

Una experiencia en la enseñanza de la programación para la permanencia de los alumnos de Ingeniería Electrónica

Norma Arellano, María Verónica Rosas, Mariela E. Zuñiga, Jacqueline Fernandez , Roberto Guerrero
Universidad Nacional de San Luis,
Ejército de los Andes 950
Tel: 02664 420823,
San Luis, Argentina
{nmare, mvrosas, mezuniga, jmfer, rag}@unsl.edu.ar

Resumen La programación es una disciplina de las Ciencias de la Computación con diversas y variadas aplicaciones. Aprender a programar problemas computacionales representa un profundo 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 dicho. El proceso de encontrar una solución adecuada al mismo provoca en el alumno un conflicto cognitivo por no dominar un sistema de estrategias que le permitan afrontar la situación de manera satisfactoria. Del análisis de la realidad en el aula que se viene presentando con los alumnos que ingresan a la carrera de Ingeniería, se ha detectado que el rendimiento académico de los estudiantes es cada vez menos satisfactorio agravado por la complejidad intrínseca de los contenidos de la asignatura. En este trabajo se presenta la experiencia realizada durante el dictado del primer curso de enseñanza de la 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, Argentina.

Palabras clave: Resolución de Problemas, Pensamiento Computacional, TICs, Enseñanza y Aprendizaje de la Programación, B-Learning

Abstract: Programming is a computer science discipline that has many applications in the real world. Learning to program computational problems represents an intellectual challenge. The main difficulty lies not in expressing the solution in terms of elementary instructions, but in solving the problem itself. To find an appropriate solution for a problem causes with 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. This paper presents the experience made during the first programming course the students receive in the first year of electronic

engineering, Faculty of Physics and Mathematics, and from the National University of San Luis, Argentina.

Keywords: Problem Solving, Computational Thinking, ICTs, Learning and Teaching Programming, B-Learning

1. Introducción

Aprender a programar computadoras involucra 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 incorporar conceptos computacionales, simultáneamente se está aprendiendo estrategias para solucionar problemas. Habilidades que no sólo son de utilidad para los expertos en computación sino, también, para muchas de las actividades cotidianas de la vida de las personas, sin distinción de edad, origen, intereses u ocupación [6, 8].

Existen diferentes modelos de resolución, Polya, Mason, Burton, Stacey, Bransford, Guzmán entre otros, que establecen cómo actuar frente a los problemas. Los modelos de Mason, Burton y Stacey, analizan los comportamientos de los sujetos reales involucrados en la resolución de problemas y sus características propias (reacciones, sensaciones, emociones, altibajos, retrocesos o inspiraciones) considerando positivos los atascos y errores que pudiesen surgir. Bransford y Guzmán proponen hábitos mentales útiles mientras que el método heurístico definido por el matemático Polya en 1957, describe el comportamiento de quien resuelve el problema al recorrer secuencialmente determinadas fases fundamentales pasando de una a otra sólo cuando se ha concluido con la anterior. El modelo de Polya define un marco conceptual que consiste de cuatro etapas fundamentales: entender el problema donde se identifican los datos dados y se definen las incógnitas, trazar un plan donde se establece la relación entre los datos y las incógnitas y se buscan patrones, ejecutar el plan de la posible solución donde se comprueba si son los pasos correctos y finalmente revisarlo, etapa de la visión retrospectiva donde es muy importante observar qué fue lo que se hizo, verificar el resultado y el razonamiento seguido. 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]. Las abstracciones son las herramientas mentales

de la computación. El poder de las herramientas mentales aumenta con el poder de las herramientas metálicas, la computadora. Computación, en consecuencia, es la automatización de las abstracciones [16].

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]. Es necesario conocer las potencialidades de estas herramientas para hacer un uso adecuado de las mismas.

En este contexto se encuentra el Blended learning o aprendizaje que combina la enseñanza presencial con la enseñanza no presencial. Modelo híbrido donde se deja de lado la “enseñanza asistida por computadora” para incorporar el “aprendizaje basado en la computadora” y es incorporado como complemento de la modalidad presencial para definir estrategias que permitan mejorar los resultados académicos de los alumnos [17, 18].

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.

Cada año 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 en un factor que incide en su rendimiento académico.

