, paralelismo, eventos, condicionales, operadores lógicos y matemáticos, datos, etc); Prácticas computacionales, dado que el uso sólo de conceptos no es suficiente sino que deben ser usados siguiendo un conjunto de estrategias o prácticas (experimentación e iteración, testeo y debugging, reutilización de proyectos anteriores, abstracción y modularización, etc.); y Perspectivas computacionales que involucran la importancia de conectarse con otros y producir colaborativamente ayudando a resolver problemas mediante el desarrollo de herramientas propias.

Román-González y colaboradores [7] proponen un test que permite evaluar los conocimientos que posee un individuo sobre el Pensamiento Computacional, teniendo en cuenta la capacidad de abstracción, de descomposición, de reconocimiento de patrones y la habilidad para comprender y corregir algoritmos. Aunque este test está orientado a los niveles primario y secundario de escolaridad, consideramos que es aplicable y válido para los primeros años de las carreras universitarias. El test consta de 28 preguntas, donde cada pregunta evalúa diferentes capacidades: abstracción, descomposición, reconocimiento de patrones y algoritmos. Algunas de las consignas tienen en cuenta algunas de estas capacidades, y otras evalúan todas las capacidades, lo cual se resume en la Tabla 1.

**Tabla 1:** Capacidades de Pensamiento Computacional evaluados en el test de Román-González

Capacidades evaluadas	Preguntas
Abstracción y Algoritmo	1, 2 y 3
Abstracción y Descomposición	7
Abstracción, Descomposición y Algoritmo	13 y 21
Abstracción, Descomposición, Reconocimiento de Patrones y Algoritmo	11, 12, 15, 22, 23, 25, 26, 27 y 28
Abstracción, Reconocimiento de Patrones y Algoritmo	14
Algoritmo	16, 19 y 24
Descomposición, Reconocimiento de Patrones y Algoritmo	4, 5, 6 y 10
Reconocimiento de Patrones y Algoritmo	8, 9, 17, 18 y 20

A modo de ejemplo, en la Figura 1 se presenta la consigna 26, que es una de las que el test de Román-González evalúa las cuatro capacidades.

<p>Si tenemos el siguiente conjunto de órdenes, al que llamamos 'my function', y que dibuja un triángulo de 50 píxeles de lado:</p>  <p>¿Qué le falta a la siguiente secuencia para que el artista dibuje el siguiente diseño? Cada uno de los lados de cada triángulo mide 50 píxeles.</p>  	<p>Opción A: 15</p> <p>Opción B: 5</p> <p>Opción C: 4</p> <p>Opción D: 3</p>
<p>Consigna 26</p>	<p>Respuestas posibles</p>

**Fig. 1:** Ejemplo de pregunta del cuestionario, pregunta 26 del test de Román-González.

Con respecto a las dimensiones propuestas por Brennan y Resnik [6], algunas son evaluadas por el test de Román-González [7]. Respecto a la dimensión Conceptos Computacionales, que considera conceptos usados durante la programación, el test evalúa los componentes secuencias, bucles, condicionales y operadores, pero no evalúa eventos, paralelismo y datos. Respecto a la dimensión Prácticas Computacionales, es decir prácticas de resolución de problemas usadas durante el proceso de programación, el test evalúa parcialmente la aplicación de testeo y debugging, la reutilización de proyectos anteriores y la abstracción y modularización, y no evalúa el uso de experimentación e iteración. Finalmente, la dimensión Perspectivas Computacionales, que corresponde a cambios que produce el aprender a programar (forma de expresarse, conexión con otros y cuestionamientos) no son evaluados mediante el test de Román-González.

