

Estrategias de Enseñanza Colaborativa para un Curso de Programación de Primer Año de la Lic. En Sistemas

Edith Lovos¹, Alejandro Gonzalez², Inés Mouján¹,
Rodolfo Bertone², Cristina Madoz²

¹Universidad Nacional de Río Negro, Sede Atlántica, Av. Don Bosco y Leloir
8500 Viedma, Río Negro, Argentina {[elovos](mailto:elovos@unrn.edu.ar), [imoujan](mailto:imoujan@unrn.edu.ar)}@unrn.edu.ar

²Instituto de Investigación en Informática III-LIDI. Facultad de Informática, Universidad
Nacional de La Plata, 50 y 120
1400 La Plata, Buenos Aires, Argentina
alejandro.gonzalez@presi.unlp.edu.ar, {[pbertone](mailto:pbertone@lidi.info.unlp.edu.ar), [cmadoz](mailto:cmadoz@lidi.info.unlp.edu.ar)}@lidi.info.unlp.edu.ar

Abstract. Los futuros Licenciados en Sistemas se encontrarán con un ámbito laboral en el cual uno de los rasgos distintivos en el desarrollo de software es considerarlo una actividad colaborativa. Teniendo en cuenta este aspecto, el artículo presenta los motivos que generan la necesidad de incorporar estrategias de enseñanza colaborativa en los cursos de programación de los primeros años de las carreras de informática, a fin de generar conciencia de trabajo en grupo para el desarrollo de las actividades informáticas. Se presentan además, las diferentes estrategias de enseñanza utilizadas en la enseñanza de la programación de nivel universitario y la propuesta de enseñanza de programación para un curso de primer año de la Lic. en Sistemas de la UNRN.

Keywords: programación, enseñanza, colaborativa, sistemas

1 Introducción

Abordar la enseñanza y aprendizaje de la programación de algoritmos para resolver problemas usando una computadora implica una complejidad que precisa ser tratada desde el inicio de la formación de grado en las carreras de Sistemas. A medida que los estudiantes avanzan en la carrera, profundizan los conceptos vinculados al desarrollo de un producto de software, que es en sí mismo una actividad colaborativa. Teniendo en cuenta la forma de trabajo de los futuros profesionales, surge la necesidad de utilizar estrategias colaborativas en el ámbito de la enseñanza y del aprendizaje de la programación.

El presente trabajo indaga acerca de los beneficios que aportan el uso de la tecnología de la información y comunicación (TIC), como mediadora de estrategias colaborativas en la enseñanza y aprendizaje de la programación en los primeros cursos de las carreras informáticas de nivel universitario; y presenta los avances de una investigación (tesis de maestría) llevada adelante en la Lic. en Sistemas de la UNRN. Para ello se analizan además, las características presentes en la enseñanza tradicional de la programación de nivel universitario y se definen estrategias que aborden el trabajo colaborativo.

1.1 Motivación

La resolución de problemas utilizando una computadora se descompone en varias etapas, el modelado del problema, selección de las estructuras de datos que mejor se adapten al mismo, escritura del algoritmo, implementación del mismo en un lenguaje de programación de alto nivel, compilación, ejecución y depuración del programa resultante [1]. En este sentido, es importante contar con aquellos conocimientos teóricos que permitan optimizar el uso de los recursos disponibles para el desarrollo de un programa (memoria de la computadora por ejemplo), sin perder de vista que la mejor forma de adquirir destreza en programación es a través de la actividad práctica. Una vez adquiridos los conocimientos mínimos de programación, una metodología colaborativa permite potenciar la enseñanza de la programación pues el trabajo en pequeños grupos maximiza el trabajo individual y colectivo [2].

En un contexto educativo, el aprendizaje colaborativo es un modelo de aprendizaje interactivo que invita a los miembros a sumar esfuerzos, talentos y competencias mediante una serie de transacciones que les permitan alcanzar juntos las metas propuestas [3].

