

Herramientas de autoevaluación tipo test on-line y off-line freeware: Hot-Potatoes y SED.

Las NNTT aplicadas a la educación y particularmente el uso de Internet permiten la creación de cuestionarios tipo test interactivos que son muy útiles a la hora de motivar al alumno en las tareas de aprendizaje y de permitir que sea él mismo el que constate si posee ya los conocimientos sobre los que el profesor lo evaluará posteriormente.

La utilización de herramientas que generan y sirven para cumplimentar cuestionarios test hacen pensar al profesor en usarlas también para evaluar realmente al alumno. Sin embargo, posibles fallos de seguridad en las mismas y la necesidad de un soporte escrito y firmado por el alumno que garantice la autoría y veracidad de la prueba provocan que seamos reacios muchas veces a evaluar utilizando dichas herramientas. En la siguiente ponencia se explicarán dos herramientas freeware utilizadas para la generación/cumplimentación de cuestionarios test por web: Hot Potatoes (off-line) y el Sistema Experto Docente (SED) (on-line) desarrollado por el GIDIVI (Grupo de Investigación de Docencia Virtual) de la Universidad Miguel Hernández . Además se comentará la integración de las mismas de forma que Hot Potatoes se utilice para confeccionar preguntas cuando no se dispone de conexión a Internet y dichas preguntas puedan ser importadas por el SED para hacer cualquier tipo de cuestionario, ya sea de autoevaluación o realmente de evaluación. Por otra parte discutiremos ciertas técnicas de seguridad para evitar la copia y certificar la autoría como la reordenación de preguntas, los tiempos límite de examen y la generación de copias encriptadas que permiten ser leídas y por el alumno y no modificadas, dichas copias de examen además pueden ser guardadas por el mismo como justificante. Añadiremos también algún video en formato AVI donde se muestre el uso de dichas herramientas.

Autores:

Carlos Pastor Antón (Autor Principal)
Marco A. Marhuenda García
Alejandro Pomares Padilla
Salvador P. Sánchez Pérez
Salvador Alcaraz
Mar Galotto Tebar
Yolanda Marhuenda García

Introducción

La necesidad de constatar fehacientemente los conocimientos adquiridos en un proceso de aprendizaje es un proceso ineludible tanto para profesores como para los mismos alumnos interesados en conocer su progreso; como simple y actual botón de muestra tenemos por ejemplo que el borrador de la 'Ley de Calidad de la Enseñanza'¹ hace un gran hincapié en dicha evaluación, en concreto con la implantación de la Prueba General de Bachillerato también conocida como Reválida.

Con la aparición del uso de las Nuevas Tecnologías (NNTT) en el ámbito de la Educación, y especialmente con el desarrollo de plataformas de docencia virtual (DV) vía web se corre

el riesgo de perder la interacción directa profesor-alumno que conlleva la evaluación de los conocimientos adquiridos, ya sea mediante pruebas escritas u orales típicas, o simples preguntas y gestos que se producen en una clase presencial.

Para evitarlo la mayoría de las herramientas comerciales de DV (WebCT^{II}, TopClass^{III}, IBM MindSpan Solutions (Lotus Learning Space)^{IV}) incorporan módulos de evaluación tipo test que intentan solventar esta carencia. Dichos módulos de test se podrían clasificar en 2 tipos:

Online (realizados en tiempo real contra el servidor web de la aplicación, y cuyos resultados se pueden almacenar) y Offline (descargados de la aplicación al computador del alumno y cuyo resultados pueden ser devueltos al servidor de forma opcional).

En el mundo de las herramientas freeware de DV, y teniendo en cuenta la metodología de diseño de dichas herramientas^V, no tenemos constancia de ningún software que realice tal labor. Debido a esto, y al gran desarrollo que está teniendo actualmente el software GNU^{VI}, el Grupo de Investigación de Docencia Virtual (GIDIVI^{VII}) decidió incorporar a su herramienta de DV Sistema Experto Docente (SED)^{VIII} que realizaba test Online, una herramienta de test Offline, HotPotatoes^{IX} que es gratuita para educación.

