

PROYECTO: SISTEMAS INTELIGENTES APLICADOS A LA ENSEÑANZA DE LA PROGRAMACION EN INGENIERIA

Carlos Alberto Bartó, Laura Cecilia Díaz

Laboratorio de Educación Virtual
Departamento de Computación
Facultad de Ciencias Exactas, Físicas y Naturales
Universidad Nacional de Córdoba
Av. Vélez Sarsfield 1611, Córdoba
0351-4334409

cbarto@gmail.com , lcd_ic@yahoo.com.ar

Resumen

El proyecto objeto de esta presentación, se desarrolla actualmente en la Facultad de Ciencias Exactas, Físicas y Naturales de la Universidad Nacional de Córdoba. Ha sido presentado para su acreditación ante la Secretaría de Investigación y en Ciencia y Tecnología de esta Universidad en la convocatoria bianual 2012-2013 de proyectos I+D. Es el primer antecedente formal en esta línea de investigación aplicada a la enseñanza en Ingeniería en esta Casa de Estudios.

Su mayor relevancia consiste en el impacto que se espera de él para las carreras de Ingeniería, especialmente la de Computación, tanto para el rendimiento académico de los estudiantes como para el fortalecimiento de los recursos humanos – investigadores, docentes y estudiantes- en investigación.

El eje central consiste en la utilización de sistemas tutores inteligentes y aplicaciones con robots para la enseñanza de Programación en Ingeniería.

Es importante destacar que tanto los integrantes de este equipo de investigación como otros docentes e investigadores de la Casa, han realizado acciones de investigación, transferencia y evaluación en la aplicación de Tecnologías de Información y Comunicación como herramientas soporte para la enseñanza presencial de grado. Estas acciones se han visto favorecidas desde 2006 en el marco del Programa para el Mejoramiento de la Enseñanza en Ingeniería que contempló la adquisición de un importante volumen de recursos informáticos. Por otra parte, los integrantes de este equipo cuentan además con antecedentes en el sentido específico de esta línea de investigación. Éstas son las fortalezas que alentaron a la propuesta que aquí se presenta.

Palabras Clave: Tutores Inteligentes, Robótica, Ingeniería

Contexto

En la FCEFYN de la UNC, atento a los procesos de acreditación de CONEAU, este proyecto es parte de un compromiso tomado para el fortalecimiento en acciones de Investigación para la carrera de Ingeniería en Computación. Por otra parte, es una línea de investigación que atiende tanto a la formación de docentes, investigadores y alumnos como a las deficiencias en el rendimiento académico de los estudiantes de las carreras de Ingeniería.

Para la concreción de este trabajo se dispone de las siguientes facilidades:

- Infraestructura básica necesaria para su realización consistente en espacio, equipamiento del Laboratorio de Educación Virtual (LEV) del Departamento de Computación, dependiente de la F.C.E.F.y N con equipamiento informático suficiente a nivel de servidores y estaciones de trabajo.
- Infraestructura del Laboratorio de Lego equipado con 10 robots y software para el desarrollo de actividades de programación

Introducción

Planteamiento del problema

Los procesos de evaluación en las universidades argentinas que se desencadenaron a partir de la Ley de Educación Superior (Ley 24521, 1995) pusieron en evidencia déficits institucionales y académicos, despertando la necesidad de mejorar los indicadores de rendimiento académico. En este sentido, en la FCEFYN de la UNC, la acreditación de las carreras

de Ingeniería se realizó a partir de la Resolución N° 1232/01 del Ministerio de Educación. En este escenario, el Proyecto de Mejoras en la Enseñanza de las Ingenierías (PROMEI), surgió como respuesta a las necesidades observadas en el Laboratorio de Informática, permitiendo la creación del Laboratorio de Educación Virtual (LEV) bajo la plataforma Moodle .

