

# ACTAS

DE LAS I JORNADAS NACIONALES III  
JORNADAS DE LA UNC: EXPERIENCIAS E  
INVESTIGACIÓN EN EDUCACIÓN  
A DISTANCIA Y TECNOLOGÍA EDUCATIVA

14 Y 15 DE MARZO DE 2013



Universidad  
Nacional de  
Córdoba



Programa de  
Educación a  
Distancia



Secretaría de  
Asuntos  
Académicos

# Generador de ejercicios sobre web para la enseñanza de la Programación, basados en evaluación objetiva y respuesta corta para el LMS Moodle

Marcelo Cebollada y Verdaguer

---

## Resumen

La masividad de la Universidad Pública genera algunos problemas relacionadas a las cursadas de materias comunes, donde la cantidad de alumnos hace poco sostenible un modelo lo suficientemente denso de seguimiento y evaluación.

Los problemas inminentes son:

1. ¿como disponer de un grupo de ejercicios lo suficientemente grande para evitar la copia durante la ejecución de la evaluación, la memorización de los ejercicios o pasar los resultados de una comisión a otra?.
2. ¿como establecer para la evaluación objetiva alternativas que se correspondan con modelos mentales erróneos reales y verosímiles?.
3. ¿como regularizar la retroalimentación de los errores?.

Es evidente que sin el uso de un LMS y en nuestro caso particular Moodle, hace poco viable un modelo formado por siete evaluaciones conceptuales, con no menos de 20 ítem curriculares evaluados. Esto se llevó a cabo con herramientas "ad hoc" para la resolución de respuestas cortas.

Este trabajo se centra en el desarrollo de un contexto web para generar ejercicios para Moodle utilizando el formato Moodle XML para dos tipos de preguntas, las de respuesta corta y las de múltiple opción.

El contexto permite manejar variable definidas "al vuelo" de manera que el usuario puede crear sus propios modelos de generación, con funciones básicas.

## Palabras claves

Evaluación objetiva, LMS, Moodle, rendimiento académico, enseñanza de la programación.

---

## Introducción

La enseñanza de la programación contiene aristas difíciles de resolver en el proceso de aprendizaje y uno de ellos es la acreditación de conocimiento. En esta parte del proceso hay que decidir que se evalúa, como se evalúa y con que se evalúa. El elemento conceptual unitario lo denominaremos "ítem" (Guzmán E., Conejo R., y Pérez de la Cruz J. L. 2007).

En su momento, en discusiones de Cátedra, se tomo la decisión de evaluar al principio 39 ítem pero la logística de implementación nos obligo rápidamente ir a 20. En la instancia de preparar

el material de las evaluaciones, ¿cuanto material era necesario y de que formato?, estos interrogantes nos llevo a dos decisiones: la primera en el uso de los "Quiz" de Moodle, donde el tipo de pregunta se eligió por facilidad constructiva la respuesta corta, la segunda la cantidad de ejercicios, que comenzó con cantidades para un mismo ítem de evaluación de alrededor de 400 ejercicios similares pero diferentes para las primeras evaluaciones y de alrededor de 200 ejercicios para las ultimas evaluaciones. En los primeros intentos generamos mas de 20.000 ejercicios, y la tarea era tediosa sobre todo hacer programas "ad hoc" para generar los ejercicios, donde la tarea consiste en como hacer el programa para generar, esto traslada la tarea principal al como construir las preguntas y no a las preguntas, ya que cada una de las preguntas requiere un programa a la medida de generación. Esto nos condujo a la construcción de un prototipo inicial en una aplicación "standalone", para separar el proceso de generación del ejercicio propiamente dicho, donde el esfuerzo sea el diseño del "ítem" a medir. Esta meta se llevo a cabo orientado al tipo de preguntas denominadas en Moodle de respuesta corta. Con esta aplicación se construyeron las preguntas para los alumnos de C++, en plazos realmente cortos cubriendo los aspectos de orientarse al "ítem" (Cebollada y Verdaguer M., Eschoyez M. y Bartó C. 2010).

Al avanzar conceptualmente y profundizar en temas de evaluaciones orientadas al "ítem", la respuesta corta es solo un modelo que permite ver el universo de respuestas que perciben como verdaderas los alumnos, pero la diversidad del conjunto de las respuestas no permite establecer todos los patrones de respuestas, pero sin embargo si estadísticamente aquellas que repetidas veces percibe el estudiante como verdaderas, mas allá de la verdad o no que expresen las respuestas. Estos modelos que percibe el estudiante son modelos mentales (Johnson-Laird, Phillip N. 1993) y (Bornat R., Dehnadi S., Hamilton S. 2008).

