

El Desarrollo del Pensamiento Computacional para la Resolución de Problemas en la Enseñanza Inicial de la Programación

M. E. Zúñiga, M. V. Rosas, J. M. Fernández, R. A. Guerrero

Departamento de Informática -F.C.F.M. y N.

LIDIC- Universidad Nacional de San Luis

Ejército de los Andes 950

Tel: 0266 4520300, San Luis, Argentina

{mezuniga, mvrosas, jmfer, rag}@unsl.edu.ar

Resumen

El abordaje de la enseñanza y aprendizaje de la programación de algoritmos para resolver problemas triviales usando una computadora implica una complejidad que precisa ser tratada desde el inicio de la formación de grado en las carreras afines a la computación. La verdadera dificultad no reside en expresar la solución del problema en términos de instrucciones elementales, sino en el desarrollo del pensamiento computacional involucrado para la resolución del problema propiamente dicha.

El objetivo de esta investigación es promover nuevas propuestas didácticas que permitan afrontar dicha complejidad considerando, además, la problemática recurrente en relación con el desempeño académico de los alumnos ingresantes a la UNSL en lo que respecta al aprendizaje de la programación en carreras que la incluyan como asignatura en el bloque de las ciencias básicas de sus correspondientes planes de estudio.

Se prevé que la investigación propuesta habilitará a encontrar nuevas dimensiones de análisis acerca de las prácticas docentes que aporten

conocimiento significativo al campo de la didáctica de la programación.

Palabras Claves: Programación, Resolución de Problemas, TICs, Pensamiento Computacional.

Contexto

La investigación presentada en este trabajo forma parte de un conjunto de líneas de abordaje correspondientes al proyecto '*Estrategias para la mejora de la enseñanza de la Programación a alumnos ingresantes de las carreras de Ciencias e Ingenierías*' en el ámbito de la Facultad de Ciencias Físico Matemáticas y Naturales de la Univ. Nacional de San Luis.

La propuesta surge a partir de uno de los principios básicos que guían a la educación pública de contribuir a la mejora del rendimiento académico de los alumnos, promoviendo un mayor nivel de retención de los ingresantes, impartiendo educación de calidad.

1. Introducción

En computación el acto de programar implica: a partir de un problema expresado en lenguaje natural, especificar sus límites, diseñar o modelar una solución, y convertir dicha solución en un programa de computadora correcto, expresado en algún lenguaje de

programación de propósito particular [1].

“...la solución de problemas mediante la programación de computadoras demanda de los estudiantes encontrar diversas maneras de abordar problemas y de plantear soluciones. Además, desarrollar habilidades para visualizar rutas de razonamiento divergentes, anticipar errores y evaluar rápidamente los diferentes escenarios mentales.” [2]

Los estudiantes deben desarrollar una gran variedad de habilidades que van más allá de la simple codificación de un programa, pues *implica aprender a entender un problema* (abstraer, modelar, analizar), *plantear soluciones efectivas* (reflexionar sobre una abstracción, definir estrategias, seguir un proceso, aplicar una metodología, descomponer en problemas más simples), *manejar lenguajes para expresar una solución* (codificar, entender y respetar una sintaxis), *utilizar herramientas que entiendan esos lenguajes* (programar, compilar, ejecutar, depurar), *probar que la solución sea válida* (entender el concepto de corrección y de prueba), *justificar las decisiones tomadas* (medir, argumentar), entre otras. En este contexto, la resolución de problemas computacionales supone el desarrollo de habilidades genéricas que forman parte del llamado **pensamiento computacional** [3, 4].

La complejidad, entonces, no está en aprender la rigurosidad de un lenguaje de programación y expresar un algoritmo con ese lenguaje, sino en poder definir un algoritmo que resuelva el problema inicial. Es decir, el desafío está en desarrollar el pensamiento computacional para resolver problemas utilizando una computadora.

La programación permite materializar la idea de abstracción, uno de los procesos claves del pensamiento computacional, e incluso dentro de esta actividad se demuestra qué tan útil es dominar esta idea. El pensamiento computacional, entonces, también se hace concreto cuando aprendemos a programar.

“Enseñar el pensamiento computacional no solamente podría inspirar a las generaciones futuras a entrar en el campo de las Ciencias de la Computación dada la aventura intelectu-

al, sino que beneficiaría a la gente en todos los campos” [3].