El presente trabajo presenta la experiencia llevada a cabo con los alumnos de la asignatura Fundamentos de la Informática de la carrera Ingeniería Electrónica con Orientación en Señales digitales de la Facultad de Ciencias Físico, Matemáticas y Naturales (FCFMyN). Su dictado es responsabilidad de docentes del Dpto. de Informática de dicha facultad y representa el primer curso de programación que reciben los alumnos de esta carrera. La asignatura se organiza en seis horas semanales de clases teórico-prácticas hasta completar, según el plan de la carrera, un total de 105 horas. Inicialmente las clases teórico-prácticas obligatorias completaban un crédito horario de 90 horas completándose el crédito total con clases de consultas que no resultaban motivadoras para los alumnos y, por lo tanto, no eran aprovechadas.

Del análisis de los resultados académicos obtenidos en los últimos períodos, fue posible identificar algunos aspectos críticos de los alumnos que ingresan año a año y que pudieron influir de forma negativa en su desempeño:

- Las “habilidades académicas” que caracterizan a la mayoría de los alumnos 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 decide el estudio y definición de diferentes propuestas metodológicas que contribuyan a mejorar los procesos educativos involucrados en la enseñanza de la programación y el análisis de la factibilidad de su puesta en marcha. El proceso se desarrolló en forma gradual. En una primera etapa se realizó la incorporación de prácticas de laboratorios complementarias con la intención de motivar a los alumnos y en una segunda etapa se integró el uso del campus virtual definiendo un nuevo canal de comunicación y cuyo potencial provee, además, la posibilidad de definir actividades de aprendizajes significativos que complementan el trabajo que se venía realizando hasta el momento.[2] [10].

3. Justificación de las herramientas utilizadas

Para diseñar las prácticas de laboratorio de programación fue necesario seleccionar cuidadosamente el software con el cual se iba a trabajar. Las distintas herramientas de software que se utilizaron se fueron incorporando en base a su complejidad de uso y en correspondencia con las prácticas de aula, dando soporte a las mismas.

Se incorpora el uso del Lenguaje 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 permite introducir al alumno en la noción de algoritmo y los conceptos básicos necesarios para la construcción de los mismos basado en el paradigma de la programación estructurada. TIMBA es un lenguaje que permite la definición de algoritmos combinando las tres estructuras de control básicas y que consisten, en definitiva, 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, 19].

La aplicación Dia (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. La herramienta 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 [20].

Para iniciar a los alumnos en la construcción de programas o algoritmos computacionales similares a los lenguajes formales de programación, sin tener que lidiar con las particularidades estrictas de sintaxis, se incorpora un software multiplataforma de distribución libre y gratuito denominado PSeInt (Pseudo code Interpreter). Con la inclusión 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 [21].

La simpleza de sintaxis de esta herramienta le facilita al alumno la tarea de escribir algoritmos en un pseudo-lenguaje similar al lenguaje natural haciendo hincapié en la apropiación significativa de conceptos básicos vinculados a 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 [21].

Para añadir el modelo Blended Learning, se incorporó la utilización de la plataforma virtual Moodle que permite definir un canal más de comunicación e intercambio con los alumnos, generando un laboratorio virtual que logra un ambiente común e integrado de trabajo. Fundamentalmente, con la incorporación de esta herramienta se desea, en una primera etapa, promover el trabajo colaborativo de los alumnos con sus pares, sacar ventaja de la independencia de tiempo y espacio, y definir actividades que resulten más motivadoras como así también propuestas de autoevaluación que orienten al alumnos en su proceso de aprendizaje [22].

4. Experiencia

La asignatura Fundamentos de la Informática de la carrera Ingeniería Electrónica con Orientación en Señales digitales corresponde, según el plan de estudios, al segundo cuatrimestre del primer año. Para fomentar la permanencia de aquellos alumnos que no alcanzan los objetivos en un cuatrimestre y reducir el índice de deserción de los mismos, la FCFMyN desarrolla diferentes actividades, como por ejemplo repetir el dictado de la materia en el primer cuatrimestre del año siguiente.

A partir de los contenidos mínimos que la materia debe cubrir, los conceptos que en ella se dictan se organizan en tres ejes fundamentales: introducción a la informática, lógica proposicional y, finalmente, resolución de problemas y definición de algoritmos, el cual es el eje fundamental de la materia. Este eje se basa principalmente en el desarrollo del modelo descriptivo definido por G. Polya el cual establece las necesidades del alumno para aprender a resolver problemas. Es menester definir soluciones siguiendo un enfoque lógico y algorítmico que permita enfrentar al alumno con la problemática de analizar y resolver problemas de carácter general y problemas computacionales, y la transformación de estos últimos a programas que puedan ser resueltos por una computadora. Para desarrollar las cuatro fases de este

