

SISTEMAS INTELIGENTES APLICADOS A LA ENSEÑANZA DE LA PROGRAMACION EN INGENIERIA

Carlos Alberto Bartó, Laura Cecilia Díaz, Marcelo Cebollada y Verdaguer, Maximiliano Eschoyez y Nicanor Romero Venier

Laboratorio de Educación Virtual, Departamento de Computación. F. C. E. F. y N

Resumen

Este proyecto se desarrolla en la Facultad de Ciencias Exactas, Físicas y Naturales de la Universidad Nacional de Córdoba mediante subsidio de la Secretaría de Ciencia y Tecnología de esta Universidad en la convocatoria 2012-2013 de proyectos I+D e implementado en Laboratorio de Educación Virtual.

La aplicación de la psicología cognitiva para el aprendizaje de los principios fundamentales de la Programación de Computadoras, ha tenido dificultades para alcanzar el cambio conceptual deseado en los estudiantes de estos cursos (Carretero, 2005). Además, 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). 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 desalientan luego a continuar con los estudios (Bartó, 2010). Pero 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.

En los cursos de Introducción a la Programación 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) y en la programación de robots, se incorporaron evaluaciones permanentes sobre la comprensión de textos escritos en el lenguaje Python o C++ y la traducción de la solución de un problema dado como especificaciones en un castellano técnico al lenguaje de programación, habiéndose producido sólo algunas mejoras en los cursos de programación (Cebollada y Verdaguer, Eschoyez y Bartó, 2010) y también en (Díaz, Algorry y Natali, 2011).

Ante estos problemas se han seleccionado Sistemas Tutores Inteligentes (STI) que permitan su implementación sin necesidad de programación a bajo nivel. Algunos están disponibles como servicios web y otros se pueden instalar en la plataforma Moodle. En los sistemas seleccionados, ASPIRE (Mitrovic, Martin y Suraweera, 2007) y 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 retro alimentación. Para la enseñanza basada en respuestas visibles y concretas se usaron el robot virtual RUR-PLE para Python del cual se dispone además un STI asociado (des Jardins, Ciavolino, Deloatch y Feasley, 2011) y kits de LEGO Mindstorms (Blais, 2010).

Durante los dos cuatrimestres del ciclo lectivo de 2012 se ha realizado un estudio en profundidad de las evaluaciones conceptuales y se han seleccionado dos grupos de estudiantes para poder elegir un conjunto de preguntas a implementar luego en SIETTE durante 2013. También se ha estudiado el impacto en dichos grupos de las técnicas de programación de robots tanto virtuales como físicos.

Palabras Clave: Sistemas Tutores Inteligentes, Robótica, Programación, Modelos mentales.

Introducción

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.

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).

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, Eschoyez 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.

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.

La búsqueda de soluciones a los problemas planteados, atendiendo a los recursos disponibles, se ha encaminado 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, tanto física mecánica como las simulaciones virtuales.

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 algunos 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) se describen los conocimientos mediante ontologías a partir de las cuales el sistema contruye reglas de restricciones que verifican la validez de las expresiones y enunciados de las soluciones propuestas a los problemas, SIETTE (Guzmán, Conejo y Pérez de la Cruz, 2007) permite construir evaluaciones que se ajustan adecuadamente a modelos de cada estudiante en acuerdo a las actuales teorías de la evaluación como a su construcción adaptativa mediante modelos Bayesianos para guiar el proceso educativos, en el caso de tutores para lograr el aprendizaje en un entorno de interacción en lenguaje natural se ha seleccionado TQ-Bot cuyo objetivo general es mejorar el rendimiento académico en cursos de programación de computadores, utilizando herramientas de la Inteligencia Computacional y los objetivos específicos son desarrollar un conjunto de Redes Conceptuales para la enseñanza del lenguaje Python, y a partir de dichas Redes Conceptuales desarrollar un Tutor Inteligente implementado en AIML en la plataforma Moodle de la UBP y también implementar un ambiente de programación en Python para la web que también provee un sistema de depuración gráfico.

Entre las herramientas disponibles para una enseñanza basada en la robótica se usaron 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).