Sin embargo, la mejora en infraestructura informática e incorporación de TIC'S como soporte a las clases presenciales, es una acción que debe ser complementada con otras para lograr una mejora en los indicadores de rendimiento académico en cursos de Introducción a la Programación. En investigaciones actuales (Yadin, 2011) se destacan las acciones que mejoraron estos indicadores. Éstas atendieron aspectos asociados a la problemática de la incorporación de un lenguaje formal y a los procesos de abstracción necesarios en los mismos.

Muchas otras investigaciones han tratado de hallar soluciones alrededor de las ideas de Piaget y de la psicología cognitiva para el aprendizaje de los principios fundamentales de la programación, sin embargo, se observaron dificultades para alcanzar el cambio conceptual deseado en los estudiantes de estos cursos (Carretero, 2005) .

Por otra parte, la detección temprana de las capacidades de los estudiantes para obtener buenos rendimientos en el aprendizaje de la programación ha sido reflejado como factor significativo en (Dehnadi y Bornat, 2006) y posteriormente en (Dehnadi y Bornat, y Hamilton, 2008). La tendencia universal, revela que la gran dificultad reside en la naturaleza formal de los lenguajes de programación, que los distancia de los lenguajes naturales y su enseñanza. A partir de la teoría de Los Modelos Mentales (Johnson-Laird, 1993) los autores diseñan un experimento mediante pruebas que ponen de manifiesto todos los posibles modelos que pueden generarse y los categorizan, demostrando que las dificultades para el cambio de concepciones erróneas hacen casi inútil la idea de un aprendizaje basado en la comprensión formal.

Atendiendo a este escenario, en los cursos de Introducción a la Programación de la FCEFYN de la UNC se centró la estrategia de mejora en la elección de un lenguaje de programación que mantuviera al pensamiento concreto como un factor crítico (Bartó y Weber, 2008), asociando también este concepto, a la práctica inherente a la ingeniería de construcción de productos que satisfacen los objetivos perseguidos. El producto obtenido se convierte en una evidencia objetiva

de que la planificación y los conocimientos son válidos, como por ejemplo, en la programación de robots.

Además, en la FCEFYN se observó que, si bien el conocimiento previo de lógica formal puede utilizarse como predictor del éxito académico, el escaso desempeño sistemático en las evaluaciones que manifiestan los estudiantes los desalientan luego a continuar con sus estudios (Bartó, 2010)

Las evaluaciones permanentes sobre la comprensión de textos escritos en el lenguaje Python o C++ y la posterior evaluación de traducciones de la solución de un problema dado como especificaciones en un castellano técnico al lenguaje de programación, han producido algunas mejoras en los cursos de programación (Cebollada y Verdaguer, Eschoy y Bartó, 2010) .

Sin embargo, la falta de mejoras significativas en los resultados, se ponen de manifiesto en los indicadores de rendimiento académico en estos cursos (Díaz, Algorry y Natali, 2011),

Los modelos mentales de los estudiantes no son equivalentes a los modelos conceptuales, propios de la enseñanza de los conocimientos científicos, y aun cuando guardan una gran relación con ellos, resulta muy difícil construir relaciones entre ambos tipos de modelos.

Las diferentes formas de representación del conocimiento para describir los modelos conceptuales, tales como las redes semánticas u ontologías, las reglas de producción o relaciones causales, y sus respectivos diagramas e imágenes (relaciones geométricas y analógicas estructurales) pueden compararse con los lenguajes de programación de alto nivel, y ponen de manifiesto las dificultades señaladas.

Esta distancia entre Modelos Mentales y Modelos Conceptuales para los cursos de programación, materializada en las dificultades para abstraer y para familiarizarse con un lenguaje formal, hace muy difícil, sino imposible, tanto su enseñanza como su aprendizaje hasta lograr su asimilación y acomodación.