En este punto es importante contextualizar a estas nuevas generaciones, en particular la población compuesta por los alumnos ingresantes al nivel universitario. Cada año miles de estudiantes, provenientes de distintos lugares del país y de diferentes instituciones educativas, ingresan a nuestra universidad. Como docentes de asignaturas del primer año en carreras universitarias debemos contribuir a la adaptación de los estudiantes a la vida universitaria. Ésta supone la apropiación de nuevas responsabilidades, normativas y hábitos, todo un desafío para muchos ingresantes, constituyéndose así en un factor que incide en su rendimiento académico.

A partir de nuestra experiencia docente y de los resultados académicos obtenidos en los últimos períodos, fue posible identificar algunos aspectos críticos de los alumnos ingresantes que pudieron influir de forma negativa en su desempeño:

- Las ‘habilidades académicas’ que caracterizan a la mayoría de los alumnos ingresantes y que se relacionan principalmente con la falta de hábito de estudio, problemas en la comprensión de textos y dificultad en la resolución de problemas.
- La falta de motivación o poco interés de los alumnos. Potenciado por el hecho de que están acostumbrados a trabajar y contar con diferentes recursos tecnológicos que, si bien son utilizados, no son considerados el eje principal.
- Bajo umbral de tolerancia al fracaso.

Éstos y otros factores, desde la perspectiva de los alumnos ingresantes, quedan desfavorablemente expuestos en el alto porcentaje de deserción y el bajo rendimiento obtenido.

Sin embargo, es posible considerar a estas nuevas generaciones a partir del concepto de *Nativos Digitales*, acuñado por Marc Prensky, el cual manifiesta una serie de características que rigen su comportamiento, habilidades y pensamientos [5].

Diferentes autores que continúan el desarrollo de este concepto consideran que este colectivo de alumnos está más predispuesto a incluir las tecnologías en sus actividades de estudio y aprendizaje, que lo que las instituciones educativas les pueden ofrecer; resultando en una relación que podría estar interviniendo de forma negativa en el rendimiento académico de los mismos [6, 7].

“La forma en que enseñamos y aprendemos hoy en día en las aulas [...] no difiere mucho de aquella que se utilizaba en el pasado. [...] Sin embargo, los alumnos que asisten a esos cursos forman parte de nuevas generaciones acostumbradas a manejarse con las TICs de una manera cuasi innata y a aprovecharlas para modificar la forma en la cual se relacionan con sus pares, acceden a la información y al conocimiento, trabajan y, en síntesis, participan de la vida en sociedad.” [8].

Es posible, entonces, resignificar la actividad docente a la luz de estas nuevas concepciones teóricas promoviendo prácticas innovadoras que incluyan las TICs en los procesos de enseñanza-aprendizaje de la programación con el objetivo de motivar a los alumnos y así incidir de manera positiva en su rendimiento académico [9].

Las TICs son herramientas pedagógicas que pueden contribuir a la consecución de los fines educativos y a desarrollar la capacidad creativa de los sujetos. Sin embargo, para garantizar que la integración de estas tecnologías en el proceso de enseñanza-aprendizaje no se convierta en una trampa antipedagógica, es necesario analizar exhaustivamente cada uno de los elementos que inciden en el mismo [10, 11].

“La pregunta no es si se ha de usar tecnología, ni siquiera qué tecnologías usar, sino por qué y cómo usarlas.” [12].

2. Líneas de Investigación y Desarrollo

En función a lo anteriormente expresado, se propone abordar la problemática a partir de dos grandes ejes principales:

1. *La Resolución de problemas:* aplicando nuevos modelos de enseñanza-aprendizaje, técnicas como la programación por pares y el trabajo colaborativo, entre otros, para favorecer el desarrollo del pensamiento computacional con el propósito de abordar los problemas y plantear las soluciones factibles.
2. *El uso de las TICs como soporte en el proceso de aprendizaje de la Programación:* la utilización de distintos recursos tecnológicos que permitan y faciliten a los alumnos la práctica de la programación como complemento y soporte a la resolución de problemas; como por ejemplo, la incorporación de un módulo de programación en entornos virtuales de aprendizaje para contar con un espacio más de desarrollo e intercambio, utilizando su potencialidad para mejorar la eficiencia de todo el proceso educativo en lo que respecta a solución de problemas mediante la programación.