En los siguientes apartados describiremos ambas herramientas y comentaremos luego su integración y algunas consideraciones sobre seguridad a la hora de realizar los test.

SED

El SED^X es una herramienta de DV desarrollada sobre Linux^{XI}, con Apache^{XII} de servidor web y MySQL^{XIII} como gestor de base de datos. Las páginas de la aplicación están programadas en PHP^{XIV}. Actualmente se encuentra en producción a disposición de toda la enseñanza reglada de la Universidad Miguel Hernández^{XV}. Concretamente, su módulo de test se divide en dos partes, una para el alumno y otra para el profesor, en la siguiente figura tenemos el interfaz del profesor:



Ilustración 1: Interfaz del Profesor del Módulo de Autoevaluación

El método para utilizarlo en Test de Autoevaluación al alumno es sencillo, según lo podemos constatar viendo por ejemplo la introducción de un test donde la única pregunta ejemplo es:

Una página web tiene extensión:

- *Html o htm*
- *Doc o mdb*
- *Exe*

Donde la respuesta correcta es "Html o htm"

1. Crear *Base de datos de Cuestiones*. Las cuestiones se asocian a temas de una asignatura, hay dos tipos de cuestiones: *Test* y de *Desarrollo*. El aspecto del interfaz gráfico de la introducción de una cuestión en la Base de Datos de cuestiones sería:

The screenshot shows a web browser window with the following elements:

- Page Title:** Sistema Experto Docente - Microsoft Internet Explorer
- Section: Texto de la Pregunta**
 - Castellano: Una página web tiene extensiones:
 - Inglés: (empty)
 - Valenciano: (empty)
- Section: Opciones - Respuestas**

a)	Cas:	html o htm	Ing:	
b)	Cas:	doc o mdb	Ing:	
c)	Cas:	exe	Ing:	
- Section: Temas**
 - 1 - ¿tiene falta modem para conectarse a Internet?
 - a) No, con un navegador y una red local con acceso a Internet sobra.
 - b) Si, porque sino solo tenemos acceso a archivos locales.
 - c) Las dos anteriores son validas.
- Page Footer:** Tema 5: El Lenguaje HTML Básico

Las cuestiones *Test* tienen una única respuesta válida (**Selección Única**) y la representación en XML^{XVI} de la pregunta ejemplo anterior sería la siguiente:

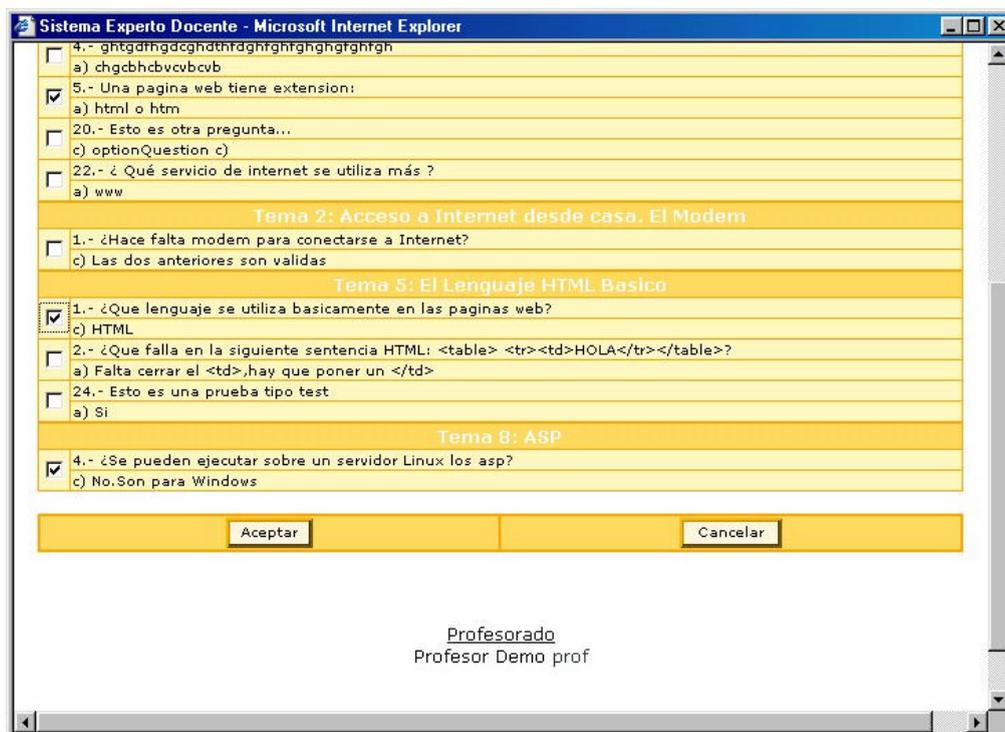
```

<?xml version="1.0" ?>
- <question>
  <subjectCode>Asignatura_Demo</subjectCode>
  <themeNumber>1</themeNumber>
  <questionNumber>1</questionNumber>
  <type>TEST</type>
  <text>Una pagina web tiene extension:</text>
  <correctOption>1</correctOption>
- <optionQuestions>
  - <optionQuestion>
    <optionNumber>1</optionNumber>
    <text>Html o htm</text>
  </optionQuestion>
  - <optionQuestion>
    <optionNumber>2</optionNumber>
    <text>doc o mdb</text>
  </optionQuestion>
  - <optionQuestion>
    <optionNumber>3</optionNumber>
    <text>exe</text>
  </optionQuestion>
</optionQuestions>
</question>