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.

Objetivos generales y específicos

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 en el ambiente de enseñanza basado en la plataforma Moodle del Laboratorio de Educación Virtual (LEV) de la FCEFyN de la UNC.

Los *objetivos específicos* consisten en:

1. Detectar modelos mentales entre los estudiantes de los cursos de programación en Ingeniería a partir de la interpretación de los resultados de respuesta corta de las Evaluaciones Conceptuales y la construcción de programas que sostengan la semántica procedimental de dichos modelos más frecuentes.
2. Desarrollar Modelos Conceptuales para los Sistemas Tutores Inteligentes a partir del conocimiento formal del lenguaje Python descrito en las publicaciones disponibles para el dictado de la asignatura Informática.
3. Implementar los Sistemas Tutores Inteligentes Siette, Aspire y TQ-Bot así como otras herramientas complementarias concurrentes y evaluar su impacto en el rendimiento académico de los estudiantes.
4. Desarrollar, implementar y evaluar actividades prácticas basadas en la programación de robots LEGO y RURPLE 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

Durante los dos cuatrimestres lectivos de 2011 se realizaron las Evaluaciones Conceptuales (EC) habituales de la materia Informática para estudiantes de varias carreras Ingeniería donde se ha implementado el curso en la programación en el lenguaje Python y que consisten en siete instancias de evaluación a lo largo del cuatrimestre constituidas por un total de cuarenta ítems con respuesta corta en forma automatizada a partir de un banco de 25000 preguntas implementadas en la plataforma Moodle del LEV. El objetivo de las mismas consiste en poner en funcionamiento la capacidad para comprender el texto en un lenguaje de programación, en forma

equivalente a la comprensión de una lengua extranjera, pero de carácter formal y en particular el significado procedimental ya que las preguntas están dirigidas a obtener los resultados de la ejecución los mismos. Del total de estudiantes se tomó un 30% para suministrarles ítems seleccionados que luego serán implementados en los tutores inteligentes a partir del descubrimiento de sus modelos mentales, pero sin recurrir a la aleatoriedad de los bancos de preguntas de cada concepto.

Estas evaluaciones han recibido dos tipos de estudios para analizar la pertinencia de las mismas con el fin de ser usados para la construcción de los modelos mentales que emplearon los estudiantes para realizarlos. Uno de tipo cuantitativo que está orientado a la validez estadística como muestra de la totalidad del universo de preguntas aleatorias que constituyen, por razones de seguridad, un espacio muy amplio y que necesita compararse con la muestra elegida por razones que se estudian en (Díaz y otros, 2013) y el presente de carácter cualitativo que permite interpretar los razonamientos llevados a cabo por los alumnos para dar las respuestas a los diferentes ítems.

Utilizando las facilidades de la plataforma se han analizado los resultados de la totalidad de las preguntas en términos de frecuencia de resultados comunes y a partir de los mismos se han interpretado los mismos de tal manera de hallar un significado de los resultados erróneos y luego la elaboración de su semántica procedimental ya que es la única alternativa de acuerdo a los conceptos de la psicología cognitiva que permite dar una respuesta consistente a dichos modelos.

A continuación se presenta una salida típica del análisis que realiza Moodle para una pregunta de respuesta corta según Moodle Quiz Reports (Moodle, 2012).

A partir de estas salidas se muestran las interpretaciones verbales de las respuestas erróneas hasta un nivel de participación del 2% del total de las respuestas a las siete EC (ver APENDICE 1) ya que luego las mismas serán usadas para confeccionar preguntas con respuestas de opción múltiple a ser incluidas en el sistema de evaluación del sistema Siette, cuyo análisis se realiza en (Marangunic, 2013).

Informática - Primer Semestre - 2012

LEV ► INF-1-12 ► Cuestionarios ► Evaluacion Conceptual 3 (Python).

Actualizar Cuestionario

[Información](#)
[Resultados](#)
[Vista previa](#)
[Editar](#)
[Visión general](#)
[Recalificar](#)
[Calificación manual](#)
[Análisis de ítems](#)

Ver todas las calificaciones del curso

Grupos separados:

Tabla de análisis de ítems ?

