

Estrategia metodológica de la enseñanza de la programación para la permanencia de los alumnos de primer año de Ingeniería Electrónica

Norma Arellano¹, Jacqueline Fernandez¹, María Verónica Rosas¹, Mariela E. Zuñiga¹

¹ Universidad Nacional de San Luis, San Luis, Argentina
{nmare, jmfer, mvrosas, mezuniga}@unsl.edu.ar

Resumen

La programación es una disciplina de las Ciencias de la Computación con muchas aplicaciones, incluyendo problemas no triviales cuya resolución constituye un desafío intelectual. La verdadera dificultad no reside en expresar la solución del problema en términos de instrucciones elementales de un lenguaje de programación específico, sino en la resolución del problema propiamente dicha. El proceso de encontrar una solución adecuada a un problema provoca en el alumno un conflicto cognitivo pues no dispone de un sistema de estrategias que le permitan responder de manera satisfactoria. Del análisis de la realidad áulica que se viene presentando con los alumnos ingresantes a la carrera de Ingeniería, se ha detectado que el rendimiento académico de los estudiantes es cada vez menos satisfactorio sumado a la complejidad propia de los contenidos de la asignatura.

En este trabajo se presenta la experiencia realizada durante el dictado del primer curso de programación que reciben los alumnos de primer año en la carrera de Ingeniería Electrónica de la Facultad de Ciencias Físico-Matemáticas y Naturales, de la Universidad Nacional de San Luis.

Palabras clave: Pensamiento Computacional, TICs, Enseñanza, Programación.

Abstract

Programming is a computer science discipline that has many applications in the real world including the resolution of nontrivial problems that involve great intellectual challenge. The main difficulty lies not in expressing the solution in terms of elementary instructions, but in solving the problem. To find an appropriate solution for a problem causes in the student

a cognitive conflict due of they do not dominate a system of strategies in order to respond satisfactorily.

The analysis of what happen in the classroom today with the first level students to the Engineering, exhibit that the academic performance of students is becoming less satisfactory added to the complexity of the subject content.

The experience made during the dictation of the first programming course students receive in the first year of electronic engineering, Faculty of Physics and Mathematics, and from the National University of San Luis, is presented in this paper.

Keywords: Computational Thinking, ICT, Teaching, Programming.

1. Introducción

Enseñar, o mejor aún, aprender a programar computadoras incluye un proceso en el que la persona pone en juego una gran variedad de habilidades y capacidades. En el proceso de aprender a programar, es dable esperar que los alumnos adquieran habilidades y desarrollen capacidades fundamentales en la resolución de problemas. No se está simplemente aprendiendo a programar sino que al mismo tiempo se está programando para aprender; pues, además de comprender ideas computacionales conceptuales y específicas, simultáneamente se está aprendiendo estrategias para solucionar problemas. Habilidades que no solo son útiles para los expertos de la computación sino para muchas de las actividades cotidianas de la vida de las personas, sin distinción de edad, origen, intereses u ocupación [6] [8].

Existen diferentes metodologías de resolución como el método heurístico para resolver problemas definido por el matemático Polya en 1957, el cual consiste fundamentalmente en entender el problema, trazar un plan, ejecutar el plan y revisarlo. Siendo el análisis el elemento fundamental del proceso de resolución. Esta

estrategia permite transformar el problema en una expresión más sencilla que se sepa resolver. Esta metodología puede pensarse como el instrumento heurístico que permite descubrir, interrelacionar y desarrollar el pensamiento crítico y reflexivo, la creatividad y capacidad de inventiva con el pensamiento computacional necesario para implementar la transformación [1] [15].

En la era digital el pensamiento computacional es una habilidad que se encuentra al alcance de todos. Es un proceso de solución de problemas que se caracteriza por organizar los datos de manera lógica para su análisis, representar datos mediante abstracciones, formular soluciones a problemas computacionales y automatizarlas algorítmicamente. El pensamiento computacional implica un pensamiento recursivo y un procesamiento paralelo [13] [15].

Los avances tecnológicos han ampliado significativamente la capacidad de resolución de los problemas y, por lo tanto, los estudiantes no sólo necesitan aprender sino practicar nuevas habilidades como la del pensamiento computacional que les permitirá aprovechar el potencial generado por los rápidos avances en las TICs [4] [12].