```

Ilustración 2: Sed.xml

2. Crear *Pruebas de Autoevaluación*. Seleccionaremos las preguntas que queremos que compongan una prueba concreta y dejaremos dicha prueba en modo OCULTO hasta que consideremos que ha de ser realizado por el alumno, que será cuando lo pasemos a modo REALIZABLE. En la figura siguiente se muestra cómo se crea una prueba de Autoevaluación a partir de la base de datos de cuestiones:



3. Una vez que el alumno haya hecho el Test podemos pasar a *Ver Resultados* para conocer su progreso. Fijémosnos que en la herramienta OffLine (HotPotatoes) esto será imposible ya que no tiene una base de datos asociada donde se guarden las calificaciones de los alumnos.

HotPotatoes

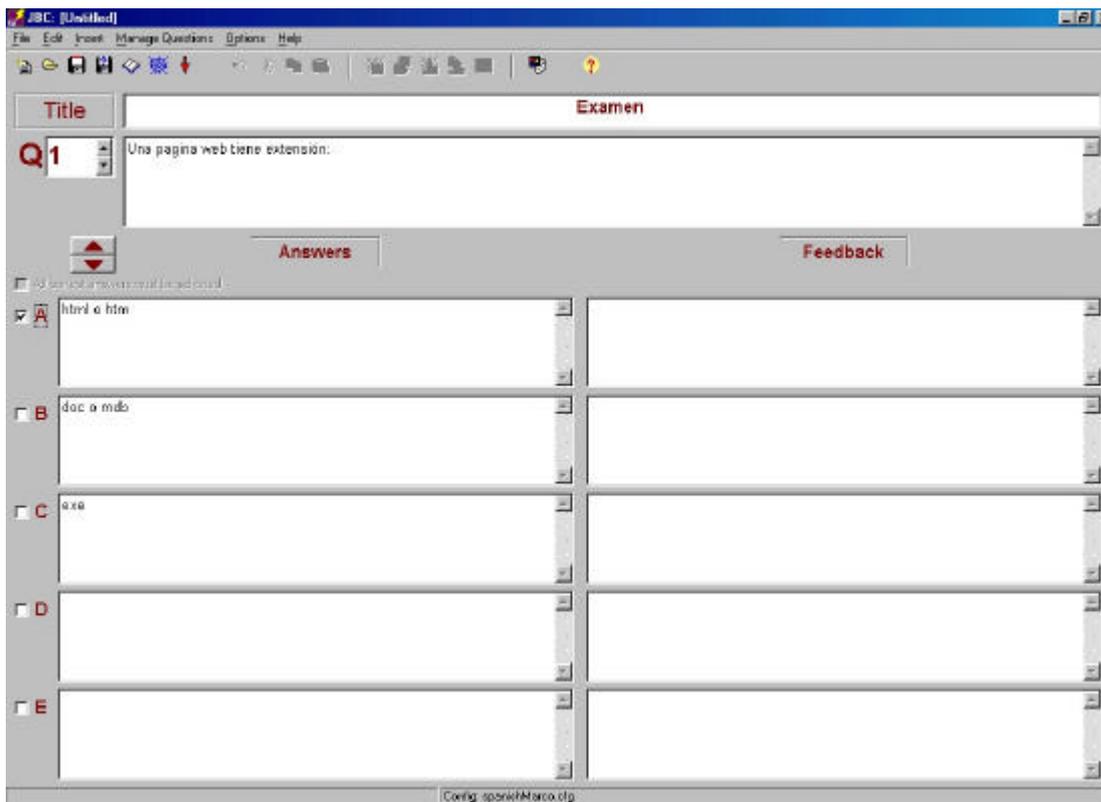
Como se ha comentado anteriormente, HotPotatoes es una herramienta de creación de test off-line gratuita para educación desarrollada por Half Baked Software^{XVII} permite crear en la web test del tipo:

