

Sistema de Generación Personalizada de Ejercicios de Lógica Computacional

Jose M. Alcaraz Calero^{1,2}, Luis Daniel Hernandez Molinero¹,
Santiago Paredes Moreno¹

¹Departamento de Ingeniería de la Información y las Comunicaciones
Universidad de Murcia
30100 Murcia
{jmalcaraz, ldaniel, chapu}@um.es

²Cloud and Security Lab
Hewlett-Packard Laboratories
BS34 8QZ Bristol
{jose.alcaraz-calero}@hp.com

Resumen

El presente artículo describe el diseño, desarrollo e implementación de un framework que permite a los docentes la generación tanto de boletines entregables de ejercicios como de tutoriales interactivos de ejercicios con su explicación y solución. Estos recursos son personalizados para cada alumno porque se utiliza un repertorio de ejercicios adecuado al nivel del conocimiento que cada cual tenga sobre la materia. Este framework también dificulta algunos malos hábitos entre los alumnos como la copia de ejercicios o la distribución de los mismos, posiblemente con soluciones incorrectas. Es especialmente útil en asignaturas donde se requiere una fuerte componente de notación matemática tales como son la asignatura de lógica computacional o de matemáticas ya que está basado en el procesador de textos *LaTeX*. Además, esta herramienta da soporte a nuevas metodologías docentes más personalizadas tanto en el aula como fuera de ella.

Summary

This article describes the design, development and implementation of a framework which enables teachers the generation of both home works and interactive tutorials of exercises with explanation and solution. These resources are personalized for each student since it is used a exercises repository adapted to the knowledge level of every student have about the subject. This framework makes difficult some bad habits such as plagiarism and distribution of such exercises, maybe with

incorrect solutions. It is especially useful in subjects in which is required a strong components of mathematical notation such as computational logics and mathematics since it is based on *Latex* word processor. Moreover, this tools offer support for new teaching methodologies more personalized in both classroom and autonomous learning.

Palabras clave

Ejercicios, Notación, Personalización, TIC, Recurso Metodológico, Interactivo.

1. Motivación

En cualquier asignatura que requiere de un lenguaje formal como son las asignaturas de lógica computacional y matemáticas se presenta una problemática importante en el diseño de actividades para el alumno y su posterior corrección. En efecto, la propuesta de actividades de dificultad creciente requiere de un gran dedicación en su diseño y puede que aún más en su escritura formal, además si éstas son de desarrollo, p.e. la demostración de un teorema, y debido a su naturaleza, es muy posible que requiera de mucho esfuerzo para su corrección y si son de ejercitación, p.e. calcular una tabla de verdad, es muy fácil el plagio entre alumnos. Tradicionalmente, este tipo de asignaturas están asociadas al uso de la pizarra como principal mecanismo de interacción alumno-profesor en clase por lo que se plantea una problemática

especialmente acentuada cuando se desea hacer uso de las tecnologías de la información y las comunicaciones (TIC) para el diseño de actividades en formato electrónico, pues la usual interfaz hombre-máquina no está precisamente adaptada a notaciones formales, complicando mucho la realización de actividades basadas en ordenador, tanto en clase como fuera de ella.

La mayoría de las herramientas disponibles hoy en día, especialmente las orientadas a la evaluación, crean ejercicios demasiado estrictos en la introducción de datos del usuario [1]. Si bien es cierto que existen herramientas, que recurriendo a tecnologías estándares, ayudan a mostrar el lenguaje formal matemático de forma on-line, p.e. [2], no es menos cierto que éstas son difícilmente parametrizables a las preferencias de cada profesor o al nivel de dificultad de las necesidades formativas propias de los alumnos.

Por ello, este artículo describe un framework capaz de proporcionar al profesor la posibilidad de generar tanto boletines entregables de ejercicios personalizados para cada alumno como tutoriales interactivos de ejercicios con sus respectivas explicaciones y soluciones proporcionando un nuevo mecanismo de interacción alumno-profesor y dando pie a un posible cambio de metodología de enseñanza tanto en el aula como en el diseño de las actividades que conforman el trabajo autónomo del alumno, necesidad que pone de manifiesto el EEES [3].