La universidad es consciente de su rol en la sociedad del conocimiento, donde se ha pasado de un paradigma concentrado en la enseñanza a un paradigma concentrado en el aprendizaje apoyado en la construcción colaborativa del conocimiento. Es pertinente entonces renovar los grados de innovación con el objetivo de favorecer la reducción del nivel de deserción, mejorar el desempeño académico de todos los estudiantes y establecer bases robustas que le faciliten su permanencia y egreso del sistema universitario [7].

2. Contexto e identificación de la problemática Sección

Cada año miles de estudiantes provenientes de distintos lugares del país y de diferentes instituciones educativas, ingresan a la UNSL. Es necesario que los docentes de asignaturas del primer año en carreras universitarias deban contribuir con la adaptación de los estudiantes a la vida universitaria. Proceso que 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.

Esta experiencia se ha llevado a cabo con los alumnos de la asignatura Fundamentos de la Informática de la carrera Ingeniería Electrónica con Orientación en Señales digitales de la FCFMyN. Siendo el primer curso de programación que reciben los alumnos de esta carrera, su dictado es responsabilidad de docentes del Dpto. de Informática de dicha facultad. La asignatura se organiza semanalmente en seis horas de clases

teórico-prácticas debiéndose cumplir, según el plan de la carrera, un total de 105 horas. Inicialmente el crédito horario se completaba con clases de consultas que no eran aprovechadas por los alumnos.

Del análisis 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 con 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.

Así fue que, teniendo en cuenta la problemática detectada y considerando las características propias de los alumnos ingresantes en su calidad de “nativos digitales” y a su manejo cuasi innato de la tecnología se estudian diferentes propuestas metodológicas que contribuyan a mejorar los procesos educativos involucrados, como la incorporación de prácticas de laboratorios complementarias a las clases de consulta y con la intención de motivar a los alumnos [2] [10].

3. Justificación de las herramientas utilizadas

Las distintas herramientas de software que se utilizaron para desarrollar las prácticas de laboratorio se fueron incorporando en base a su complejidad de uso y en correspondencia con las prácticas áulicas dando soporte a las mismas.

Se incorpora el uso de la herramienta **TIMBA** (*Terribly Imbecile Machine for Boring Algorithms*) que fuera ideado con fines educativos por un grupo de docentes de la UNSL, dirigido por el Ing. Hugo Ryckeboer. Dicha herramienta fue desarrollada en respuesta a la necesidad de contar con un pseudo-lenguaje simple que permita introducir al alumno en la noción de algoritmo y los conceptos básicos para la construcción de los mismos basado en el paradigma de la programación estructurada. TIMBA es un lenguaje que permite la definición de algoritmos utilizando las tres estructuras de control básicas (secuencial, condicional e iterativa) y que consisten de una secuencia de órdenes a un ejecutor, denominado UCP, capaz de comprender un conjunto reducido de acciones primitivas y de manipular pilas de cartas [14].

La aplicación **Dia1** (*Diagram Editor*) es una herramienta muy potente y fácil de aprender que permite la creación de forma sencilla de numerosos tipos de diagramas: UML, de flujo, de red, y cronogramas. Dia fue empleada como complemento y facilitador del proceso de diseño y definición del algoritmo final a través de la realización de diagramas de flujo simples.

Finalmente, se incorpora un software multiplataforma de distribución libre y gratuito denominado **PSeInt2** (*Pseudo code Interpreter*) para iniciar a los alumnos en la construcción de programas o algoritmos computacionales similares a los lenguajes formales de programación pero sin tener que lidiar con las particularidades estrictas de sintaxis. Con la incorporación de esta herramienta se continúa con la profundización del uso de las estructuras de control agregando complejidad a los conceptos ya aprendidos al utilizar TIMBA. Se introducen los conceptos de variables, expresiones, manejo de estructuras de datos y la implementación de la Modularización.

La simpleza de sintaxis de esta herramienta le facilitan al alumno la tarea de escribir algoritmos en un pseudo-lenguaje similar al lenguaje natural haciendo hincapié en la apropiación significativa de conceptos básicos en la programación. Presenta un conjunto de ayudas y asistencias, además de brindar algunas herramientas adicionales que le ayuden al alumno a encontrar errores y comprender la lógica de los algoritmos como un paso previo a programar en un entorno integrado de programación más complejo y lenguajes de programación formales como Pascal, C o Java.

Además, se incorporó la utilización de la plataforma virtual Moodle “Aulas Virtuales”, (un entorno virtual de aprendizaje disponible desde el Centro de Informática Educativa del Departamento de Informática) que permitió contar con un espacio más de comunicación e intercambio con los alumnos, generando un laboratorio virtual que logra un ambiente común e integrado de trabajo.