Pr.#	Texto de la pregunta	Texto de la respuesta	crédito parcial	Nº R.	%R.	% Facil. Correct.	DT	Índice Disc.	Coe Disc.
(127526)	<p>es1_5,211 2010 -Barto :</p> <p>Indicar el resultado de la ejecución del siguiente programa:</p> <pre> u = 5 v = 5 s = 1 t = 3 while v < t : v+= s u-= v print u </pre>	5	(1,00)	135/163	(83%)	83%	0,378	0,93	0,76

Se pueden hacer una variedad de análisis de los resultados e interpretación de los errores en los mismos ya que la información provista tiene considerables dimensiones:

- IF: Índice de Facilidad o relación entre el promedio de las notas obtenidas y el máximo posible. $IF = X_{promedio} / X_{max}$
- %R: Porcentaje que eligió una respuesta dada.
- REA: Número de respuestas cortas erróneas alternativas (no provisto en Mood.).
- ID: Índice de Discriminación, que es un indicador de performance que para separar el promedio del tercio de las notas de aprendices fuertes del promedio del tercio de los aprendices débiles. $ID = (X_{sup} - X_{inf}) / N$ con resultado en el rango [-1 descartables para < 0,+1]
- CD: Coeficiente de Discriminación es un factor de correlación entre la respuesta del ítem (pregunta) y el del test EC. Permite discriminar las altas performances de las bajas para la totalidad y no por tercios de la población. $CD = \text{Sum}(xy) / (N * s_x * s_y)$ en [-1 descartables para < 0,+1]
- Descripción verbal del error, elaborado a partir de la ejecución de un modelo descripto por la semántica procedimental de un programa que da dicha respuesta.

Análisis de resultados

De la TABLA DE RESULTADOS es fácil observar algunas características que pueden relacionarse con la semántica de los enunciados planteados:

TABLA DE RESULTADOS (Moodle)

EC	Item	IF%	REA	ID	CD	Tipo
1	1	70	20	1	0.75	rc
	2	43	39	1	0.75	rc
2	1	1	119	0.03	0.25	rc
	2	54	3	0.22	0.73	om
	3	57	13	0.20	0.75	rc
3	1	83	9	0.93	0.70	rc
	2	85	13	0.87	0.60	om
	3	47	3	0.48	0.62	rc
4	1	12	54	0.04	0.68	rc
	2	27	92	-0.06	0.80	rc
	3	11	110	0.08	0.76	rc
5	1	38	39	0.50	0.74	rc
	2	69	29	0.26	0.72	rc
	3	29	73	0.45	0.73	rc
6	1	57	29	0.43	0.80	rc
	2	36	44	0.63	0.88	rc
	3	43	56	0.51	0.81	rc
7	1	59	27	0.82	0.74	rc
	2	43	10	0.53	0.64	rc
	3	77	17	0.71	0.66	rc

- a) Los mayores valores altos del número de Respuestas Erróneas Alternativas [42,119] como las EC2-1(3), la EC4-2(10) y la EC4-3(11) se corresponden con bajos Índices de Facilidad, o sea que son preguntas difíciles y también sus Índices de Discriminación por tercios son bajos no así sus Coeficientes de Discriminación. El tema de la pregunta EC2-1 es *Salida de información con formato*, el de la EC4-2 es *Cadenas, acceso a caracteres con índice* y el de la EC4-3 es *Cadenas, acceso a caracteres con operador de corte*. Es decir que las preguntas vinculadas con caracteres o con cadenas de caracteres generan el mayor número de modelos alternativos al no haber una clara comprensión de la relación entre los datos y la forma de mostrarlos luego en pantalla.
- b) Los valores medio-altos del número de Respuestas Erróneas Alternativas [54,73] se corresponden con las preguntas EC4-1(9), EC5-3(14) y EC6-3(17) y sus Índices de Facilidad también son bastante bajos por lo que se las puede

considerar como preguntas difíciles. En cuanto a los Índices de Discriminación por tercios son valores medios salvo en un caso y los Coeficientes de Discriminación son valores altos. El tema de la pregunta EC4-1 es *Cadenas, de números a cadenas*, el de la EC5-3 es *Arreglos bidimensionales*, el de la EC6-3 es *Definición de funciones*. Es decir que no hay una característica de dificultad específica del tema, salvo en el primero que se mantiene el tema de cadenas y en los otros las estructuras secuenciales o arreglos. Otra característica que justifica la dispersión de errores es el anidamiento de las estructuras de control lo cual puede producir un producto cartesiano de errores muy considerable.