El resto de este artículo está distribuido de la siguiente manera: La sección 2 describe los trabajos relacionados con la utilización de las TIC para la personalización de ejercicios y tutoriales. La descripción de la arquitectura del sistema propuesto esta detallada en la sección 3. La sección 4 describe los lenguajes utilizados para describir la batería de ejercicios y boletines utilizados en el sistema. La sección 5 expone algunos detalles de implementación. Finalmente, la sección 6 proporciona algunas conclusiones y trabajos futuros.

2. Trabajos Relacionados

En los últimos años han ido apareciendo numerosos trabajos sobre el uso de las TIC como mecanismo mediador de comunicaciones alumnos-profesor dentro de las metodologías docentes.

Así, *Castel de Haro et al.* [4] proporciona un enfoque sobre la utilización de las TIC para hacer más interesante el estudio de la lógica computacional. Mientras que esta es una primera aproximación a las metodologías docentes adecuadas al EEES, todavía existe una carencia importante en el soporte de herramientas que permitan la personalización de ejercicios y tutoriales al alumno.

En este sentido, *Lu* [5] proporciona un sistema recomendador de material de e-learning personalizado para el alumno de acuerdo a sus necesidades basado en el feedback proporcionado por el alumno. En la misma línea, *Holohan et al* [1] proporciona un mecanismo para llevar a cabo la generación de ejercicios para *e-learning* basados en ontologías.

No es menos cierto que en el intento de avanzar tecnológicamente nos encontramos con ciertos efectos colaterales como indica *Lathrop and Foss* [6] al poner de manifiesto los nuevos retos contra el plagio y el engaño en la nueva era de Internet, enfatizando la necesidad de diseñar sistemas capaces de identificar estos hechos.

Si bien la idea de *Lu* y *Holohan et al* está muy cercana al procedimiento que presentamos en este artículo para la selección del material de *e-learning* ofrecido al alumno en su proceso de auto-aprendizaje, también consideramos imprescindible la personalización de los boletines de prácticas para dificultar el plagio y el engaño de los alumnos. La consecuencia de todo ello es la creación de una infraestructura que permite la creación de tutoriales de aprendizaje interactivos.

3. Arquitectura del Sistema

La Figura 1 representa una visión global de la del sistema de entrega de ejercicios y/o tutoriales.

El sistema de entrega presenta una arquitectura cliente-servidor en la que el estudiante hace uso de cualquier navegador web para acceder al sistema. Una sesión normal en esta arquitectura es la siguiente:

- 1.- Inicialmente, el estudiante accede al módulo de autenticación mediante su dirección de correo electrónico, o DNI, y contraseña. Este mecanismo de seguridad restringe el uso del sistema y permite establecer un entorno controlado del mismo, de tal manera que solo los

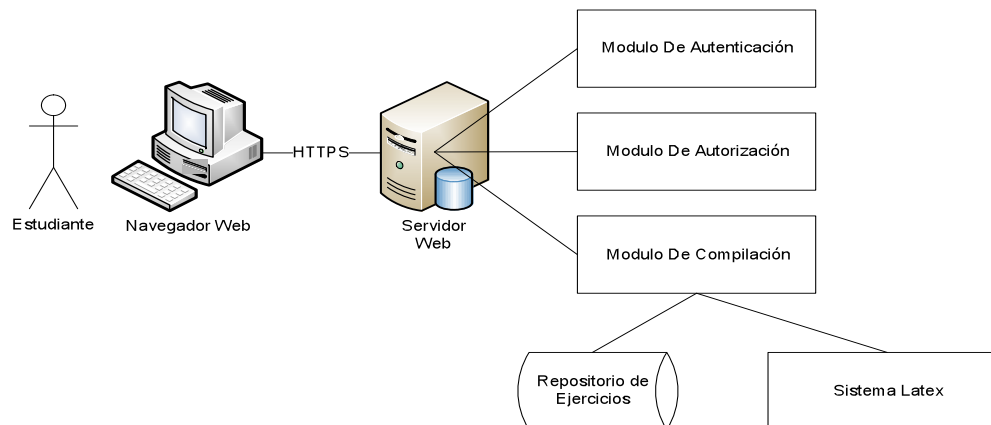


Figura 1. Visión general de la arquitectura del sistema

usuarios que el profesor decida puedan utilizar la aplicación.