Partiendo de estos conceptos se entiende que los alumnos pueden trabajar colaborativamente, en la solución de un problema resoluble por computadora, sin que esto implique descartar el trabajo individual, sino que se propone como una estrategia de apoyo al aprendizaje con la intención de fortalecer el desarrollo global del alumno [4].

A entender de los autores Marina Velasco y Fidel Mosquera[5] en su trabajo “Estrategias didácticas para el Aprendizaje Colaborativo”, si se analiza el aprendizaje colaborativo desde el punto de vista sociológico, este representa un atributo, un componente y un soporte esencial del aprendizaje social. Aprender con otros y de otros, hace referencia en lo que la psicología se conoce como zonas de desarrollo próximo [6], y en los enfoques pedagógicos se plantea como estrategia dialógica[7]. Ambas perspectivas permiten valorar el trabajo que desempeña un sujeto con otros. En pos de un aprendizaje determinado, la importancia que se le asigna al compartir experiencias y labor con otros individuos abre las puertas para generar estrategias de enseñanza y aprendizaje centrado en el conocimiento y la producción colectiva.

Desde la perspectiva de los docentes, a través del aprendizaje colaborativo guiado, es posible que obtener un conocimiento más detallado de los problemas que presentan los alumnos en la resolución de un problema. De esta forma, los alumnos obtienen una ayuda más personalizada, y un conocimiento más profundo empleando menos tiempo en asimilar los inconvenientes presentados[8].

En este punto es importante señalar que un individuo que participa en el aprendizaje colaborativo, consigue conocer su propio ritmo de aprendizaje y con ello, compararlo y compartirlo con el resto. Este conocimiento le permite aplicar las estrategias metacognitivas para mejorar su aprendizaje, incrementando su motivación al compartir responsabilidades dentro del grupo y con ello su autoestima[9].

1.2 Ambientes Colaborativos

Las herramientas colaborativas pueden fortalecer aspectos como el razonamiento, el auto-aprendizaje y el aprendizaje colaborativo [4]. En este sentido, los entornos de desarrollo para el aprendizaje de la programación, persiguen la idea de proveer a los usuarios (estudiantes) de un ambiente que les facilite las tareas relacionadas con el desarrollo de software.

Las herramientas comerciales que se utilizan a nivel profesional, para el desarrollo del software, presentan una amplia cantidad de opciones y de información que los alumnos que recién se inician en una carrera de sistemas, no pueden comprender tan fácilmente porque aún no tienen los conceptos necesarios para manipularlas[10].

El hecho que a nivel profesional existen herramientas de software que permiten el desarrollo colaborativo del software, genera la necesidad de preparar a los futuros Licenciados en Sistemas en el uso de estos ambientes en el contexto educativo. A continuación se detallan, algunos ambientes para el desarrollo colaborativo del software y experiencias en el ámbito académico, resultantes de la indagación realizada hasta el momento

- EclipseGavab: versión personalizada del entorno de desarrollo Eclipse especialmente diseñada para la docencia; que permite la implementación del Aprendizaje Basado en Proyectos a través de la utilización de características colaborativas. Soporta los lenguajes de programación Pascal, C y Java, los cuales son ampliamente utilizados en la enseñanza de la programación de los primeros años de las carreras informáticas [13].
- Virtual Programming Lab (VPL): producto creado por el Departamento de Informática y Sistemas, de la Universidad de Las Palmas de Gran Canaria; que permite la gestión de prácticas de programación sobre Moodle, de esta forma se permite la incorporación del ambiente de desarrollo de software al aula virtual de las materias donde se utiliza [14].
- Ambiente Instruccional SABATO: herramienta informática personalizada que integra los paradigmas de enseñanza: Aprendizaje Basado en Problemas (PBL) y el aprendizaje colaborativo apoyado en computadora (CSCL) [15].

En cuanto a las Experiencias de Trabajo Colaborativo en la Enseñanza-Aprendizaje de la Programación es posible citar:

- Aplicación de Técnicas de Aprendizaje Cooperativo en la Enseñanza del Desarrollo de Software[11]: Se basa en la aplicación de las técnicas de aprendizaje cooperativo como rompecabezas, estudio de casos, rally y proyecto en grupo en los temas de análisis, diseño e implementación orientado a objetos.