- c) Los valores medio-bajos del número de Respuestas Erróneas Alternativas [27,44] se corresponden con las preguntas EC1-2(2), EC5-1(12), EC5-2(13), EC6-1(15), EC6-2(16) y EC7-1(18) y sus Índices de Facilidad oscilan entre 40% y 60% por lo que pueden considerarse preguntas de dificultad aceptable. El tema de la pregunta EC1-2 es *Constantes y operaciones aritméticas*, el de la EC5-1 es *Arreglos unidimensionales. Operador de corte*, el de la EC5-2 es el *Arreglos unidimensionales. Añadido y supresión de elementos*, el de la EC6-1 es *Definición de funciones, arreglos como argumentos y alcance de variables*, el de la EC6-2 es *Definición de funciones, arreglos como argumentos y alcance de variables*, y el de la EC7-1 es *Definición de funciones, llamado anidado de funciones*. En este caso hay un claro predominio del concepto de funciones, operadores y expresiones funcionales que muestran claramente un mayor conocimiento previo y por ende una mayor comprensión de los conceptos.
- d) Los valores bajos del número de Respuestas Erróneas Alternativas [3,20] se corresponden con las preguntas EC1-1(1), EC2-2(4), EC2-3(5), EC3-1(6), EC3-2(7), EC3-3(8), EC7-2(19) y EC7-3(20) y sus Índices de Facilidad oscilan entre 50% y 85% lo que se corresponde con preguntas que se pueden considerar como fáciles. El tema de la pregunta EC1-1 es *Herramienta de programación – Ensamblador*, el de la EC2-2 es *Entrada y salida gráfica (opción múltiple)*, el de la EC2-3 es *La estructura de decisión doble*, el de la EC3-1 es *Las estructuras de repetición indefinidas*, el de la EC3-2 es *Las estructuras de repetición definidas*, el de la EC3-3 es *Las estructuras de repetición definidas. Aplicaciones gráficas (opción múltiple)*, el de la EC7-2 es *Recursión*, y el de la EC7-3 es *Definición de funciones, arreglos como parámetros*. Podría decirse que los

conceptos de control en la toma de decisiones son relativamente fáciles de aprender y su vinculación con la lógica es evidente.

Del análisis de resultados intrínsecos de las preguntas realizadas en este trabajo y el de su relación entre el universo de respuestas a preguntas aleatorias de similares características realizado en trabajo ya citado (Díaz, 2013) se ha llegado a la necesidad de reelaborar el banco de preguntas con criterios que eviten las preguntas inconvenientes tanto por su exceso o déficit de dificultad intrínsecas como por su representatividad frente al universo completo de respuestas frente a preguntas con una variabilidad acotada.

Desarrollo de un nuevo Banco de Preguntas

Del anterior análisis han surgido los siguientes criterios para la generación de un nuevo Banco de Preguntas de opción múltiple que permita su implementación en el Sistema Inteligente Siette:

- a) Todas las preguntas deben evaluar un solo tema del programa y deben evitar en la medida de lo posible generar un espacio de posibles modelos erróneos indefinido.
- b) Todas las preguntas serán de tipo opción múltiple, tanto porque permiten un análisis estadístico coherente como su comparación entre las poblaciones de los cursantes en el sistema Moodle y los del sistema Siette, como una manera de evitar una exagerada dispersión de modelos mentales erróneos.
- c) Extremar el diseño de los enunciados de las preguntas vinculadas con caracteres y cadenas de caracteres y limitar el alcance de las mismas de acuerdo a lo indicado en a) y b).
- d) Desarrollar para cada pregunta una opción múltiple de cuatro respuestas erróneas a partir de los errores más frecuentes encontrados o nuevos en el caso de necesitar nuevas preguntas más simples, a partir de una interpretación del modelo mental basado en la semántica procedimental de un programa Python que la interprete y de las consideraciones de los principios de parsimonia y de los mecanismos de construcción de modelos en particular la consideración de la influencia semántica sobre problemas exclusivamente sintácticos como los de los lenguajes informáticos (Johnson-Laird, 2005). Estas preguntas vinculadas al

tema de Funciones se han incluido junto con sus respectivos modelos procedimentales en lenguaje Python.