2.- Autenticado el usuario, se detectan los boletines y/o tutoriales a los que puede acceder en función del grupo al que pertenece utilizando el módulo de autorización. Esto permite a la aplicación la versatilidad suficiente para gestionar diferentes grupos de usuarios/ asignaturas. Además, mediante este mecanismo de autorización se controla la privacidad entre los contenidos de diferentes asignaturas de tal manera que los usuarios no puedan acceder a boletines de otras materias con el fin de que se centren solo y solo en una asignatura.

3.- Determinados los boletines a los que el alumno tiene acceso, de entre todos los disponibles en el repositorio de boletines, se le da la posibilidad de seleccionar un boletín sobre uno o varios tipos de ejercicio. Entendemos por boletín al compendio de enunciados de ejercicios que deben cumplimentarse por el alumno y pueden, o no, presentarse con su solución. Obviamente, este boletín es totalmente personalizado y particularizado para ese alumno. Para determinar cómo se lleva a cabo este proceso de personalización véase la sección 4 en la que se describe el diseño del repertorio de ejercicios.

4.- Generado el boletín de ejercicios, se le entrega al usuario que puede, si así lo desea, descargarlo en su disco local.

La clave de todo el proceso está en disponer de una buena base de ejercicios y el formato en que éstos deben mostrarse, y es en lo que incidimos en los siguientes apartados.

4. Repositorio de Ejercicios

La arquitectura utiliza diferentes fuentes de datos para representar el repositorio de ejercicios. Se utiliza un único fichero denominado *fichero de boletines* para definir los diferentes boletines y/o tutoriales que se administran en la aplicación. De ahora en adelante se referirá indistintamente a boletín y a tutorial, pues aunque atiende a conceptos distintos la aplicación soporta ambos por igual. Cada boletín está estructurado necesariamente por una cabecera, un pie de página y un conjunto de ejercicios. Este fichero se explica con detenimiento en la sub-sección 4.1.

Aunque la cabecera, el pie de página y los ejercicios sean elementos diferentes, todos se gestionan de la misma manera dentro de la aplicación. La aplicación posee tres directorios denominados *headers*, *footers*, *exercises* y ellos contienen todas las posibles cabeceras, pies de página y ejercicios que pueden ser manejados por la aplicación. Cada uno de estos directorios posee un par de ficheros denominado *fichero de plantilla* y *fichero de instancias* para cada pie de página, cabecera y tipo de ejercicio (asistido o no) que puede aparecer en un boletín/tutorial. Estos ficheros definen una plantilla del tipo de ejercicio a generar en la aplicación y una batería de ejemplos para esa plantilla, respectivamente. Ambos serán explicados en las subsecciones 4.2 y 4.3. La Figura 2 muestra la estructura de directorios y ficheros utilizada en la arquitectura. En dicha figura se pueden observar los tres directorios con sus respectivos pares de ficheros