selección múltiple, respuesta corta, crucigramas etc... Para el caso que nos ocupa, dichos test se dejan en cualquier web o en el propio disco duro del alumno y se pueden realizar tantas veces como se quiera.

Como el objetivo del presente trabajo es integrar HotPotatoes con el SED, cabe destacar que una pregunta del tipo selección múltiple se puede convertir en una de selección única si sólo una respuesta es válida. La herramienta de HotPotatoes para realizar este tipo de cuestiones se denomina JBC.

Veamos el método para utilizar HotPotatoes y crear la misma pregunta puesta como ejemplo en el SED:

1. Creamos la pregunta en el módulo JBC.



2. La guardamos como 'pregunta.jbc' y editamos su contenido a efectos de comprobar su similitud con lo generado desde el SED, del cual sólo nos interesa la parte siguiente:

```

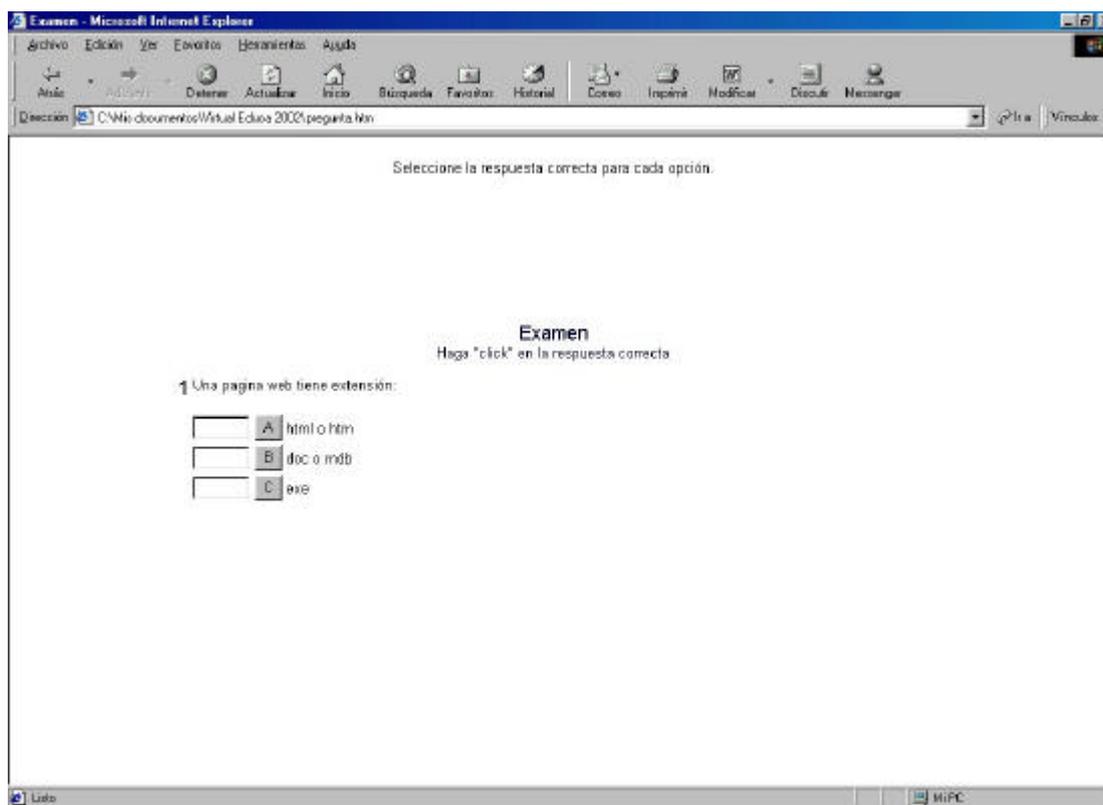
<?xml version="1.0" ?>
- <question-record>
  <question>Una pagina web tiene extension:</question>
  <must-select-all>0</must-select-all>
- <answer>
  <text>html o htm</text>
  <correct>1</correct>
  <feedback />
</answer>
- <answer>
  <text>doc o mdb</text>
  <correct>0</correct>
  <feedback />
</answer>
- <answer>
  <text>exe</text>
  <correct>0</correct>
  <feedback />
</answer>
</question-record>