modelo se organizó a los contenidos y sus respectivas actividades prácticas en un esquema en forma de espiral positiva, es decir el método de resolución de problemas se abordó en dos momentos sucesivos, incluyendo dos herramientas distintas, siendo la segunda instancia de mayor complejidad y profundidad que la primera. La etapa de Comprender el problema, requiere de un proceso inicial que promueva en el alumno actividades vinculadas a la resolución de problemas y cuya solución, en principio, puede ser expresada de una manera flexible poniendo en práctica la técnica del Refinamiento Sucesivo y utilizando el lenguaje natural. Una vez que se han precisado los límites del problema, determinado los datos, incógnitas y sus relaciones se continúa con la etapa de Desarrollo de un plan que redundará en la definición del algoritmo. Es en esta etapa donde se incluye en la práctica el uso del Lenguaje Timba con el objetivo de favorecer al alumno en la tarea de aprender a caracterizar el comportamiento del flujo de control de un algoritmo. Una vez determinado el mismo se continúa con la codificación respetando la rigurosidad sintáctica propia del lenguaje. Se obtienen así programas codificados en Lenguaje Timba los cuales posibilitan continuar con las etapas de Ejecución del plan y de Examinar la solución definida para evaluar la eficiencia y eficacia de la misma.

Se continúa luego con la aplicación de cada una de las fases de este modelo pero esta vez incluyendo la práctica con PSeInt cuya potencialidad permite la introducción de conceptos fundamentales para la programación como manejo de variables, estructuración de datos, modularización, entre otros.

Esta experiencia se implementó 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 utilizando papel y lápiz. La práctica desarrollada incluyó el uso de la herramienta TIMBA, se realizaron diagramas de flujo con la aplicación Dia y 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.

En la segunda etapa, se continuó con el horario de laboratorio para programación pero, a diferencia de la anterior, con asistencia obligatoria en reemplazo de las clases de consultas que no se aprovechaban. 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 favoreciendo la comunicación, el seguimiento y la retroalimentación personalizada.

5. Análisis de los resultados

Los resultados observados durante esta experiencia surgen de la percepción de los docentes en relación con las formas de aprendizaje de los estudiantes y su desempeño académico considerando diferentes elementos como la interacción, el liderazgo y las respectivas prácticas dentro de las actividades colaborativas. Además, se llevó a cabo

una encuesta a los alumnos que asistieron a la asignatura con el objetivo de conocer, desde su punto de vista, los propios hábitos de estudio y la influencia que la nueva estrategia pudo haber tenido en los respectivos procesos de aprendizaje.

5.1. Desde la perspectiva docente

Se considera como positiva a 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 predispuesta por parte de los mismos, si bien, cabe destacar que este incremento en la participación se produjo en forma paulatina y con diferentes niveles de entusiasmo. Un aspecto a mejorar es la propuesta de actividades tendientes a promover el trabajo colaborativo entre pares ya que predominó la comunicación alumno-docente y prácticamente no existió la comunicación alumno-alumno.
- Con la incorporación del aula virtual y de sus herramientas se logró un registro digital del 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

Para evaluar la experiencia se elaboró una encuesta individual con el propósito de conocer la opinión de los alumnos con respecto a la nueva estrategia planteada en lo que se refiere a la utilización de la plataforma virtual. La encuesta fue respondida de manera anónima y voluntaria.

En la Figura 1 se observan los resultados de las respuestas de alumnos a la consulta para determinar los conocimientos previos en relación al manejo de herramientas virtuales. El 67% de los encuestados manifiestan estar familiarizados con el uso de la plataforma virtual ya que se utiliza en otras asignaturas previas.

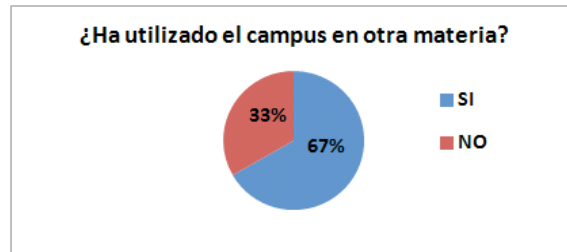


Figura 1

En relación a la frecuencia con la que el alumno accedía a la plataforma, en la Figura 2 es posible observar que el 50% de los alumnos reconocen haber accedido frecuentemente al campus evidenciando que la gran mayoría incorporaron al aula virtual como un recurso importante de consulta. En la Figura 3 se observa que la mayoría reconoce a la incorporación del campus como una continuidad de la práctica de aula que beneficie al proceso de aprendizaje y la minoría lo considera una sobrecarga de dedicación.