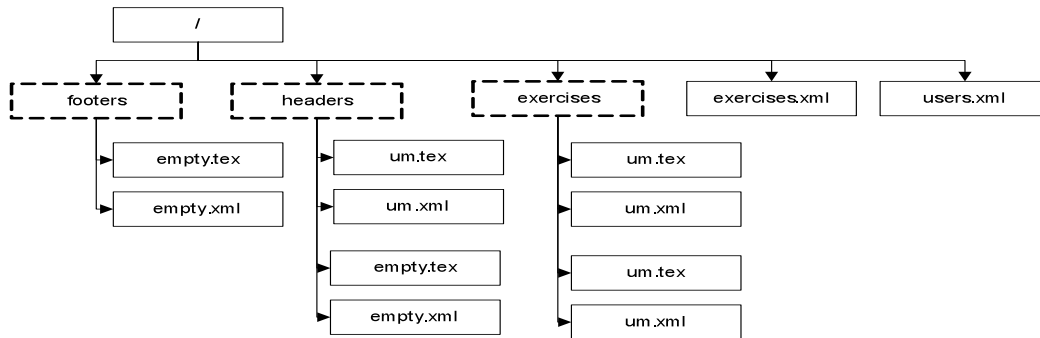


Figura 2. Estructura de ficheros propuesta

plantilla y de instancias. Además, el fichero de boletines (*exercises.xml*) y el fichero en el que se indican los usuarios, contraseñas y roles de cada usuario del sistema (*users.xml*).

4.1. Fichero de Boletines

El fichero de boletines contiene todos los boletines que el profesor desea gestionar en la aplicación (*exercises.xml*). Éste está estructurado necesariamente por cabecera, pie de página y un conjunto de ejercicios. El Listado 1 muestra un ejemplo de definición de boletín que se encuentra dentro de este fichero:

```

<bulletin bulletinId="entrega1"
  description="1º Entrega"
  randomOrder="false"
  groupsId="g1-logica">
  <header headerId="um"/>
  <exercise exerciseId="L0-n0-a"
    randomEntry="true"/>
  <exercise exerciseId="L0-n0-a"
    randomEntry="false"
    exerciseEntryId="1"
    shuffleCases="true"/>
  <footer footerId="empty"/>
</bulletin>
  
```

Listado 1. Ejercicio de definición de boletín

Como puede observarse, un boletín (identificado con la etiqueta XML *boletín*) puede

pertenecer a uno o más grupos de alumnos (atributo *groupsId*). Un grupo de alumnos es una agrupación lógica que decide el profesor y que puede atender tanto al concepto usual de grupo teórico, como al de subgrupo de prácticas, así como a cualquier otro tipo de grupo con el número de alumnos que considere adecuado el docente para aplicar cualquier metodología e-blending o activa [7][8]. Solo los alumnos del grupo al que está asociado el boletín pueden ver el mismo. Además un boletín puede configurarse en modo *orden aleatorio*, o no, indicando si los ejercicios que componen el boletín se presentan en el orden preestablecido por el profesor o por el contrario los ejercicios se presentan en un orden aleatorio, dificultando así el plagio, copia o engaño por parte del alumno. Esta opción es especialmente útil si desea generarse boletines de prácticas muy individualizados o bien incluso para generar exámenes que afecten a la calificación final de la asignatura.

Note que cada boletín está compuesto de una cabecera, una serie de ejercicios y un pie de página, todos ellos indicados mediante un *identificador*, las expresiones terminadas en *Id*. Este identificador ha de coincidir con el nombre de los ficheros que contienen los ejercicios que se desea utilizar. Este manera de enlazar boletines con ejercicios permite a los profesores mucha versatilidad y sencillez en los cambios, pues añadir un nuevo ejercicio o una nueva cabecera solo consiste en crear un nuevo fichero, colocarlos en la ubicación correspondiente y usar el nombre de ese fichero como identificador.

Cuando nos referimos a un ejercicio hay que distinguir dos niveles de abstracción. Por un lado

está el tipo de ejercicio para hacer referencia a aquel que se diseña para conseguir un objetivo formativo de la asignatura. Por otro lado están las instancias de ese tipo de ejercicios para hacer referencias a casos particulares de ejercicios. Por ejemplo, un tipo de ejercicio es calcular tablas de verdad, y es una instancia aquel que pide determinar la tabla de verdad para una expresión lógica particular.