- Trabajo cooperativo en el aprendizaje de la programación del shell de UNIX [12] es una experiencia de trabajo colaborativo aplicada a las prácticas de laboratorio, focalizada en la enseñanza-aprendizaje de la programación del shell de Unix con la intención de lograr un mayor grado de compromiso por parte de los alumnos en las actividades de laboratorios, un seguimiento más detallado de sus trabajos y motivarlos en el aprendizaje de los contenidos

2 La programación: su enseñanza

En la educación de las Ciencias de la Computación, la enseñanza de la programación es uno de los pilares y uno de los primeros cursos que deben tomar los alumnos ingresantes [17]. Para autores como Costelloe[18] y Lahtinen & Ala-Mutka [19], la enseñanza y aprendizaje de programación en estos cursos, es una actividad intelectual compleja y dificultosa, tanto para los alumnos como para quienes llevan adelante la enseñanza.

La resolución de un problema usando una computadora (por medio de algoritmos) requiere un pensamiento analógico pues se debe utilizar una representación de la realidad a través de un modelo[20]. Así, la solución al problema expresada en un programa implica varios pasos: obtener un modelo del problema, diseñar un algoritmo y la implementación del mismo en un lenguaje de programación. Un programa entonces, puede definirse como el conjunto de instrucciones que la computadora tendrá que seguir para resolver el problema, y donde el responsable de proporcionarlas en el orden correcto, es el programador.

Para Costelloe[18], la escritura de un programa involucra dos etapas:

- resolución del problema
- implementación

En la primera, se diseña la solución al problema. En la segunda, la solución propuesta se traduce en instrucciones del lenguaje de programación elegido.

Así, un alumno de un curso de programación, necesitará de dos habilidades diferentes para aprender programar: resolución de problemas e implementación[18]

En los cursos de programación de los primeros años de las carreras informáticas, se espera que el alumno al finalizarlos, conozca y maneje conceptos básicos de programación: estructuras de control (secuencia, selección, iteración), estructuras de datos estáticas y dinámicas (registros, arreglos, listas), y alguna técnica de programación.

Para la enseñanza de la programación, existen diversas estrategias y herramientas . Costelloe[18] en su investigación “Teaching Programming. The State of the Art” ha realizado una clasificación de las mismas en :

- **Clases magistrales y de laboratorio:** a entender de la autora, en las clases teóricas se tratan temas que pueden ser reforzados con el material bibliográfico recomendado por la cátedra. Luego en el laboratorio se trabajan sobre la resolución de ejercicios que ponen en práctica los conceptos vistos en las teorías. De esta forma, en este ambiente, los alumnos se comportan, en general como receptores pasivos de la información, con mínima interacción, en particular en los casos donde las clases son

numerosas. Respecto de este método tradicional de enseñanza, Boyle et al [16] afirman, “*las estrategias de enseñanza tradicionales restringen las habilidades naturales que tienen los estudiantes, para la resolución de problemas*”.

- **Software Visualization:** consiste en representar las ideas abstractas presentes en el código de un programa en representaciones visuales que asistan al programador o usuario a comprender el artefacto observado (algoritmo, pasos de la ejecución de un programa). En la enseñanza de programación se utilizan principalmente tres tipos de visualizaciones:
 - **Visualización de Programas:** se focaliza en el uso de representaciones gráficas de un programa en ejecución y sus datos. El nivel de abstracción presente en este tipo de visualización es muy bajo, ya que existe una representación directa del código y los datos del programa. El usuario puede realizar interacciones predefinidas como cambiar la velocidad de ejecución o los datos de entrada del programa.
 - **Animación de Programas:** a diferencia de la visualización de programas, aquí el foco está puesto en mostrar las operaciones fundamentales del algoritmo. En esta categoría se encuentran por ejemplo, los programas que permiten visualizar el comportamiento de los algoritmos de ordenación (quicksort, heapsort, etc) que se utilizan comúnmente en el aprendizaje de estructuras de datos.
 - **Programación Visual:** se define como el uso de componentes visuales (gráficos, animaciones o iconos) para la construcción del programa[18].