4. Experiencia

Teniendo en cuenta que la experiencia se ha llevado con alumnos de la carrera de Ingeniería Electrónica con Orientación en Señales digitales en cuyo plan de estudios, la asignatura Fundamentos de la Informática corresponde al segundo cuatrimestre del primer año de la carrera pero se repite su dictado en el primer cuatrimestre, se decide implementar la nueva propuesta de trabajo en dos etapas durante el año 2013.

En la primera etapa, desarrollada durante el primer cuatrimestre, se propuso la incorporación de un horario para laboratorio de programación de modalidad opcional, donde se utilizaron aplicaciones que permiten al alumno verificar y depurar los programas resueltos en las clases prácticas. Para iniciar a los alumnos en el uso de una herramienta de programación y comenzar a familiarizarse con los conceptos básicos de

programación estructurada, tales como algoritmos, estructuras secuenciales, condicionales e iterativa se utilizó el software **TIMBA**. La realización de diagramas de flujo con la aplicación **Dia** permitió a los alumnos complementar de forma gráfica la resolución de los algoritmos. Finalmente, para afianzar los conceptos fundamentales de la algoritmia computacional y acercar al alumno al uso de un entorno de desarrollo integrado de programación se utilizó el software **PSeInt** que permite modificar, ejecutar y depurar los algoritmos.

En la segunda etapa, se incorporó un horario de laboratorio para la práctica de la programación, a diferencia de la primera etapa con modalidad obligatoria. Se continuó con el uso de las herramientas Timba y Pseint, incorporando su editor de diagramas de flujo. Además, con la intención de generar un espacio común de trabajo y promover un espíritu colaborativo en el proceso de enseñanza aprendizaje se incorporó la utilización de la plataforma virtual Moodle “Aulas Virtuales”. Se usaron diferentes recursos que provee dicha plataforma principalmente aquellos que favorecen la comunicación, el seguimiento y la retroalimentación personalizada.

5. Analisis de los Resultados

5.1. Desde la perspectiva docente

La cátedra considera como positiva la experiencia en diferentes sentidos:

- Los medios utilizados permitieron una nueva forma de comunicación e intercambio, sustentado en el hecho que los alumnos, en su mayoría, no sólo cuentan con una serie de recursos tecnológicos para la comunicación sino que, además la utilización de los mismos es más frecuente que la de los medios tradicionalmente propuestos por la universidad.
- Al redefinir la propuesta de enseñanza, a partir de la disponibilidad tecnológica y en busca de mejorar la comunicación con los alumnos se favoreció el logro de una participación más activa y predisposta por parte de los mismos, si bien, cabe destacar que este incremento en la participación se produjo en forma paulatina y con diferentes grados de entusiasmo.
- Con la incorporación del aula virtual y de sus herramientas se logró un registro digital de: el progreso individual de cada alumno, la retroalimentación grupal e individual sobre cada actividad evaluada y la especificación de los criterios de evaluación en las diferentes etapas, todo esto influyendo de manera positiva en el ejercicio de la práctica docente.
- La utilización del aula virtual permitió proyectar una nueva estrategia de trabajo que modificó el estilo habitual del equipo docente, logrando una comunicación más activa, integrada y participativa.

5.2. Desde la perspectiva del alumno

Luego de haber finalizado las instancias de evaluación, se elaboró una encuesta individual para los alumnos a modo de cierre de la materia, con el propósito de conocer sus opiniones con respecto a la nueva estrategia planteada en lo que refiere a la utilización de la plataforma virtual.

La encuesta fue contestada de manera anónima y voluntaria. Entre las consideraciones aportadas por los alumnos destacamos las siguientes:

- Los alumnos manifiestan estar familiarizados con el uso de las “Aulas Virtuales” ya que las mismas se utilizan en otras asignaturas de la misma facultad, esto se ve referenciado en el gráfico de la Figura 1.

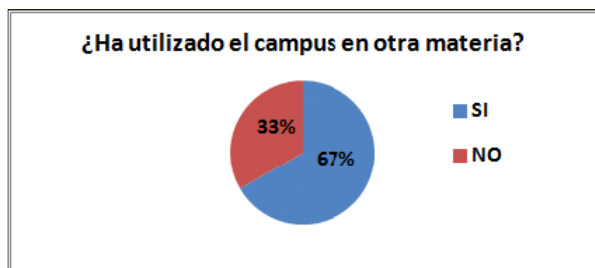


Figura 1

- En relación a la frecuencia en el acceso a la plataforma, como se observa en el gráfico de la Figura 2 el porcentaje mayor se encuentra en la frecuencia de acceso referenciada como “3 a 6 veces en la semana”, con lo que se evidencia que la mayoría de los alumnos realizaron un ingreso periódico al aula virtual.