El identificador *exerciseld* denota el tipo de ejercicio que se va a incluir en el boletín pero no así la instancia concreta de dicho ejercicio. Es decir, los ficheros de boletines especifican el tipo de ejercicios que deben generarse y no entran en qué ejercicios concretos (instancias) se desean incluir.

El motivo de este diseño es para conseguir la suficiente versatilidad para poder extender fácilmente la batería de ejercicio a la vez que se consigue la separación de niveles de dificultad de los mismos, la ordenación de los diferentes tipos de ejercicios y la fácil modificación de todos ellos (a través de la plantilla) por tanto el sistema se ha diseñado de manera que cada tipo de ejercicio tiene asignada la misma dificultad en todas sus instanciaciones del repertorio y por tanto dos ejercicios diferentes podrían tener exactamente el mismo enunciado (fichero plantilla) pero las instancias proporcionan un nivel de dificultad mayor o menor.

Además, la característica de aleatoriedad sobre los tipos de ejercicios hace que cada alumno puede tener potencialmente ejercicios análogos a los de sus compañeros pero no iguales, obligándolo, en cierta medida, a la necesidad de realizarlo si desea realmente aprender y dificultando notablemente el plagio de los mismos. Por otro lado, el profesor, si así lo desea, puede también generar siempre el mismo ejercicio para todos los alumnos (ver el segundo ejercicio del Listado 1), para ello ha de desactivar la característica de aleatoriedad (atributo *randomEntry*) e indicar el identificación de la instanciación que desea utilizar (versión sección 4.3 para más información).

Otra característica diferenciadora de este sistema es la posibilidad de barajar (*aleatorizar*) las diferentes opciones que se den en la instancia de un ejercicio concreto (atributo *shuffleCases*). Obviamente, esta opción solo tienen sentido cuando la respuesta es de elección múltiple, V/F,

etc. Esto se utiliza para evitar situaciones en las que todos las instancias de ejercicios del mismo tipo tiene un patrón predecible para mostrar la solución y los alumnos podrían detectarlo y intuir siempre la solución correcta.

4.2. Fichero de Plantilla

Este fichero se escribe utilizando el lenguaje *LaTeX*, y es el resultado de unir la cabecera, los ejercicios instanciados y el pie de página de un boletín. Una vez generado, es compilado con *pdfLaTeX* para obtener el fichero PDF correspondiente y que es el enviado al alumno.

El contenido de estos ficheros plantilla es totalmente personalizable y la única exigencia real es que el resultado de las concatenaciones de estos ficheros ha de ser un fichero *LaTeX* que compile correctamente. Aun así, para el contexto de la generación de ejercicios interactivos y tutoriales, se ha optado por la utilización de *AcroTeX* [9], un paquete de ficheros para *LaTeX orientados a la docencia*. Este paquete permite hacer ejercicios utilizando toda la potencia hipermédia, formularios y javascript embebido en ficheros PDF. Así, por ejemplo, se pueden diseñar ejercicios con soluciones interactivas, enlaces, animaciones gráficas, ejercicios de múltiple sección, ejercicios con gráficos y un largísimo etcétera. *AcroTeX* es la piedra angular de la arquitectura presentada en este trabajo de docente. Concretamente, *Acrotex* define un entorno *sortquiz* en el que se indican los ejercicios y un entorno *solution* en el que se indican las soluciones a dichos ejercicios y las explicaciones de dichas soluciones (para hacer tutoriales).

El Listado 2 muestra un ejemplo de fichero de plantilla que, utilizando los entornos *sortquiz* y *solution* de *AcroTeX*, generará un ejercicio que se auto-corrige de forma automática cuando el alumno introduce su solución. Responde a un sencillo ejercicio del tipo V/F en el contexto de la lógica computacional: se muestra al alumno una fórmula y se pregunta si es una fórmula bien formada o no.