La visualización entonces, se usa para clarificar conceptos complejos y permite que los alumnos desarrollen modelos mentales de esos conceptos. A entender de Ben Ari[25], “*el constructivismo afirma que durante el aprendizaje, cada individuo crea estructuras cognitivas (modelos), los datos sensoriales se combinan con el conocimiento existente para crear nuevas estructuras cognitivas, que servirán de base para futuras construcciones*”. De esta forma Costelloe[18], compara la visualización del software con el constructivismo, en el sentido que la primera, asiste en la construcción del conocimiento creando modelos de conceptos complejos: el alumno puede controlar la actividad y recibe respuesta en forma inmediata. En ambos casos, existe una construcción activa del conocimiento.

- **Robots:** el uso de robots en la enseñanza de programación, no solo facilitan el aprendizaje de forma constructiva sino que además proveen de la posibilidad de experimentar con máquinas reales y no solo simuladas. Así los alumnos pueden trabajar en un ambiente complejo haciéndolo más interesante [21]. Los alumnos aprenden conceptos básicos de programación en forma intuitiva y lúdica, explorando instrucciones y sentencias del lenguaje para su manipulación, experimentando sus resultados en forma interactiva y mediante la observación directa del robot.
- **Aprendizaje Basado en Problemas (ABP):** es un método didáctico que se apoya en la enseñanza, denominada aprendizaje por descubrimiento y construcción, donde el protagonista es el propio estudiante, quien debe apropiarse del proceso de aprendizaje: buscar información, seleccionarla,

organizarla e intentar resolver los problemas que se le plantean[22]. Los alumnos trabajan en grupos pequeños, con la asistencia del docente (tutor) que facilita recursos para resolver el problema, pero la resolución en sí depende de las elecciones que realice el grupo respecto a que necesitan aprender de las áreas relevantes. La aplicación de este método de enseñanza en las ciencias de la computación, se ve favorecido por factores como el hecho de que la computación está dirigida por problemas y el aprendizaje debe ser continuo debido a la naturaleza cambiante de la industria informática, por otra parte el desarrollo de proyectos en grupos es predominante en el ámbito profesional. El método favorece la autonomía y las habilidades para el trabajo en equipo, una característica que no es tenida en cuenta en los métodos tradicionales de enseñanza[18].

- **Aprendizaje Cognitivo:** Es un modelo de aprendizaje basado en la teoría de cognición situada, que sostiene que el aprendizaje esta ligado naturalmente a actividades reales, contexto y cultura [23].

Para Chalk[26], el advenimiento de la web y Java, produjeron una proliferación de applets interactivos y bibliotecas de software que pueden ser usadas con fines educativos, animando de esta forma la estrategia de enseñanza experimental. El mismo autor ha realizado estudios experimentales aplicando los recursos disponibles en la web, en la enseñanza de materias de las ciencias de la computación, que incluyen el uso de ambientes interactivos de aprendizaje basados en Web (webworlds). Sobre estas experiencias el autor ha concluido : “la web, el soporte colaborativo y las herramientas de modelado en conjunto, proveen el significado para que pueda desarrollarse un aprendizaje virtual en la práctica de ingeniera del software”.

Esta forma de aprendizaje presenta los siguientes beneficios [18]:

- promover en los alumnos el desarrollo de las habilidades meta-cognitivas, a través de la propia reflexión y el auto-análisis.
- promover el aprendizaje y la colaboración en un contexto social.
- promover el desarrollo de otras habilidades como la verbalización y la comunicación.

Si a las técnicas y estrategias constructivistas expuestas, se les suma la posibilidad de trabajar en un ambiente colaborativo, el aprendizaje se convierte en un proceso social, donde los estudiantes aprenden trabajando en conjunto con otros, observando cuales son las estrategias de resolución de problemas que aplican sus compañeros, generando un espacio motivador para la discusión de ideas y toma de decisiones a lo largo de todo el proceso[7].