Estudio de antecedentes

La búsqueda de soluciones a los problemas planteados, atendiendo a los recursos disponibles, podría encaminarse hacia los Sistemas Tutores Inteligentes (STI), ya que permiten establecer una adecuada realimentación semántica así como un medio para poner en evidencia el conflicto cognitivo, usando como base para su desarrollo los modelos cognitivos

(Koedinger, 2001) y a la utilización de la robótica como herramienta de enseñanza ya que constituye una actividad altamente motivante (Fernandez Panadero y otros, 2010) y los estudiantes pueden autoevaluarse observando si el robot hace o no lo pedido en las consignas de trabajo.

Entre los STI desarrollados específicamente para la cursos de programación que ponen de manifiesto los diferentes modelos conceptuales que les sirven de base, pueden hallarse en (Moritz y otros, 2005) . Se describen las arquitecturas de los STI, en (Cataldi, Zulma, Salgueiro, Lage y García Martínez, 2010) y la edición de 2007 de IEEE Intelligent Systems dedicada a los STI describe un amplio panorama (Aroyo, Graesser y Johnson, 2007) .

De las herramientas orientadas al desarrollo de STI estudiadas, se han seleccionado aquellas que permitan elaborar tutores virtuales inteligentes para este proyecto si la necesidad de una programación de bajo nivel, ya que algunas están disponibles como servicios web y otras se pueden instalar en la plataforma Moodle. En los sistemas seleccionados como ASPIRE (Mitrovic, Martin y Suraweera, 2007) , SIETTE (Guzmán, Conejo y Pérez de la Cruz, 2007) , solo es necesario desarrollar los conocimientos del tutor en lenguajes de alto nivel y relacionar las evaluaciones con una adecuada realimentación.

Entre las herramientas disponibles para una enseñanza basada en la robótica se usarán el robot virtual RUR-PLE para Python del cual se dispone además un STI asociado (desJardins, Ciavolino, Deloatch y Feasley, 2011) y kits de LEGO Mindstorms (Blais, 2010) .

Hipótesis planteadas en el Proyecto

El bajo rendimiento académico de los estudiantes de Ingeniería en el aprendizaje de la programación de computadoras, se puede mejorar mediante el uso combinado de dos herramientas:

- Los Sistemas Tutores Inteligentes que detectan modelos mentales erróneos y que vía retroalimentación guían al estudiante en la construcción de modelos conceptuales correctos.
- La programación de robots por parte de los estudiantes para motivarlos en el proceso de aprendizaje en la acción concreta.

Resultados y Objetivos

Este trabajo tiene por objetivo general mejorar el rendimiento académico en cursos de programación de computadores de alumnos de Ingeniería, utilizando herramientas de la inteligencia computacional y de la robótica.

Los objetivos específicos consisten en:

1. Detectar modelos mentales entre los estudiantes de los cursos de programación en Ingeniería utilizando herramientas de inteligencia computacional.
2. Desarrollar modelos conceptuales para los Sistemas Tutores Inteligentes.
3. Implementar los Sistemas Tutores Inteligentes y evaluar su impacto en el rendimiento académico de los estudiantes.
4. Desarrollar, implementar y evaluar actividades prácticas basadas en la programación de robots en los cursos de programación en Ingeniería.
5. Evaluar la interacción entre las dos estrategias.
6. Formar Recursos Humanos para facilitar la incorporación de STI en otros cursos de Programación.

Resultados

Con el objeto de llevar adelante este proyecto, se realizarán las siguientes tareas y experimentos, orientadas al logro de los objetivos:

1. Determinación de modelos mentales de aprendizaje, entre los estudiantes de cursos de programación de Ingeniería, a partir de información disponible de las cursadas 2010-2011, usando redes neuronales como herramienta para detectar patrones.
2. Elaboración de la ontología de base para la enseñanza de la programación y producción de las reglas de restricciones para construir el mapa conceptual de la asignatura.
3. Puesta a punto y prueba del STI con estudiantes de 2013, validando los resultados obtenidos mediante la comparación con STI basado en teorías de prueba.
4. Desarrollo e implementación de actividades prácticas utilizando Python y C++, nativos para RUR-PLE y compatibles con LEGO. Aplicación de las actividades

prácticas a cursos de programación de las carreras de Ingeniería.