En el Listado 2 tiene una mención especial los *anclajes*. Un anclaje es un patrón de la forma $\{variable\}$ donde *variable* es el nombre del anclaje (en el ejemplo hay cinco). Cada uno de estos anclajes serán posteriormente rellenados con los valores de las diferentes instancias de ejercicios que se introduzca en el fichero de

instancias. El número de anclajes depende totalmente de la plantilla que se esté diseñando (del tipo de ejercicio) y el sistema ha sido diseñado para soportar cualquier número de anclajes. Este número determinará cuantas variables deberá tener el fichero de instancias que se defina para este ejercicio.

Note también que el Listado 2 está dividido en 3 partes, que es el único patrón estructural que, por conveniencia y no por limitaciones de programación, conviene respetar en el diseño de los boletines. La primera de ellas se ha realizado con la única finalidad de separar al máximo las variables que han de instanciarse del resto de la plantilla. En esta sección pueden verse 3 comandos (*\expresion* *\valoruno* *\valordos*) que se redefinen con los valores que serán instanciados con el fichero de instancias y que permiten referenciar a estos valores antes de ser instanciados. Luego se define el enunciado (en el que se usa la expresión anteriormente definida) y luego se indica la solución, indicando que *\valoruno* es la solución correcta si el valor de la variable/anclaje *\$R1_solucion* es 1 o incorrecta si es 0.

```
%---- Variables del ejercicio
\renewcommand{\expresion}{
${expresion}}

\renewcommand{\valoruno}{
${R1_valor}}

\renewcommand{\valordos}{
${R2_valor}}

%---- Ejercicio

%---- Enunciado
\begin{shortquiz}
Observa la siguiente expresión y
pincha sobre la opción correcta:
$$\expresion$$

%---- Soluciones
\begin{answers}{2}
\Choices[random]
  \Ans${R1_solucion}
  \valoruno \eAns
  \Ans${R2_solucion}
  \valordos \eAns
\end{Choices}
\end{answers}
\end{shortquiz}
```

\newpage

Listado 2. Ejemplo de plantilla de ejercicio

4.3. Fichero de Instancias

Este fichero contiene todas las instancias de un tipo de ejercicio y tiene formato XML. Cada una de estas instancias (definida con la etiqueta *exerciseInstance*) está compuesta por tantas variables como puntos de anclaje tenga la plantilla asociada. La asociación entre fichero de plantilla y de instancias se hace a través del nombre del fichero (ambos se llaman igual, pero con diferente extensión), lo que dota de versatilidad al sistema.

```
<exerciseInstance id="1">
  <v name="expresion">p \neg q</v>
  <v name="R1_valor">No es fbf</v>
  <v name="R1_solucion">1</v>
  <v name="R2_valor">Es fbf</v>
  <v name="R2_solucion">0</v>
</exerciseInstance>
```

Listado 3. Ejemplo de instancia del ejercicio.

El Listado 3 muestra un ejemplo de instancia para la plantilla anteriormente descrita en el Listado 2. Nótese como las 5 variables disponibles en la plantilla son reemplazadas por sus correspondientes valores. Además, nótese la facilidad en la que un profesor añade un nuevo ejercicio: simplemente indica que una expresión cualquiera, tiene dos valores: “No es Formula Bien Formada (FBF)” y “Es FBF” y que uno de ellos es correcto. Fruto de la experiencia se ha comprobado que simplemente escribiendo el enunciado del ejercicio como comentario XML al principio del documento, este fichero se puede rellenar sin necesidad de recurrir a la plantilla para nada, ya que un simple *copy & paste* del ejercicio anterior te da la información suficiente para hacer otra nueva instancia para el repositorio.

Cada instancia tiene un identificador que puede ser utilizado para indicar en los boletines que es esa instanciación concreta la que se desea que aparezca en los mismo (atributo *id*). Además, merece la pena mencionar que el sistema seleccionar la instancia del ejercicio (o al azar o una concreta) y reemplazara los valores en la plantilla, *instanciándola* y generando la porción

de fichero l atex que ser  concatenada a las otras porciones que componen el bolet n.