```

Ilustración 3 : HotPotatoes.xml

Fijémonos que este fragmento del fichero que es donde realmente se almacena la pregunta tiene una estructura muy similar al generado por el SED.

3. Creamos la página web automáticamente desde *File->Export To Web* del JBC. El resultado sería el siguiente:



Integración entre SED y HotPotatoes y consideraciones sobre seguridad

Por último vamos a ver lo sencillo que resultaría crear un software que permitiese convertir el ejemplo de la pregunta en XML del SED al XML del HotPotatoes (quedando para el lector la conversión a la inversa). Con ello quedaría zanjada la utilización de ambas herramientas de manera conjunta pudiendo importar/exportar enunciados de preguntas test entre ambas.

Para realizar dicha conversión utilizaremos XSL Transformations^{XVIII} que es un lenguaje creado 'ex profeso' para realizar este tipo de cambios de formato, el fichero al que llamaremos *sed2hotpotatoes.xsl* y que muestra las órdenes de conversión de un formato a otro sería:

```

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <!-- para cada una de las preguntas del Sed -->
    <xsl:for-each select="question">
      <question-record>
        <question><xsl:value-of select="text"/></question>
        <must-select-all>0</must-select-all>
        <!-- Opcion correcta -->
        <xsl:for-each select="optionQuestions/optionQuestion[optionNumber=/question/correctOption]">
          <answer>
            <text>
              <xsl:value-of select="text"/>
            </text>
            <correct>1</correct>
            <feedback></feedback>
          </answer>
        </xsl:for-each>
        <!-- Opciones incorrectas -->
        <xsl:for-each select="optionQuestions/optionQuestion[optionNumber!=/question/correctOption]">
          <answer>
            <text>
              <xsl:value-of select="text"/>
            </text>
            <correct>0</correct>
            <feedback></feedback>
          </answer>
        </xsl:for-each>
      </question-record>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>

```

Ilustración 4 sed2hotpotatoes.xml

Y para ejecutar dicho proceso podríamos utilizar una de las innumerables herramientas, ‘parsers’ de XML, que en nuestro caso hemos escogido *xt*^{XIX} que utilizando *sed.xml* y *sed2hotpotatoes.xml* generará *hotpotatoes2.xml* desde una ventana MS-DOS:

xt sed.xml sed2hotpotatoes.xml hotpotatoes2.xml

Queda por recalcar que el archivo *hotpotatoes2.xml* ya es compatible con Hot-Potatoes y es idéntico al *hotpotatoes.xml* creado inicialmente.