3 Curso de Programación: propuesta de enseñanza

Este artículo se enmarca en el desarrollo de la tesis de maestría Tecnología Informática Aplicada en Educación de la Facultad de Informática de la UNLP. En este apartado presentamos algunos avances respecto a su marco teórico y las estrategias de enseñanza colaborativa, mediada por tecnología que se implementarán.

El objeto de análisis será un curso de programación del primer año de la Licenciatura en Sistemas de la UNRN: Programación I. Desde los inicios de la carrera (2009), del mismo participan alumnos recientemente egresados del nivel medio, cuyas edades oscilan entre los 17 y 20 años y para la mayoría de ellos, esta materia es su primer contacto con el aprendizaje de la programación.

El curso es de carácter presencial, con clases teóricas y prácticas (laboratorio). En las clases teóricas se desarrollan los conceptos teóricos del plan de estudio (resolución de problemas, estructuras de control, modularización, estructuras de datos) sobre ejemplos prácticos. Tanto en las clases teóricas como prácticas, se propone la participación colaborativa de los alumnos en la resolución de problemas, para lo cual se plantean actividades grupales, como el desarrollo de actividades prácticas entregables (APE). Estas últimas consisten en la resolución de problemas de mediana complejidad, cuya solución puede implementarse usando un lenguaje de programación de alto nivel tipo Pascal.

El curso tiene como soporte un aula virtual sobre Moodle. Allí, los alumnos pueden encontrar el material teórico práctico, cronogramas de actividades, novedades, etc. Este soporte, se convierte en un espacio extra de comunicación para todos los participantes del curso (docentes, alumnos). Las APE se entregan y se evalúan a través de la plataforma. Las APE hacen uso del ABP y tienen como objetivo mostrar el desarrollo cognitivo alcanzado a partir de los conocimientos de partida, los debates y procesos generados para la toma de decisiones en conjunto, y el producto generado[27].

La UNRN es una universidad en pleno proceso de formación, así uno de los problemas con los que nos enfrentamos en las clases prácticas es la falta de espacio y recursos para realizar las actividades de laboratorio. Por esta razón, proponemos que las APE se desarrollen en todas sus fases, a través de alguna herramienta colaborativa sobre el aula virtual, de forma análoga a como se trabaja en el desarrollo de software a nivel profesional.

Para lograr estos objetivos nos proponemos:

- Identificar interfaces colaborativas que permitan probar algoritmos en el ámbito de la programación, en particular basadas en WIKI y herramientas de groupware para programación.
- Construir instrumentos que permitan la evaluación de las interfaces y de la participación de los alumnos del grupo de trabajo colaborativo.

Al finalizar el curso, los alumnos novatos habrán adquirido los conceptos y habilidades necesarias que le permitan iniciarse en la práctica de la programación: resolución de problemas, capacidad de sintetizar e integrar la información, así como también la planificación del tiempo, la comunicación, la toma de decisiones y el desarrollo de la creatividad, estas últimas, habilidades muy requeridas en el ámbito profesional del desarrollo de software.

Conclusiones y trabajo futuro

En esta primera etapa se ha revisado la bibliografía referida al uso de las TIC como mediadoras de estrategias de enseñanza colaborativa en los cursos de programación

de nivel superior y se han planteado las diferentes estrategias y herramientas que se utilizan con la intención de facilitar y potenciar el aprendizaje de una de las temáticas fundamentales en las ciencias de la computación (Costelloe[18], Ben-Ari[25]. Chalk[26]).

Revisando las experiencias de implementación del curso, objeto de estudio, se analizaron los inconvenientes mas comunes y se generó una posible estrategia que utilice herramientas colaborativas que permitan la participación virtual y activa de los alumnos.

Se definieron las características deseables para las herramientas de manera que faciliten la actividad grupal de un equipo de desarrollo de software a nivel educativo

La estrategia colaborativa propuesta se implementará haciendo uso de las herramientas provistas por Moodle (Wiki, foros, mail) para las etapas de resolución de problemas y diseño de la solución. En la etapa de programación se usará VPL. La elección de VPL se fundamenta en el hecho de que permite el seguimiento y la orientación personalizada por parte del tutor, generando de esta forma una instancia de feedback muy importante en el proceso de aprendizaje.