Aunque el nombre que se puede dar a los puntos de anclaje es totalmente libre para el usuario, n tese que en el Listado 3, algunos tienen el patr n: “ R ” + *Numero* + “_” + *Nombre de variable*”. Esta es la manera de indicar que si se desean reordenar las opciones de forma aleatoria, solo se va a aleatoria aquellas que empiecen con R + *Numero* + “_” y adem s, todas las que tienen el mismo n mero han de mantener el mismo remplazo de aleatoriedad. Por ejemplo, para el Listado 3, si el ejercicio del bolet n tiene la opci n (*shuffleCases* activada), puede ocurrir que se haga el reemplazo ($1 \leftrightarrow 1$) o ($1 \leftrightarrow 2$). Suponiendo el  ltimo caso, el ejercicio quedar  finalmente como se muestra en el Listado 4.

```
<exerciseInstance id="1">
  <v name="expresion">p \neg q</v>
  <v name="R1_valor">Es fbf</v>
  <v name="R1_solucion">0</v>
  <v name="R2_valor">No es fbf</v>
  <v name="R2_solucion">1</v>
</exerciseInstance>
```

Listado 4. Ejemplo de instancia del ejercicio despu s de ser aleatorizado

5. Implementaci n

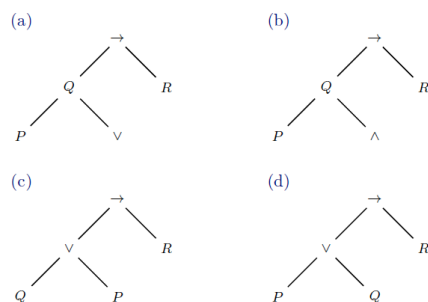
El sistema ha sido implementado utilizando *MiKTeX* como motor de compilaci n de *LaTeX*, en su versi n portable, lo que permite hacer una aplicaci n web que pueda ser distribuida sin necesidad de dependencias del sistema operativo. Adem s, la aplicaci n web ha sido implementada en Java *Servlet* y *JSP* para llevar a cabo la implementaci n de la interfaz del usuario. Para la gesti n de plantillas y reemplazo de variables se ha utilizado una librer a Java llamada *FreeMarket*. La aplicaci n ha sido publicada bajo el nombre de *Acrologic* con licencia GPL en el portal *sourceforge* y puede ser descargada gratuitamente en la siguiente direcci n: <https://sourceforge.net/projects/acrologic/>. Los ejercicios dise ados para la asignatura de l gica computaci n no son distribuidos, ni han sido publicados bajo ninguna licencia.

La Figura 3 muestra un ejemplo de ejercicio que ha sido generado autom ticamente para el usuario registrado en el sistema y entregado al

mismo a trav s de Internet mediante un fichero PDF que se abre autom ticamente en el ordenador del alumno usando el tipo mime apropiado.

EJERCICIO 1. Observa cada  rbol y explica si puede ser un  rbol sint ctico que se corresponda con la siguiente expresi n¹:

$$(P \vee Q) \rightarrow R$$



¹Pincha sobre la letra de cada uno para conocer la respuesta.

Figura 3. Ejemplo de ejercicio generado

6. Discusi n

La personalizaci n hacia los alumnos y la creaci n de materiales que fomenten el trabajo aut nomo del alumno son dos de los pilares fundamentales en los que se basa el EEES. El sistema propuesto en el presente art culo permite ser utilizado en un gran abanico de metodolog as docente que contemple actividades en clase o fuera de ella, la realizaci n de pr cticas y boletines, la construcci n de tutoriales o ejercicios de autoevaluaci n en funci n del nivel de cada alumno o la realizaci n de ex menes tanto on-line como en papel. Sin duda es un sistema flexible que permite ahondar en innovaciones docentes que utilicen las nuevas tecnolog as dentro de la ense anza, de forma que conformen una parte activa dentro de la metodolog a de aprendizaje del alumno. N tese que esto es especialmente relevante en carreras como la Ingenier a Inform tica en la que, en general, conviene mostrar con el ejemplo la utilizaci n de los ordenadores como medios de ense anza.