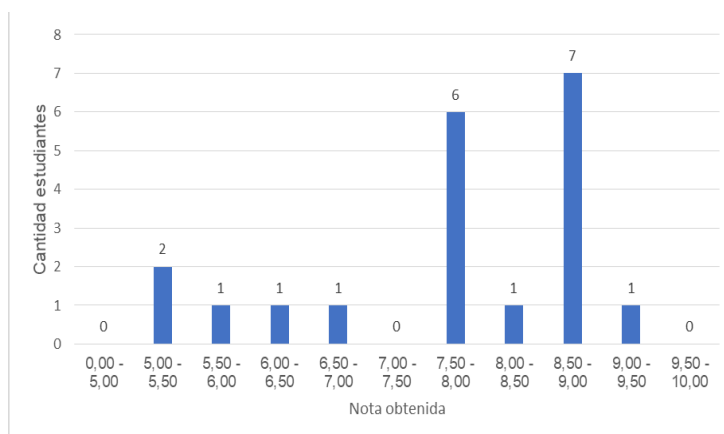
Aunque el test no evalúa todas las dimensiones propuestas por Brennan, consideramos que es adecuado para lo que se quiere analizar, ya que permite evaluar la adquisición de los conceptos y prácticas involucradas en el Pensamiento Computacional en los alumnos de nuestras carreras.

### 3 Experimentación y Discusión de Resultados

Luego de realizar una introducción general a la problemática de algoritmos y estructuras de datos, a mediados del segundo cuatrimestre de 2023 se realizaron una serie de actividades lúdicas donde en base a un problema los alumnos tenían que resolverlo aplicando técnicas de otras asignaturas o del conocimiento que traían desde la escuela secundaria, tratando de lograr una de las consignas más complejas para el alumno que es el proceso de abstracción en el pensamiento lógico. Luego, se aplicó el test de Román-González a los alumnos de primer año de las asignaturas de programación de las carreras de Ingeniería en Computación y Licenciaturas en Agroinformática y en Bioinformática de la Facultad de Tecnologías e Innovación para el Desarrollo, de la Universidad Nacional de Rafaela. El objetivo fue evaluar la obtención de conocimientos sobre el Pensamiento Computacional, sin asociarlos a ningún lenguaje de programación, logrado por estos alumnos con la didáctica convencional de asignaturas de programación antes de iniciar con los conceptos avanzados de programación orientada a objetos.

Para realizar el test, se utilizó el campus virtual de la Universidad de Rafaela. Cada consigna bien resuelta obtuvo un puntaje de 0,36 puntos, y cada consigna no respondida se calificó con 0 puntos. Se asignó un tiempo total de 90 minutos para responderlo. La realización del test se realizó en forma simultánea para todo el grupo de alumnos en un aula de informática de la Facultad de Tecnologías e Innovación para el Desarrollo.

En la Figura 2, se puede observar que 20 estudiantes, que representan el total de la matrícula de estudiantes del segundo cuatrimestre del primer año de dichas carreras, realizaron el cuestionario planteado en un promedio de 38 minutos cuando el tiempo total asignado fue de 90 minutos, con un resultado final obtenido de promedio 7,71. En esta figura se visualiza que el valor mínimo para la nota obtenida fue 5 con sólo 3 alumnos y el máximo valor de nota fue 9 con 8 alumnos. No hubo alumnos que obtuvieron una nota inferior a 5.

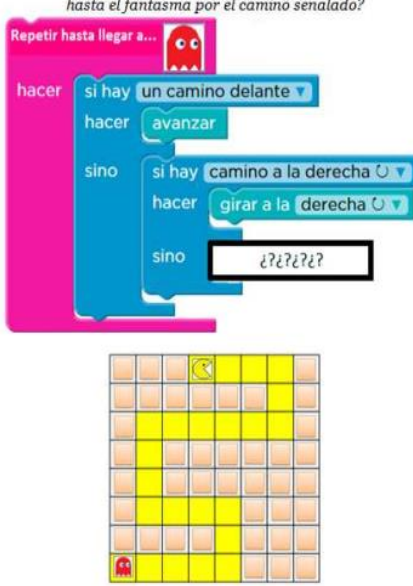

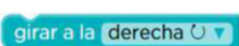



**Fig. 2:** Cantidad de estudiantes que alcanzaron cada rango de calificación

Algunas preguntas (12, 15, 22 y 23) de la categoría Abstracción, Descomposición, Reconocimiento de Patrones y Algoritmo y una (pregunta 19) de la categoría Algoritmo, resultaron con un puntaje inferior al promedio general denotando que no se alcanzaron los objetivos planteados para las mismas. En particular, la consigna 19 (Algoritmo) no fue respondida correctamente por ningún alumno, lo que llevó a revisar el cuestionario para determinar cuál fue la problemática o inconveniente por parte de los estudiantes. Si bien, en las primeras preguntas los alumnos fallaron en el grado de abstracción y descomposición del problema, se pudo observar que en el ámbito del planteo de la pregunta 19 el concepto de algoritmo no pudo ser desarrollado en forma adecuada. Aunque el alumno pudiera comprender el problema, abstraer y descomponer una posible solución no logró la lógica adecuada para el planteo de la secuencia lógica de la solución.