Como líneas de trabajo futuro proponemos:

- Poner a prueba la estrategia en la cohorte 2012, durante el segundo cuatrimestre
- Elaborar instrumentos que permitan evaluar la experiencia; teniendo en cuenta tanto el resultado final de las APE (programa de computadora), como así también el proceso de aprendizaje a nivel grupal e individual.
- Revisar la estrategia con la intención de poder aplicarla a otros cursos de similares características.

4 Referencias

1. Joyanes Aguilar, Luis. (2003). Fundamentos de Programación. Libro de Problemas. Mc Graw Hill.
2. Johnson, D. W., Johnson, R., & Holubec, E. (1993). Circles of learning. Edina, MN: Interaction Book Company.
3. Araiza Vázquez María de Jesús, Dörfer Claudia, Castillo Corpus Rosalinda (2011). "Una Experiencia de Desarrollo y Aplicación de las Competencias Profesionales en los Estudiantes de la Licenciatura de Tecnologías de Información". Universidad de Nueva León. Presentado en XIV Encuentro Iberoamericano de Educación Superior a Distancia en la Universidad Técnica Particular de Loja. <http://memorias.utpl.edu.ec/aiesad-2011>
4. Collazos O, César Alberto, Guerrero Luis, Vergara. Adriana. (2001). Aprendizaje Colaborativo: un cambio en el rol del profesor. <http://www.dcc.uchile.cl/~luguerre/papers/CESC-01.pdf>
5. Velazco Marina, Mosquera Fidel (2007). Estrategias didácticas para el Aprendizaje Colaborativo. http://acreditacion.udistrital.edu.co/flexibilidad/estrategias_didacticas_a_prendizaje_colaborativo.pdf
6. Vygotski, Lev. S. (1978) . El Desarrollo de los procesos psicológicos superior. Barcelona . Grupo editorial Grijalbo.
7. Freire, Paulo. (1970) La educación como práctica de la Libertad, Siglo XXI: México.