Las dificultades de mantener un ambiente actualizado de la distribución del software, y dado que los usuarios iniciales deben poder utilizar el software mas allá de los limites de la institución en su ultima versión se decidio que la versión de producción del software se implementa como un Sistema WEB. Estas decisiones implican cambios en el diseño original, con problemas vinculados a partes de la ejecución de los procesos que se realizan sobre la maquina que presta el servicio web (Cebollada y Verdaguer M., Eschoyez M. y Bartó C. 2011).

## **Desarrollo**

El diseño interno de los módulos de la aplicación para atender los requisitos de "respuesta corta" que fue la primera opción desarrollada condujeron a un modelo determinado, que con un rediseño adecuado extendimos para cumplir los requisitos de "opción múltiple".

Esta condición se establece como respuesta a condicionar el numero de respuestas probables basándose a posibles modelos constructivos de los alumnos, no se pretende la exploración universal del espacio de respuestas si no a un numero suficientemente razonable que en nuestro caso fijamos en 5 respuestas probables entre verdadera y erróneas El software no provee el modelo de la respuesta errónea, si no que provee el vehículo para construir un ítem de opción múltiple en el formato "Moodle XML", los modelos erróneos debe construirlos el usuario. Posiblemente se preguntaran ¿entonces donde está el beneficio?, no hay que minimizar la complejidad de interactuar con el generador de preguntas provisto por Moodle y ver la posibilidad de generar 100 preguntas similares para utilizarlas en una población de 600 estudiantes, para que la probabilidad de preguntas iguales en maquinas contiguas no sean probables, y además considere la evaluación de una cantidad razonable de ítem 20 o 25

implicaría de 2000 a 2500 preguntas. De mediciones de tiempo realizadas solo en la construcción de la pregunta de opción múltiple demanda la escritura alrededor de 5 minutos y si partimos de un trabajo organizado la variación de valores calculados en una planilla de calculo accesoria para los 100 casos del ítem y utilizando la posibilidad de grabar como pregunta nueva la pregunta editada esto demandaría por cada variación de 2 a 3 minutos las cuentas son sencillas preparar una evaluación de 3 ítem involucraría solo en la construcción 15 hs de trabajo. Como las variantes se cargaron manualmente, esto requiere un proceso de verificación por la posible introducción errónea de valores, que implica un valor de tiempo casi similar al empleado al cargarlo. Resumiendo cargar un ítem con 100 variaciones consume aproximadamente 10 hs de trabajo. En esta representación se ha dejado de lado el diseño del ítem a evaluar que es un costo independiente de como se construya la pregunta. El generador tiene el costo que implica de la escritura original o sea en promedio 5 minutos un par de minutos para establecer las listas de variación, el tiempo que demora el servidor en calcular los resultados y construir el Moodle XML y ya tenemos el archivo para la importación de conjunto de preguntas para el ítem La verificación a diferencia del proceso manual es solo muestreo, por que cada variación ha sido validada por software lo cual elimina errores accidentales de escritura, pero no elimina los errores de maquetación del ítem

Algunas de las partes del software los mostramos a continuación, aclarando las utilidades que provee o la función del mismo.

La Ilustración 1 nos muestra el esquema de trabajo a continuación describiremos el contenido citando los rótulos asociados:

- Crear Categoría Esto permite seleccionar si quiere o no crear una categoría asociada al grupo de preguntas
- Opción Múltiple Solo es un indicador del comportamiento del proceso (respuesta corta / opción múltiple).

*Ilustración 1: contexto ítem en python*

- Categoría Línea que permite establecer el nombre de la categoría bajo la que se van a encontrar el conjunto de ejercicios. Propone una plantilla de identificación.