Respecto a las preguntas del test, para este trabajo, se eligieron en particular dos de ellas, las consignas 20 y 23, para realizar algunos comentarios.

La pregunta 20 (ver Figura 3) fue la única correctamente contestada por todo el grupo de alumnos a los que se les aplicó el test de Román-González.

<p>¿Qué bloque falta en la siguiente secuencia de órdenes para que 'Pac-Man' llegue hasta el fantasma por el camino señalado?</p> 	<p>Opción A: </p> <p>Opción B: </p> <p>Opción C: </p> <p>Opción D: <u>No falta ningún bloque</u></p>
<p>Consigna 20</p>	<p>Respuestas posibles</p>

**Fig. 3:** Corresponde a la Consigna 20, bien resuelta por todos los alumnos (100%)

La pregunta 23 (ver Figura 4) fue la que tuvo peor performance ya que sólo el 36% de los alumnos la resolvieron correctamente.

En las otras preguntas se obtuvieron valores entre estos dos (36% y 100%).



laboral de manera más eficiente. Esta situación se notó en el proceso de articulación de las diferentes carreras mencionadas, ya que si bien tienen orientaciones y perfiles diversos, la base de las mismas requiere de un fuerte apoyo y sustento de lógica y pensamiento divergente.

Para evaluar la didáctica convencional de las asignaturas de programación se realizó el test de Román-González a los alumnos antes de iniciar con los conceptos avanzados de programación orientada a objetos. A partir de los resultados obtenidos se puso en marcha el diseño y la implementación de una actividad especial articulada con las cátedras de Física, Química y Matemáticas. El objetivo fue que el Pensamiento Computacional fuera aplicado a problemas de ciencias básicas logrando así una mejora en la segunda instancia de los parciales de estas asignaturas.

A su vez, esta primera experiencia llevó a realizar modificaciones en la didáctica de las materias de programación. Por esto, se plantea el dictado de un Taller de Pensamiento Computacional en el TIEU (Taller de Ingreso a los Estudios Universitarios) en conjunto con el CFG (Ciclo de Formación General). En este taller al inicio del dictado del año lectivo 2024, se aplicó el test a los alumnos en el primer día de clase y al finalizar el segundo cuatrimestre se solicitará a los mismos alumnos que respondan nuevamente el mismo test. Esto permitirá evaluar si hubo o no mejoras con respecto a los resultados obtenidos previamente con el fin de implementar un sistema de evaluación y mejora continua en carreras STEM.

## Referencias

1. Dapozo, G., Greiner, C., Pedrozo Petrazzini, G., Chiapello, J.: Vocaciones TIC. ¿Qué tienen en común los alumnos del nivel medio interesados por carreras de informática? En IX Congreso de Tecnología en Educación & Educación en Tecnología, pp 128-139 (2014).
2. Wing, J.: Computational Thinking. View Point. Communication of ACM. Vol. 49, No. 3. Pág. 35. (2006)
3. Wing, J.M.: Computational thinking benefits society. 40th anniversary blog of social issues in computing, 2014, 26. (2014)
4. García, M., Deco, C., Bender, C., Collazos, C.: Invited paper: Robotics Based Strategies to Support Computational Thinking: The Case of the Pascual Bravo Industrial Technical Institute. Journal of Computer Science and Technology. Vol. 17 No. 1, 59–67 (2017)
5. Rodríguez del Rey, Y., Cawanga Cambinda, I., Deco, C., Bender, C., Avello-Martínez, R., Villalba-Condori, K.: Developing Computational Thinking with a Module of Solved Problems. In Computer Applications in Engineering Education. Marzo (2020).
6. Brennan, K., Resnick, M.: 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. Canadá. (2012).
7. Román-González, M., Perez-González, J.C., Jiménez-Fernández, C.: Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. Computers in Human Behavior, 72, 678-691 (2017).