3. Resultados obtenidos / esperados

En los estudios preliminares y el análisis de la realidad áulica que se viene presentando en las asignaturas que se imparten, se ha detectado que el rendimiento académico de los alumnos es cada vez menos satisfactorio. En este punto, es importante llevar a cabo acciones concretas con la intención de atenuar esta situación. Es así que se proponen los siguientes objetivos a corto, mediano y largo plazo:

1. Indagar los factores académicos que influyen en el desempeño de los alumnos ingresantes a la UNSL en lo que respecta al aprendizaje de la programación en carreras de Ciencias e Ingeniería.
2. Promover nuevas estrategias didácticas para abordar la resolución de problemas en la enseñanza de la programación.

3. Contribuir con la producción de nuevos recursos educativos para el aprendizaje de la programación potenciando el desarrollo de las habilidades y competencias que forman parte del pensamiento computacional.
4. Desarrollo y/o adecuación de un entorno integrado de desarrollo flexible y apropiado para la enseñanza inicial de la programación basado en la resolución de problemas y que incorpore soporte al aprendizaje colaborativo

4. Formación de Recursos Humanos

La presente propuesta surge como consecuencia de la inquietud de un grupo de docentes responsables del dictado del primer curso de programación que reciben los alumnos ingresantes en las carreras de Ciencias e Ingenierías dictadas por el Departamento de Informática de la Facultad de Ciencias Físico-Matemáticas y Naturales, de la Universidad Nacional de San Luis. Las acciones del grupo de trabajo tienen como objetivo final el consolidarse como equipo de investigación dedicado a profundizar en el abordaje de la enseñanza y aprendizaje de la programación de algoritmos.

En lo que respecta a la contribución en la formación de RRHH se ha colaborado a partir de:

- La propuesta de temáticas para ser abordadas como trabajos finales de carreras de grado y posgrado relacionadas con los ejes centrales del proyecto.
- La organización de cursos de capacitación y actualización destinados a docentes vinculados al bloque de ciencias básicas en las carreras de Ciencias e Ingeniería.
- La incorporación de pasantes y becarios alumnos.

Referencias

- [1] Gries D. *The Science of Programming*. Monographs in Computer Science Series. Springer-Verlag, 1981.
- [2] Stager G. En pro de los computadores, January 2004.
- [3] Wing Jeannette M. Computational thinking. *Commun. ACM*, 49(3):33–35, March 2006.
- [4] Bryant Robert, Chinn Donald, Hauser George, Folsom Michele, and Wallace Scott. Computational thinking: What is it, how is it relevant, who's doing what with it? *J. Comput. Sci. Coll.*, 25(1):170–171, October 2009.
- [5] Prensky Marc. Digital natives, digital immigrants. *On the Horizon*, 9(5), 2001. October 2001.
- [6] Adell Jordi. *Boletín Electrónico de la Cátedra UNESCO de Gestión y Política Universitaria*, chapter Los estudiantes universitarios en la era digital: la visión del profesor, pages 97–100. La Cuestión Universitaria, 2011.
- [7] Adell Jordi and Castañeda Linda. *Tendencias emergentes en educación con TICs*, chapter Tecnologías emergentes, pedagogías emergentes?, pages 13–32. Asociación Espiral, Educación y Tecnología, 2012.
- [8] Cukierman U., Rozenhauz J., and Santángelo H. *Tecnología Educativa: Recursos, modelos y metodologías*. Pearson, 2009.
- [9] Astrachan Owen. A new way of thinking about computational thinking. *J. Comput. Sci. Coll.*, 25(1):6–6, October 2009.
- [10] Cukierman U. Las TICs en la educación de ingeniería de las nuevas generaciones. In *Congreso de Información y Comunicación para la Sociedad del Conocimiento (CNIT 2009)*, 2009.
- [11] Walden James, Doyle Maureen, Garns Rudy, and Hart Zachary. An informatics perspective on computational thinking. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '13*, pages 4–9, New York, NY, USA, 2013. ACM.
- [12] Buckingham D. *Más Alla De La Tecnología: Aprendizaje Infantil en la Era Digital*. Ed. Manatíal, 2008.