- e) Desarrollo de un banco de preguntas similares a partir de modelos de base mediante el sistema WGen (Cebollada y Verdaguer, 2013).

Conclusiones

La primera y más significativa de las conclusiones es que el hecho de investigar la propia práctica docente pone de manifiesto que las dificultades en las evaluaciones, predominantemente atribuida a la falta de conocimiento de los aprendices, bien pueden estar influidas por criterios de evaluación que crean las mismas dificultades y por otra parte que la aplicación de la teoría de modelos mentales de la psicología cognitiva, no solo puede ser útil para producir diagnósticos predictivos como lo indican (Dehnadi y Bornat, 2006) sino que también se pueden construir modelos dichos modelo mediante la semántica procedimental implementada en programas informáticos.

La calidad de las evaluaciones, tanto en la descripción de los ejercicios y problemas planteados como su semántica y más aun la consideración de modelos del aprendiz basados en la Teoría de Respuesta al Ítem y el aprendizaje Bayesiano hacen conveniente el uso de herramientas de sistemas inteligentes, susceptibles de ser usados en ambientes web como el sistema Siette ya que permiten garantizar la mayoría de las objeciones a los sistemas de evaluación.

Por otra parte es necesario utilizar otras herramientas de tutoría y motivación en línea para brindar un adecuado asesoramiento en lenguaje natural, como TQ-Bot y como guía de los aprendizajes y finalmente la aplicación de aprendizaje basado en problemas con herramientas de retroalimentación y evaluación inteligente como Aspire y de carácter concreto como RUR-PLE, que permitirán una confluencia de soluciones para garantizar los aprendizajes y las evaluaciones en ambientes web que supongan hacer realidad una educación mediada de la programación de computadoras a salvo del estigma de que *los camellos tienen dos jorobas* (Dehnadi y Bornat, 2006).

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.

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

Blais, B. S.(2010): *Using Python to Program LEGO MINDSTORMS® Robots: The PyNXC Project*, The Python Papers Vol. 5, Issue 2, ISSN:18343147.

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.

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.

Cebollada y Verdaguer, Marcelo (2013): *Generador de ejercicios sobre WEB para la enseñanza de la programación, basados en evaluación objetiva y respuesta corta para el LMS Moodle*. I Jornadas Nacionales y III Jornadas de Experiencias e Investigación en Educación a Distancia y Tecnología Educativa. Marzo 2013. Universidad Nacional de Córdoba.

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

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, Marangunic, Rodolfo y Bartó, Carlos Alberto (2013): *Hacia la Detección de los Modelos Mentales de los Estudiantes de Programación*. I Jornadas Nacionales y III Jornadas de Experiencias e Investigación en Educación a Distancia y Tecnología Educativa. Marzo 2013. Universidad Nacional de Córdoba.

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

Johnson-Laird, P. N. (2005): *Mental Models, Sentential Reasoning, and Illusory Inferences*. en *Mental Models in Cognitive Psychology, Neurosciences and Philosophy of Mind*. Carsten Held, Mares Knauff, Gottfried Vosgerau. 2005 Elsevier B. V.

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.

Marangunic, Rodolfo, Bartó, Carlos y Díaz, Laura (2013): *Análisis estadístico del modelo Bayesiano de SIETTE*. I Jornadas Nacionales y III Jornadas de Experiencias e Investigación en Educación a Distancia y Tecnología Educativa. Marzo 2013. Universidad Nacional de Córdoba.

Moodle (2012): *Quiz Reports*. Moodle Documents.

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.

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.

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.

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.