5. Análisis de los resultados de los diferentes experimentos mediante herramientas estadísticas que permitan analizar el impacto en el rendimiento de las diferentes estrategias por separado y también de su acción conjunta, para lo cual se diseñaran diferentes grupos de aplicación y control.
6. Diseño y desarrollo de seminarios, talleres y otras estrategias para la Formación de Recursos Humanos con el fin de extender la aplicabilidad de los resultados obtenidos a otros cursos de Programación de ésta y de otras Casas de Estudios.

Líneas de Investigación

La concreción de este proyecto permitirá:

- Abordar la difícil problemática de la falta de rendimiento de los estudiantes universitarios de primer año que cursan las carreras de ingeniería y en particular los que cursan la asignatura Informática.
- Evitar el abandono temprano de las carreras de ingeniería como consecuencia de las dificultades cognitivas experimentadas con el aprendizaje de la programación de computadoras.
- Disponer experiencia con las tecnologías de Sistemas Tutores Inteligentes como base para el desarrollo de nuevas herramientas mediante trabajos finales de la carrera de Ingeniería en Computación.
- Difundir el uso de los Sistemas Tutores Inteligentes en la comunidad educativa de las ingenierías como un modo de abordar la masividad, la falta de conocimientos básicos de los ingresantes y facilitar la inserción de la educación mediada por las TIC.
- Disponer de experiencia en la elaboración de actividades basadas en la programación de robots e iniciar el desarrollo de robots con inteligencia computacional como base de la aplicación de las nuevas tecnologías a las ingenierías.
- Difundir el uso y la experiencia en temas de robótica entre los estudiantes de las carreras de ingeniería.

Formación de recursos humanos

Si bien se espera incorporar becarios y trabajos finales a este proyecto, por su incipiencia actualmente sólo abarca a los integrantes que son docentes, investigadores y alumnos del Departamento

Computación de la FCEFYN. No obstante, es importante destacar su naturaleza formadora ya que tanto sus integrantes como otros docentes e investigadores y los alumnos de grado están contemplados en sus objetivos de formación, como se puede observar en la lectura de su plan de trabajo desarrollado a continuación.

Bibliografía

Aroyo, Lora, Graesser Arthur, y Johnson Lewis (2007): *Guest Editors' Introduction: Intelligent Educational Systems of the Present and Future*. IEEE Intelligent Systems, Vol. 22, No 4, Ago 2007.

Bartó, Carlos A. y Weber, Juan F. (2008): *El déficit en formación lógico-formal como factor de riesgo en el desempeño en informática*. Latin American and Caribbean Journal of Engineering Education. Vol. 2, No 1. 2008

Bartó, Carlos A. (2010): *Competencias Básicas necesarias para el desempeño en la materia de Informática*. IV JORNADAS DE INTERCAMBIO SOBRE ENSEÑANZA DE LA INGENIERÍA. Educación por competencias en los primeros años de las Carreras de Ingeniería-Facultad de Ciencias Exactas Físicas y Naturales – UNC -14 de diciembre de 2010.

Blais, B. S.(2010): *Using Python to Program LEGO MINDSTORMS® Robots: The PyNXC Project*, The Python Papers Vol. 5, Issue 2, ISSN:18343147. [Jarzabek:2007] Stanislaw Jarzabek; "Effective Software Maintenance and Evolution. A Reuse-Based Approach". Auerbach Publications. NY - EEUU - 2007.

Bornat, Richard; Dehnadi, Saeed y Hamilton, Simon(2008): *Mental models, Consistency, and Programming Aptitude*. Autralian Computer Society. ACE 2008.