8. Gallego Micael, Gortázar Francisco (2009). EclipseGavab, un entorno de desarrollo para la docencia online de la programación. XV JENUI. Barcelona, 8-10 de julio de 2009. <http://jenui2009.fib.upc.edu/>.
9. Carrio Pastor María Luisa. (2007). “Ventajas del uso de la tecnología en el aprendizaje colaborativo” en *Revista Iberoamericana de Educación* n.o 41/4 – 10 de febrero de 2007. Organización de Estados Iberoamericanos para la Educación, la Ciencia y la Cultura (OEI)
10. Pérez Pérez Juan Ramón, Paule Ruiz M^a del Puerto, Cueva Lovelle Juan Manuel (2006) “Capítulo 3. Sistemas orientados a la mejora de la calidad del software” en congreso IV International Conference on Multimedia and Information & Communication Technologies in Education (m-ICTE2006).
11. Campos Pedro, Flores Luis Alberto, Pow-Sang José Antonio, Zapata Claudia (2007). Aplicación de Técnicas de Aprendizaje Cooperativo en la Enseñanza del Desarrollo de Software. VI Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento Lima, Perú. <http://inform.pucp.edu.pe/~jpowsang/papers/campos-jiisic07.pdf>
12. Prieto Izquierdo Óscar J., Hurtado M^a Aránzazu, Pascual Vivaracho Carlos (2004). “Trabajo cooperativo en el aprendizaje de la programación del shell de UNIX”. <http://giac.upc.es/JAC10/04/JAC04-OPI.htm>.
13. Gallego M., Gortázar F. (2009). EclipseGavab, un entorno de desarrollo para la docencia online de la programación. JENUI 2009. Barcelona, 8-10 de julio de 2009 ISBN: 978-84-692-2758-9. <http://jenui2009.fib.upc.edu/>
14. Rodríguez del Pino, J.C., Royo Rubio E., Hernandez F. Igueroa. VPL: Laboratorio virtual de programación para Moodle. En Actas de las XVI Jornadas de Enseñanza Universitaria de Informática, Jenui 2010, págs. 429–435, Santiago de Compostela, Julio 2010.
15. Jiménez Builes J.A., Pavony Meneses M., Alvarez Serna, A. F. Entorno de integración de PBL y CSCL para la enseñanza de algoritmos y programación en ingeniería. En revista Avances en Sistemas e Informática, ISSN 1909-0056, Vol. 5, N^o. 3, 2008, págs. 189-194. <http://www.revista.unal.edu.co/index.php/avances/article/view/10112>
16. Boyle, T. (2000) Constructivism: A Suitable Pedagogy for Information and Computing Sciences?. <http://www.ics.ltsn.ac.uk/pub/conf2000/Papers/tboyle.htm>
17. Matthíasdóttir, Á.(2006): How to teach programming languages to novice students? Lecturing or not?, Proceedings of the International Conference on Computer Systems and Technologies, June 15-16, University of Veliko Tarnovo, 2006, Bulgaria.
18. Costelloe, E. (2001). Teaching Programming. The State of the Art. Department of Computing, Institute of Technology Tallaght, Dublin 24. CRITE Technical Report, 2004a. https://www.scss.tcd.ie/disciplines/information_systems/crite/crite_web/publications/sources/programmingv1.pdf
19. Lahtinen E, Ala-Mutka K, et al.(2005) A Study of the Difficulties of Novice Programmers. 10Th annual SIGCSE conference on Innovation an technology in computer science education ItiCSE '05
20. Lovos, E. Gibelli, Tatiana. (2012). La Simulación como Recurso de Enseñanza-Aprendizaje en el Análisis de Eficiencia de Algoritmos. *Revista Iberoamerica de Tecnología en Educación y Educación en Tecnología (TE&ET)* Nro 7 ISSN 1850-9959 RedUNCI-UNLP .Pag. 36-41
21. Linder, et al, (2001) “Facilitating Active Learning With Inexpensive Mobile Robots,” *Journal of Computing in Small Colleges*,16, 4. <http://delivery.acm.org/10.1145/380000/378656/p21-linder.pdf?>

[key1=378656&key2=7062133701&coll=GUIDE&dl=ACM&CFID=15363060&CFTOKEN=16364368](#)

22. Restrepo Gomez, B.(2005). Aprendizaje basado en problemas (ABP):una innovación didáctica para la enseñanza universitaria. Educadores, Bogotá, Colombia, v. 8, p. 9-19, 2005. redalyc.uaemex.mx/redalyc/pdf/834/83400803.pdf.
23. Brown; Allan Collins; Paul Duguid. (1989). Situated Cognition and culture of Learning. Educational Researcher, Vol. 18, No. 1. (Jan. - Feb., 1989), pp. 32-42 . <http://people.ucsc.edu/~gwells/Files/Courses Folder/ED 261 Papers/Situated Cognition.pdf>
24. Materson T. R, Meyer, M. R. (2001). Journal of Computing Sciences in Colleges Volume 16 Issue 4, 2001. Pages 74-86
25. Ben-Ari, M.(2001). Constructivism in Computer Science Education. Journal of Computers in Mathematics and Science Teaching, 20(1), 45-73. Norfolk, VA: AACE. <http://www.it.uu.se/edu/course/homepage/datadidaktik/ht06/teaching/Moti-Ben-Ari-jcmst.pdf>
26. Chalk, P. (2000). Apprenticeship learning of software engineering using Webworlds. ITiCSE '00 Proceedings of the 5th annual SIGCSE/SIGCUE ItiCSE conference on Innovation and technology in computer science education Pages 112-115 . <http://ebookbrowse.com/p112-chalk-apprenticeship-learning-of-software-engineering-using-webworlds-pdf-d289928416>
27. Badía, A, García, C (2006). Incorporación de las TIC en la enseñanza y el aprendizaje basados en la elaboración colaborativa de proyectos. Revista de Universidad y Sociedad del Conocimiento . Vol 3- Nro 2. Pag 42-54