7. Conclusiones

Se ha propuesto un sistema para la generaci n de boletines y tutoriales de forma personalizada para

los alumnos utilizando para ello un repositorio de ejercicios extensible. El sistema tiene una fuerte componente de seguridad para el plagio y el engaño de los alumnos. Además, se ha propuesto un mecanismo de representación de los repositorios de ejercicios que ha sido probado con éxito y validado en la implementación que ha sido llevada a cabo del sistema.

Como trabajo futuro se pretende diseñar e implementar un sistema para recibir las respuesta de los ejercicios de los alumnos (aquellas que no sean tutoriales) e insertarlas automáticamente en una base de datos con dos fines: disponer información de todos los alumnos sobre el tipo de preguntas que son capaces de resolver (y obtener así un perfil de la clase) y disponer información individualizada (y poder así tener el perfil de alumno). Estas informaciones permiten generar nuevas versiones de los boletines incluyendo una componente más pragmática que permita adaptar los nuevos boletines generados a las necesidades reales de los alumnos. También esperamos aumentar el sistema con nuevas funcionalidades como son el acceso al PDF con contraseña, el establecimiento de tiempo límite para realizar el boletín (después del cual los resultados se enviarán automáticamente al profesor o a una base de datos), la imposibilidad de abrir el fichero PDF más de un vez, y, en definitiva, todas aquellas característica que sería deseables para adaptar el sistema de tal forma que contribuya en la metodología docente del aula y el uso de las TIC en la realización de exámenes con un cierto grado de seguridad.

Agradecimientos

Los autores quieren agradecer a la Universidad de Murcia y a la Comunidad Autónoma de la Región de Murcia su soporte para llevar a cabo esta investigación bajo la convocatoria de ayudas para proyectos de innovación docente con TIC para el curso 2010/2011. Además, los autores también quieren agradecer a la Fundación Seneca por soportar el trabajo de Jose Maria Alcaraz Calero bajo las beca post-doctoral 15714/PD/10.

Referencias

- [1] Holohan Edmond, Melia Mark, McMullen Declan, Pahl Claus. *The Generation of E-Learning Exercise Problems from Subject Ontologies*. Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06). pp. ISBN: 967-969. 0-7695-2632-2
- [2] Soler Masó, Josep; Boada i Oliveras, Imma; Prados Carrasco, Ferran; Fabregat, Ramon.; Poch Garcia, Jordi. Experiencia docente en diseño de bases de datos con la ayuda de herramientas de e-learning. Jornadas de Enseñanza universitaria de la Informática. JENUI 2009
- [3] Rué Joan. *Enseñar en la Universidad: El EEES como Reto para la Educación Superior*. Narcea, 2007.
- [4] Castel de Haro M^a Jesús, Villagrà Arnedo Carlos, Palomino Benito Juan Alejandro. *Materiales Docentes Con Los Que Aumentará Tu InterEEES Por Estudiar Lógica*. Jornadas de Investigación en Docencia Universitaria : la construcción colegiada del modelo docente universitario del siglo XXI. pp 164. 2008. ISBN 978-84-691-4559-3
- [5] Lu Jie. *Personalized e-learning material recommender system*. International conference on information technology for application. pp. 374-379, 2004.
- [6] Lathrop Ann, Foss Kathless. *Student cheating and plagiarism in the Internet era: A wake-up call*. Libraries Unlimited. 2000.
- [7] González Boticario Jesús, Gaudioso Vásquez Elena. *Aprender y formar en Internet*. Thmoson Learning. 2001
- [8] Valero García, Miguel; Vaquerizo García, Belén. Puzzles mejorados con mapas conceptuales. Jornadas de Enseñanza universitaria de la Informática. JENUI 2009.
- [9] Story D. P. The AcroTEX eDucation Bundle. 2006. Available at <http://www.acrotex.net/>