- Cantidad. Para los casos de Opción Múltiple la cantidad total de opciones.
- Problema. Espacio donde se establece el enunciado sobre el que se pregunta, este texto soporta parámetros de reemplazo. El formato de contenido es libre.
- Programa. Este espacio se inicializa con una plantilla de ayuda, donde va el cuerpo del programa donde se calculan los elementos a mostrar en el formato que especifica la plantilla. Soporta parámetros de reemplazo. Actualmente proveemos dos lenguajes de programación Python y C++.
- Retorno. A igual que Programa se inicializa con una plantilla de ayuda para construir el retorno que va a proveer el Moodle a los alumnos. Soporta parámetros de reemplazo.
- Enunciado. Visualizamos el texto en xml de las preguntas resueltas, listo para ser incorporado a Moodle
- Nombre. Denominación que va a tener cada pregunta, al final se le agrega un número de orden.
- Archivo a Guardar. Denominación sin extensión del archivo que se va a manipular
- Parámetros Definición de las cadenas que identifican los parámetros de reemplazo.

```

FB Gral.
#
FB bien#
FB mal 1#
FB mal 2#
FB mal 3#
FB mal 4

```

*Ilustración 2: Retorno opción múltiple*

```

FB Gral.
#
FB mal|

```

*Ilustración 3: Retorno Respuesta corta*

De manera más específica analicemos la Ilustración 2 y 3. En ella observamos a la izquierda la plantilla que nos ofrece la herramienta para opción múltiple y a la derecha para respuesta corta. El rotulo “FB Gral.” se refiere a la descripción genérica que Moodle solicita para cada pregunta en general. De igual manera se espera “FB bien” que es el retorno por la respuesta correcta el resto es por cada una de las respuestas incorrectas respetando el orden de generación en el programa el carácter “#” lo usamos como referencia de separador. Cada una de las respuestas erróneas corresponde a la construcción de un modelo “verdadero” por parte del alumno, por lo que debe esclarecer la naturaleza del error que ha cometido el alumno, para que tenga el alumno la posibilidad de revisar de manera posterior el porque de su error. Esto permite que el alumno aprenda hasta de las evaluaciones, y que las mismas no solo tengan carácter de acreditación. En la Ilustración 4 y 5 podemos observar con mayor detalle la plantilla propuesta en programa. Esta plantilla es compatible con el lenguaje Python, y los textos entrecomillados con triple comilla simple o doble son tomados como comentarios y no tienen valor ejecutable.

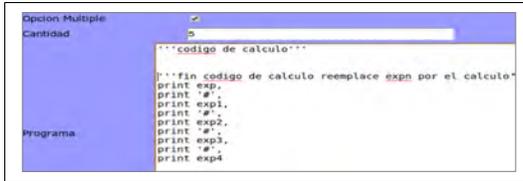


Ilustración 4: Programa Opción Múltiple

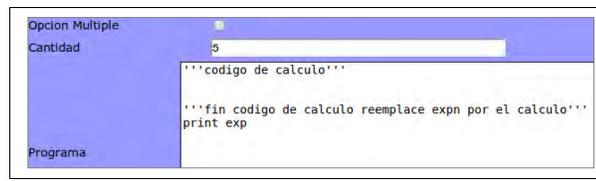


Ilustración 5: Programa Respuesta corta

Entre estos dos comentarios se debe escribir el código de programa necesario, aunque ello no es obligatorio. En la segunda línea de comentario aparece una leyenda explicativa “reemplace expn por el calculo” esto simplemente esta explayando que la variable especificada como exp, exp1, exp2, expn debe ser reemplazada cada una de ellas por una variable calculada o una expresión que reproduzca en el caso exp el valor realmente correcto y para el resto el valor para el modelo erróneo que se ha determinado para cada una de las opciones especificadas en el mismo orden en el cuadro de Retorno. La expresión “print '#,’” debe respetarse ya que la misma genera los separadores entre las respuesta que actúan como referencia de separador.

La herramienta provee algunas otras características que son accesorias a toda la tarea de generación, como se puede observar en la Ilustración 6 que corresponde a la botonera que contiene las acciones que se pueden desarrollar sobre este proceso.

Botón “Copio a Nombre”, toma el contenido del campo “Categoría” y lo copia a los campos “Nombre” y “Archivo a guardar”

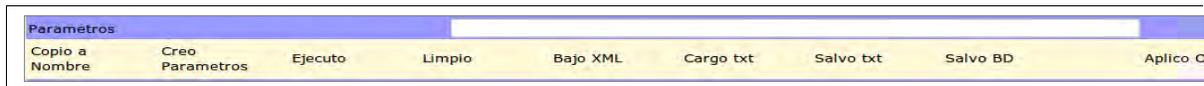


Ilustración 6: Botonera

Botón “Creo Parámetros”, toma el contenido del campo “Parámetros” que es una secuencia de nombres con la referencia de separador “#” y genera dinámicamente campos con los rótulos especificados como se observa en la Ilustración 7.