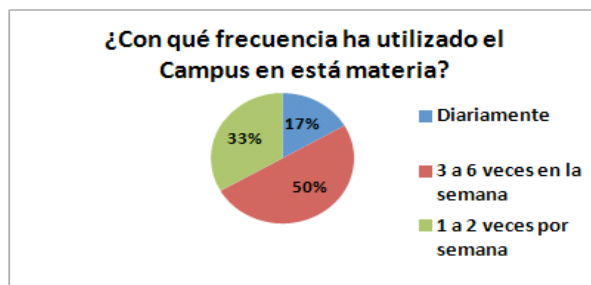


Figura 2

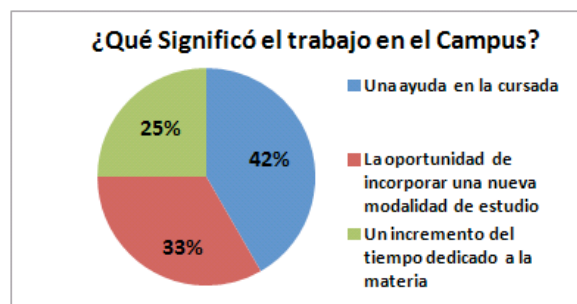


Figura 3

En la Figura 4 se observa que el 92% de los alumnos reconocen como beneficioso la incorporación del uso del aula virtual siendo casi insignificante el porcentaje de alumnos que declaran no tener una opinión formada .

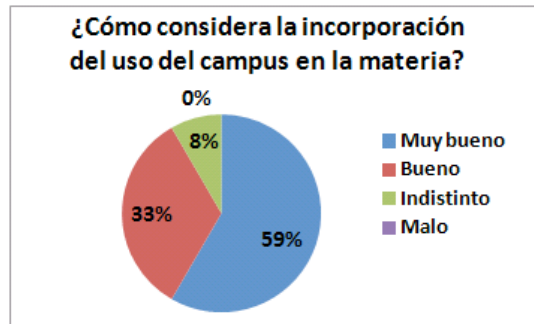


Figura 4

La Figura 5 muestra que la mayoría de los alumnos han accedido al campus con el propósito de cumplir con los requerimientos que fueron realizados por los docentes, en segundo lugar y en un porcentaje significativo, le interesó la retroalimentación del docente al alumno. Un grupo pequeño utilizó al campus como un canal para obtener el material de estudio y un porcentaje casi nulo incorporó el uso de los foros con lo que resulta evidente que un aspecto a mejorar es el promover actividades que favorezcan el trabajo colaborativo.

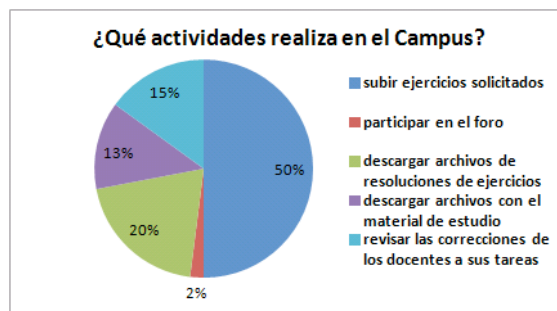


Figura 5

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 que surgen del uso del campus como el contar con todo el material de estudio en un mismo lugar, la posibilidad 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.

6. 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-aprendizaje 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), 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 [22,23].

Referencias

1. Polya, George, *Cómo plantear y Resolver problemas* - Editorial Trillas – ISBN 968-24-0064-3 (1965).
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. <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.
16. Wing, J. M. Computational Thinking and Thinking about Computing - *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences* 366 (1881) (2008) p. 3717-3725.
17. Stimolo, M. I., Caro, N. P. B-Learning: Implementación de Recursos de Internet en la Enseñanza de Estadística en la Facultad de Cs. Económicas - UNC - VI Congreso de Tecnología en Educación y Educación en Tecnología. TE&ET (2011).
18. González Mariño, J. C. B-Learning utilizando software libre, una alternativa viable en Educación Superior. *Revista Complutense de Educación* 17 (1) (2006) p. 121-133.
19. Lenguaje TIMBA (*Terribly Imbecile Machine for Boring Algorithms*) – Informe Técnico – UNSL.
20. Diagram Editor: <http://dia-installer.de/index.html.es>
21. Herramienta PSeInt: <http://pseint.sourceforge.net>
22. Plataforma Moodle: <http://moodle.org>
23. Virtual Programming Lab for Moodle: <http://vpl.dis.ulpgc.es>