Una vez visto que con un sencillo archivo en formato xsl podemos transformar cualquier pregunta test del SED a Hot-Potatoes y a la inversa, vamos a pasar a comentar algunas consideraciones sobre seguridad de los exámenes test por web:

Queremos comenzar recalcando que desde un computador conectado a internet la seguridad en un examen es imposible si el examen se hace en casa y prácticamente imposible si lo realiza en el centro ya que el alumno puede en cualquier momento navegar por otras páginas o intercambiar e-mails, mensajes, ficheros etc con sus compañeros de examen, sin embargo, algo sí se puede hacer para aumentarla hasta el extremo de que intentar copiar sea prácticamente inimaginable para el alumno, para ello las medidas son varias:

- 1) Cerrar la salida de la red del aula donde se realiza el examen, dejando abierto únicamente la IP del servidor donde se aloja el examen y el puerto 80 que es donde

se suele alojar la web. Con ello conseguiremos que los alumnos únicamente puedan acceder al servidor web donde se encuentra el examen y no a cualquier otro recurso de Internet, como e-mail etc...

- 2) No permitir las comunicaciones entre los ordenadores de los alumnos cerrando los puertos de comunicaciones de la electrónica de red del aula (siempre que obviamente, dicha electrónica sea lo suficientemente moderna, es decir, es imposible hacer esto en una red BNC o con simples hubs no-programables).
- 3) Para reconocer la autoría del examen y evitar reclamaciones, encriptar con PGP^{XX} el resultado del mismo y enviar una copia electrónica al alumno por e-mail al mismo tiempo que el sistema también guarda su propia copia. Fijémonos que la única forma de desencriptar el mensaje es utilizar la clave pública que tienen tanto el profesor como el alumno y que en una revisión ambos documentos han de coincidir sin desencriptar y una vez desencriptados. PGP se distribuye de forma gratuita y hemos previsto que el SED lo incorpore.
- 4) Por último comentar que mecanismos como los de reordenar preguntas y sus respuestas de forma aleatoria y crear varios tipos de examen también ayudan a evitar la copia.

Sin embargo, nosotros proponemos realizar este tipo de pruebas de forma tan frecuente e intensa y con un peso en la nota tan insignificante que al final al alumno no le interese cometer ningún tipo de fraude. Y además seguir evaluando parte de la asignatura utilizando métodos tradicionales. Para todo ello es fundamental la interacción entre dos herramientas de evaluación tipo test que cubran las necesidades del profesor, tanto cuando tiene una buena conexión a Internet, como cuando no existe dicha conexión o se quiere prescindir de ella por simple comodidad de trabajo.

^I Tenemos un buen comentario del borrador de dicha ley en <http://www.magisnet.com>

^{II} <http://www.webct.com>

^{III} <http://www.wbtsystems.com>

^{IV} <http://www.ibm.com/mindspan>

^V An Extended Methodology for Educational Software Design: Some Critical Points. Fernando J Lage, Yuriy Zubenko and Zulma Cataldi. 31th ASEE/IEEE Frontiers in Education Conference. October 10-13 Reno, NV.

^{VI} Worldwide software-sharing community: <http://www.gnu.org>

^{VII} <http://gidovi.umh.es>

^{VIII} Presentado en el congreso Edutec 2001 (<http://edutec.rediris.es>) y que se puede encontrar íntegramente en <http://apolo.umh.es/documentacion/articulo1/articulo1.doc>

^{IX} <http://web.uvic.ca/hrd/hotpot/>

^X <http://desarrollocc.umh.es>

^{XI} <http://www.linux.org>

^{XII} <http://www.apache.org>

^{XIII} <http://www.mysql.org>

^{XIV} <http://www.php.org>

^{XV} <http://apolo.umh.es>

^{XVI} Extended Markup Language

^{XVII} <http://www.halfbakedsoftware.com/>

^{XVIII} <http://www.w3.org/TR/xslt> Del W3C

^{XIX} <http://www.jclark.com/xml/>

^{XX} Pret Good Privacy <http://www.pgpi.org>

[VOLVER AL INDICE TEMAS](#)