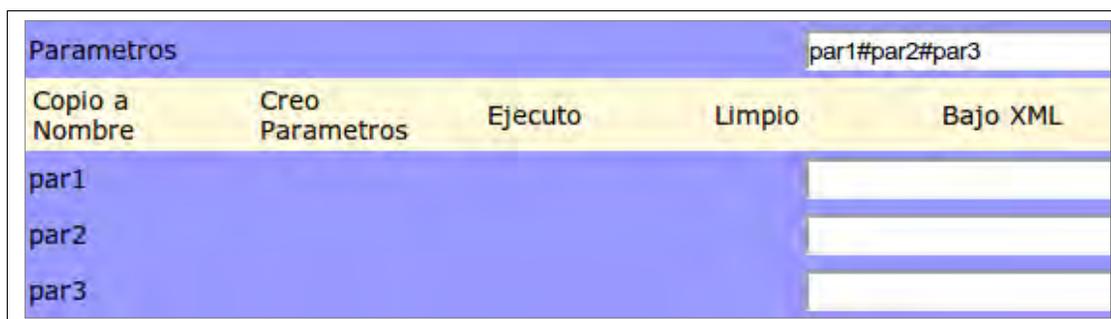


Ilustración 7: Parámetros

Los parámetros par1, par2, etc. deben representar nombres que no interfieran con el texto por lo que podría verse modificado accidentalmente. Estos nombres se utilizaran como partes de sustitución en el Problema, en el Programa y en el Retorno.



par1	Ana#Luis#Jorge#Maria
par2	15#7#22#31#75
par3	3.2#8.9#5.0#7.6#1.63

*Ilustración 8: Lista parámetros impactados*

En la Ilustración 8 vemos un ej. Rellenado, esto nos permite explicar la cantidad de ejercicios que se generan que es el resultado del producto cartesiano de la cantidad de elementos que contiene cada parámetro en este caso es  $4 \times 5 \times 5$  o sea 100 ejercicios por la combinación de los valores de los parámetros

Botón "Ejecuto", cuando se han completado los campos Problema, Programa, Retorno, Parámetros y los parámetros definidos, que son las precondiciones. Al oprimir el botón "Ejecuto" del lado del servidor se crea un programa fuente con las combinaciones de parámetros que correspondan, se ejecuta y se captura la salida. Esta salida junto con el problema y el retorno impactado con los parámetros se procesa y se construye la instancia del enunciado, repitiendo este mecanismo se construye la totalidad de los ejercicios completando el producto cartesiano antes mencionado. Cuando finaliza el proceso se muestra en el campo enunciado.

Botón "Limpio", la acción de este botón esta enfocado en inicializar los contenidos en el Problema, en el Programa, en el Retorno y en los Parámetros, Asimismo elimina los campos creados para los parámetros específicos dejando la herramienta para un nuevo ítem

Botón "Bajo XML", permite descargar como archivo el Enunciado, con el nombre provisto por Archivo a guardar al que le agrega automáticamente la extensión (.xml). Como realiza la descarga depende únicamente de como tiene configurado su navegador.

Botón "Cargo Txt", permite cargar archivos planos (.txt), reproduciendo un ejercicio que previamente ha sido guardado

Botón "Salvo Txt", permite descargar como archivo plano los contenidos de los campos en un formato especificado, con el nombre provisto por Archivo a guardar al que le agrega automáticamente la extensión (.txt). Como realiza la descarga depende únicamente de como tiene configurado su navegador.

Botón "Salvo BD", similar a "Salvo Txt" solo que produce soporte en una Base de datos y no en archivos planos. Para ello utiliza información del usuario en ese instante.

Botón "Aplico OM", su acción consiste en inicializar Programa y Retorno según corresponda con las plantillas correspondientes.

Todo lo desarrollado hasta este punto ha sido para la utilización del lenguaje Python, a continuación observa-remos las diferencias con respecto al lenguaje C++ y a como se observa el mismo.