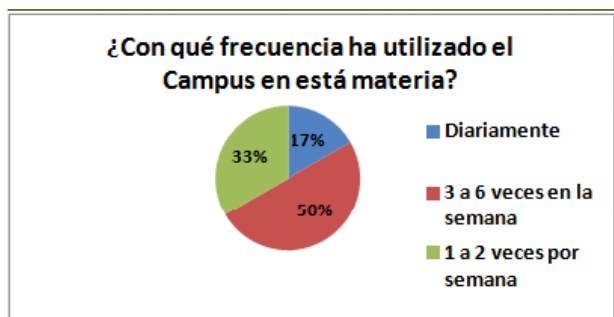


Figura 2

- El gráfico de la Figura 3 muestra que en la mayoría de los accesos que los alumnos han realizado al campus son con el propósito del envío de los ejercicios previamente establecidos en las clases prácticas para su posterior corrección y retroalimentación, para las demás tareas los porcentajes obtenidos son similares, exceptuando a la tarea de participación en los foros.

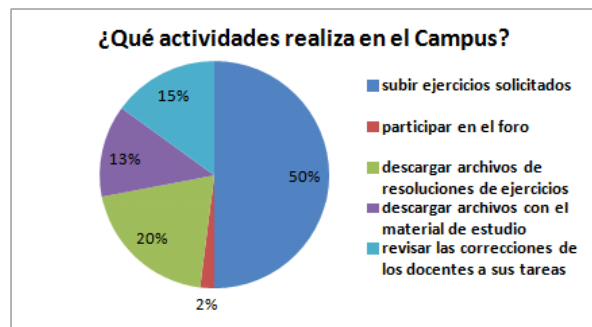


Figura 3

- Los alumnos reconocen como beneficioso la incorporación en la materia del uso del aula virtual, lo cual se observa al seleccionar principalmente los valores “Muy Bueno” y “Bueno” como se muestra en el gráfico de la Figura 4.

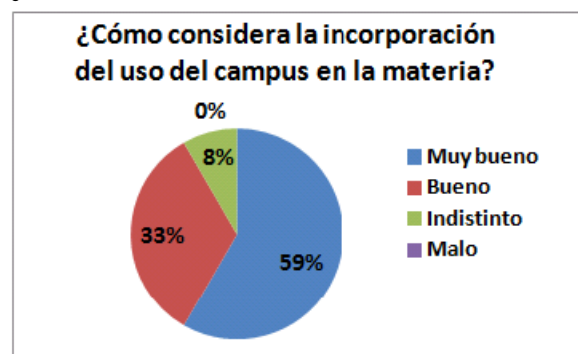


Figura 4

- La intención de utilizar el campus como una continuidad de la práctica áulica que beneficie al proceso de aprendizaje fue reconocido por la mayoría de los alumnos como tal, según se muestra en el gráfico de la Figura 5.

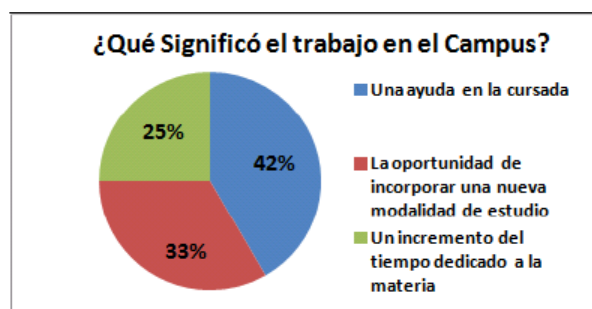


Figura 5

En una reflexión personal solicitada a cada alumno, en la misma encuesta, pudieron ampliar su opinión acerca de la asignatura y a la vez destacar otros beneficios aportados por el uso del campus como son, el contar con todo el material de estudio en un mismo lugar, la facilidad de comunicación a través de foros y chat con docentes y compañeros y, una mejor organización del

tiempo dedicado a la asignatura. Un porcentaje menor de alumnos planteó la dificultad en el uso adecuado del Campus debido a problemas de conexión a Internet.