Cataldi, Zulma; Salgueiro, Fernando; Lage, Fernando J. y García Martínez, Ramón (2010): *Sistemas Tutores Inteligentes: Los estilos del estudiante para selección de tutorado*. LIEMA Laboratorio de Informática Educativa y Medios Audiovisuales, LSI Laboratorio de Sistemas Inteligentes, Facultad de Ingeniería, UBA. Centro de Ingeniería de Software e Ingeniería del Conocimiento Escuela de Postgrado. ITBA.

Carretero, Mario y otros (2005): *Introducción a la psicología cognitiva*. Aiqué Grupo Editor. 2005.

Cebollada y Verdaguer, Marcelo; Eschoyez, Maximiliano y Bartó, Carlos (2010): *Educación Continua para Cursos Masivos: Un Software Basado en*

Web para Crear Preguntas para Moodle. IV JORNADAS DE INTERCAMBIO SOBRE ENSEÑANZA DE LA INGENIERÍA. Educación por competencias en los primeros años de las Carreras de Ingeniería-Facultad de Ciencias Exactas Físicas y Naturales – UNC -14 de diciembre de 2010.

Dehnadi, Saeed y Bornat, Richard. (2006): *The camel has two humps (working title)*. School of Computing, Middlesex University, UK.

desJardins, Marie; Ciavolino, Amy; Deloatch, Robert y Feasley, Eliana (2011): *Playing to Program: Towards an Intelligent Programming Tutor for RUR-PLE*. Association for the Advance of Artificial Intelligence.

Díaz, Laura, Algorry, Aldo y Natali Osvaldo (2011): *Las TIC'S en Educación Superior: Su Impacto en la Mejora de la Enseñanza para las Carreras de Ingeniería*. JORNADAS IRAM UNIVERSIDADES. Universidad Nacional de La Plata. Noviembre de 2011.

Díaz, Laura, Solinas Miguel y Eschoyez Maximiliano (2011): *Propuesta para facilitar la articulación entre escuela media y Universidad en las carreras de Ingeniería*. IV Encuentro Nacional y I Latinoamericano sobre Ingreso a la Universidad Pública. Univesidad Ncional del Centro. Mayo de 2011.

Fernández Panadero, Carmen y otros (2010): *Impact of Learning Experiences Using LEGO Mindstorms in Engineering Courses*, IEEE EDUCON 2010 Conference (2010), Pág.503-512, ISBN: 9781424465682

Guzmán, Eduardo; Conejo, Ricardo; y Pérez de la Cruz, José Luis (2007): *Improving Student Performance Using Self-Assessment Tests*. IEEE Intelligent Systems, Vol. 22, No 4, Ago 2007.

Johnson-Laird, Phillip N. (1993): *Mental Models, Deductive Reasoning, and the Brain*. Department of Psychoogy, University of Princeton. N. J. U.S.A.

Koedinger, Kenneth R. (2001): *Cognitive Tutors as Modeling Tools and Instructional Models*. Capítulo 5 en Smart Machines in Education, Kenneth D. Forbus y Paul J. Feltovich (eds.). AAAI Press / MIT Press. 2001.

LEY 24521. (1995): *Ley Nacional de educación Superior*. Boletín Oficial de la República Argentina No. 28.204.

Mitrovic, Antonija; Martin, Brent; y Suraweera, Pramuditha (2007): *Intelligent Tutors for All: The*

Constraint-Based Approach. IEEE Intelligent Systems, Vol. 22, No 4, Ago 2007.

Moritz, Sally H.; Wei, Fang; Parvez, Shahida M.; y Blank, Glenn D. (2005): *From Objects-First to Design-First with Multimedia and Intelligent Tutoring*. ITICSE 2005, junio 29. ACM.

Yadin, Aharon (2011): *Reducing the Dropout Rate in Introductory Programming Course*. ACM Inroads. Vol 2, No. 4, diciembre 2011.