La Ilustración 9 nos muestra de manera inicializada "Programa", esta es realmente la parte sensible y se puede hacer extensible a este punto todo lo expresado con referencia a la Ilustración 4 salvando las distancias de los equivalentes del lenguaje como por ej bloques de comentarios encerrados `/* comentario */` que pueden ser multiple lineas.

```

Opcion Multiple 
Cantidad 5
Programa
/*codigo de calculo*/
#include <iostream>
using namespace std;

int main(){

/*fin codigo de calculo reemplace expn por el calculo*/
cout << exp
<< '#' << exp1
<< '#' << exp2
<< '#' << exp3
<< '#' << exp4
<<<endl;

return (0);
}
    
```

Ilustración 9: Programa C++

Un ej basado en un ítem real de evaluación de la asignatura Informática realizada sobre lenguaje Python.

Py

Creo Categoría  Categoría Concepto 2.2 Tipos de Datos Enteros y Flotantes\_

Opcion Multiple  Cantidad 5

Problema

Que muestra en la consola el interprete Python al ejecutar la siguiente expresion:

```
>>> @16@/@2**@2e+@5@
```

Programa

```
'''codigo de calculo'''
'''fin codigo de calculo reemplac
print @16@/@2**@2e+@5@,
print '#',
print (@16@./@2@.)**@2e+@5@,
print '#',
print @16@/@2**@2e+@5@,
print '#',
print (@2**@2e@)/@16@+@5@,
print '#',
print (@16@/@2@)**(@2e+@5@)
```

Retorno

FB Gral.  
#  
Correcto !!!#  
Mal realizo la division antes que la potencia#  
Mal realizo un cociente flotante y no entero#  
Mal invirtio el orden del dividendo y el divisor#  
Mal realizo la division y la suma antes que la potencia

Enunciado

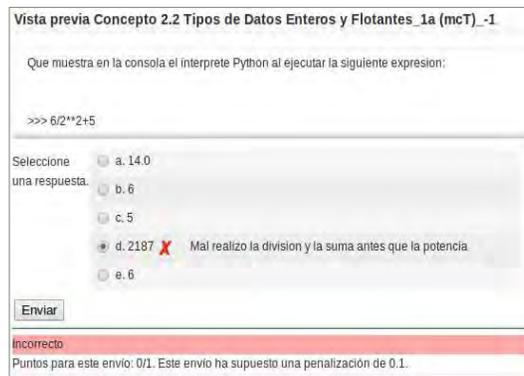
Nombre Concepto 2.2 Tipos de Datos Enteros y Flotantes\_ Archivo a guardar Concepto 2.2 Tipos de D

Parametros @16#@#@2#@#@2e#@#@5@

Copio a Nombre	Creo Parametros	Ejecuto	Limpio	Bajo XML	Cargo txt	Salvo txt	Salvo BD	Aplico
Parametros								100%
@16@				6#7#8#9				
@2@				2#3#4				
@2e@				2#3#4				
@5@				5#6#7				

Ilustración 10: ejemplo sobre la consola Python

El ej. De la Ilustración 10, se construyo a partir de una pregunta de respuesta corta idéntica sobre la que se pudo estudiar el universo de respuestas, y en este lugar se tomaron las erróneas mas frecuentes para los modelos. Esta configuración de parámetros construye 108 ejercicios distintos.



*Ilustración 11: Vista previa*

En la Ilustración 11 se puede observar la incorporación al banco de preguntas a partir de la importación que incluye la categoría También se observa la cantidad de preguntas incorporadas en 108 como reconoce la categoría entre paréntesis

La visualización de una pregunta se observa en la Ilustración 12 donde se ha seleccionado la opción y se ha enviado, se puede ver al lado el retorno que produce Moodle a partir de los contenidos realizados en la pantalla de captura.



*Ilustración 12: Preguntas importadas en Moodle*

Extendiendo el uso del software se pueden construir ejercicios relacionados a la física, álgebra, análisis matemático, etc., como se puede mostrar en el ejemplo de la Ilustración 13, donde se aborda el producto escalar entre vectores, el ej es casi trivial, pero se puede abordar los casos de aplicación de propiedades de los vectores como asociatividad por ej. Como así de igual manera del ej de 2 dimensiones llevarlo a 3 o n dimensiones.