Conclusiones

Si bien, la intención de llevar a cabo esta experiencia fue abordar una problemática identificada en una carrera específica y en un contexto determinado, es importante enfatizar que esta realidad se reitera en carreras similares en diferentes universidades. Por lo tanto, se considera significativo innovar en las propuestas de enseñanza en busca de una solución factible que permita intervenir positivamente.

De la experiencia llevada a cabo por este equipo de trabajo se pueden destacar:

La integración de recursos tecnológicos en el proceso de enseñanza de la programación tuvo un impacto positivo como estrategia para afianzar la comunicación entre docentes y alumnos, y como facilitador del desarrollo de las capacidades y competencias necesarias en alumnos que se inician en la práctica de programar.

La formación del pensamiento computacional en la resolución de problemas representa una estrategia potencialmente significativa para fomentar en los alumnos distintas habilidades que promuevan el desarrollo del pensamiento crítico y potenciar su creatividad.

El desafío a partir de esta nueva experiencia podría plantearse en dos grandes ejes:

- Creación y promoción de acciones que favorezcan el trabajo colaborativo para propiciar procesos de enseñanza-aprendizaje más significativos. El proceso de implementar esta nueva estrategia requerirá principalmente la adaptación de la tecnología a esta nueva propuesta y una capacitación continua de los docentes en el rol de tutor.
- Integración de herramientas de desarrollo al campus, generando un laboratorio virtual en la plataforma y logrando un ambiente común e integrado de trabajo. En este sentido, se está trabajando en la incorporación del módulo de programación VPL (Virtual Programming Lab), un software de código abierto que permite la gestión de prácticas de programación en el entorno Moodle posibilitando, de esta manera, la integración de herramientas de programación al campus.

Referencias

- [1] Polya, George How to solve it; Princeton University Press, (1957) Segunda Edición.
- [2] Adell, J. Los estudiantes universitarios en la era digital: la visión del profesor. La Cuestión Universitaria, (2011), pp 97-100.
- [3] Brookshear, J. Introducción a las Ciencias de la Computación, Wilmington Delaware (U.S.A.): Addison-Wesley Iberoamerican, S.A., (1995)
- [4] Cukierman, U. Las TICs en la Educación de Ingeniería de las Nuevas Generaciones. Información y Comunicación para la Sociedad del Conocimiento, (2009). Córdoba, Argentina.
- [5] Gries D. The Science of programming. Springer-Verlag. (1981).
- [6] Helminen, J., Ihantola, P., Karavirta, V., & Malmi, L. How Do Students Solve Parsons Programming Problems?. An Analysis of Interaction Traces. Proceedings of the Eighth Annual International Computing, (2012), pp. 119-126
- [7] Lovos, E., Gonzalez, A. et al. Estrategias de enseñanza colaborativa para un curso de Programación de primer año de la Lic. en Sistemas. CACIC XVIII (2012).
- [8] Mac Gaul, M., López, M., Del Olmo, P. Resolución de problemas computacionales: Analisis del proceso de aprendizaje. TE&ET, (2008).
- [9] Negroponte, N. Ser digital. (1995), Buenos Aires, Argentina: Atlántida.
- [10] Prensky, M. Digital natives, digital immigrants. On the Horizon, (2001).
- [11] Prudkin, A. (30 de Junio de 2010). Educ.ar. Obtenido de <http://portal.educ.ar/debates/sociedad/cultura-digital/manuel-castells-en-argentina-c.php>
- [12] Rozenhauz, J., Cukierman, U., & Santángelo, H. Tecnología Educativa: Recursos, modelos y metodologías. (2009). Buenos Aires: Pearson.
- [13] Stager, G. (13 de Enero de 2004). En pro de los computadores. Obtenido de Eduteka: <http://www.eduteka.org/ProComputadores.php>
- [14] Szpiniak, A., Rojo, G. Enseñanza de la programación. TE&ET: Revista Iberoamericana. (2006)

[15] Wing, J. M. Computational Thinking. Communications of the ACM, 49(3), (2006), pp 33-35..

Dirección de Contacto del Autor/es:

Norma M. Arellano
Ejército de los Andes 950
San Luis
Argentina
e-mail: nmare@unsl.edu.ar

Jacqueline M. Fernández
Ejército de los Andes 950
San Luis
Argentina
e-mail: jmfer@unsl.edu.ar

María V. Rosas
Ejército de los Andes 950
San Luis
Argentina
e-mail: mvrosas@unsl.edu.ar

Mariela E. Zuñiga
Ejército de los Andes 950
San Luis
Argentina
e-mail: mezuniga@unsl.edu.ar