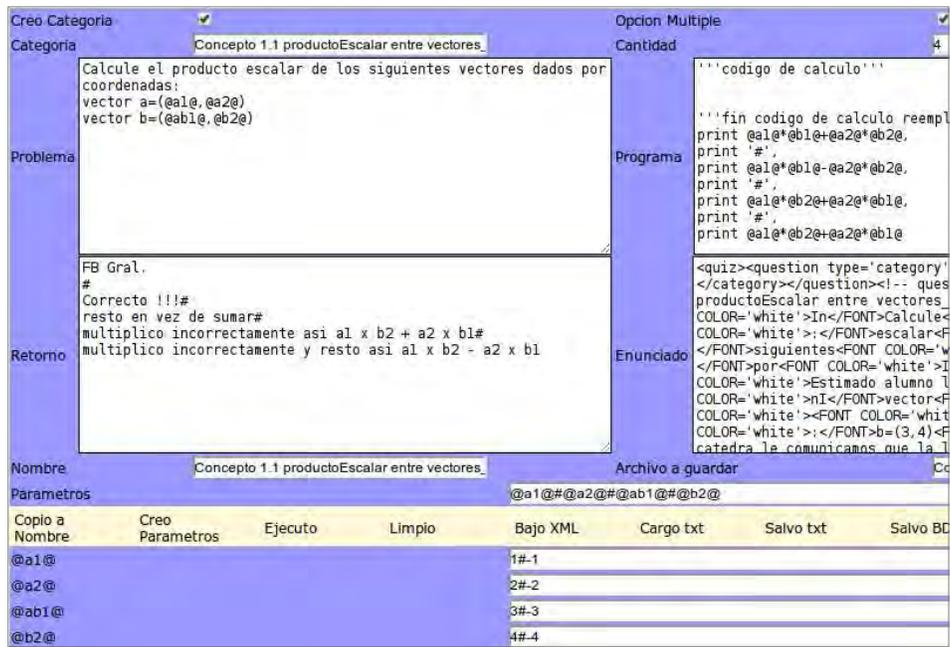


Ilustración 13: ej. Producto escalar de Vectores

Las ilustraciones 14 y 15 muestran respectivamente el banco de preguntas y la vista previa del ej “ad hoc” realizado de un tema de álgebra y observamos de esta manera la propiedad extensible de la herramienta.

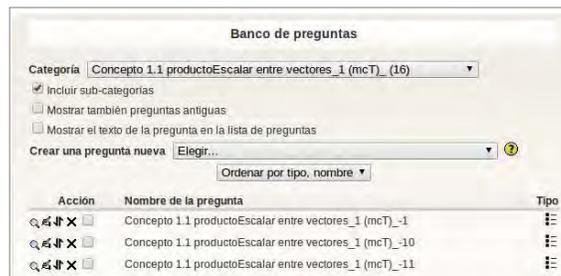


Ilustración 14: banco de preguntas

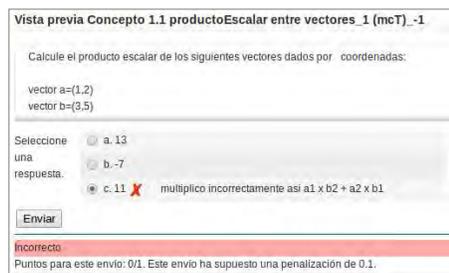


Ilustración 15: Vista previa

## Conclusiones

Inicialmente diseñamos una herramienta para resolver problemas inherentes a la evaluación de ítem de programación para ambientes automatizados y masivos, donde la variable del tiempo fuera en la etapa de construcción de la evaluación lo suficientemente pequeño para garantizar la verificación muestreo con una densidad del 10% de los ejercicios y nos encontramos al final que la proporción de tiempo para la generación de un ítem esta el orden de una hora con verificación incluida para una cantidad de 100 ejercicios +- 10%, se debe remarcar que el diseño de los modelos de las respuestas no es parte de este calculo y que solo responde a la generación de variaciones.

## Bibliografía

- Bartó, Carlos y Díaz Laura (2012):PROYECTO: SISTEMAS INTELIGENTES APLICADOS A LA ENSEÑANZA DE LA PROGRAMACION EN INGENIERIA, WICC Workshop de Investigadores en Ciencias de la Computación. 26 y 27 de abril de 2012 - Posadas, Misiones
- Bornat, Richard Dehnadi, Saeed y Hamilton, Simon(2008): Mental models, Consistency, and Programming Aptitude. Autralian Computer Society. ACE 2008.
- 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; Eschoyez, Maximiliano y Bartó, Carlos (2011): Educación Continua para Cursos Masivos: Un Software Basado en Web para Crear Preguntas para Moodle. CRUNIC 2011 Congreso Red Universitaria Nacional de Ingenieria en Computación, Córdoba 29 de septiembre al 1ro de octubre del 2011.
- 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.