BIGT. Banco de metaItems para la Generación de Test

Juan Carlos Rodríguez del Pino, Margarita Díaz Roca,
José Daniel González Domínguez, Zenón Hernández Figueroa
Dpto. de Informática y Sistemas
Universidad de Las Plamas de Gran Canaria
{jcrodriguez, mdiaz, dgonzalez, zhernandez}@dis.ulpgc.es

Resumen

En este trabajo se presenta un sistema de generación y evaluación de tests de forma automática. Se hace hincapié en la claridad y sencillez del esquema de representación de la información, y en minimizar el esfuerzo de creación de un gran número de items distintos. El núcleo del sistema lo constituye la definición de un DTD para representar bancos de metaitems a partir de los cuales se obtienen los items que constituyen los tests. Además, se ha desarrollado un programa informático que permite usar con facilidad los datos para generar tests, tanto para su aplicación en un ordenador como en papel, con la posibilidad de autoevaluación.

1. Introducción

La evaluación forma parte esencial del proceso de aprendizaje como instrumento para identificar las fortalezas y debilidades de la acción formativa y medir el grado de consecución de sus objetivos. Se distingue [3] entre: evaluación diagnóstica, que se realiza al inicio de la acción y sirve para determinar las condiciones de partida de los alumnos; la evaluación formativa, que se realiza durante el desarrollo de la acción con el fin de detectar sus desviaciones a tiempo de introducir medidas correctoras; y la evaluación sumativa, que se realiza al final de la acción para determinar sus resultados.

En el proceso de evaluación se pueden usar herramientas muy diversas, entre otras: exámenes de desarrollo, de preguntas cortas, tipo test, etc. De entre todas, los exámenes tipo test destacan por su objetividad y porque son susceptibles de un alto grado de informatización —otros tipos de prueba también admiten algún grado de informatización, que puede llegar a ser alto en algunos aspectos en casos como trabajos prácticos consistentes en la realización de programas de ordenador [14, 15, 16, 2], pero, probablemente, ninguna puede superar a los exámenes tipo test.

El esquema típico para la gestión mediante ordenadores de exámenes tipo test consiste en un banco de items del que se seleccionan algunos para generar cada test que se precise. Normalmente, un ítem está formado por un enunciado, una pregunta, una respuesta correcta y varias respuestas incorrectas o distractores, generalmente de dos a cuatro. El banco de items puede ser lineal o estar organizado en sub-bancos, y los items en sí pueden estar calibrados en función de parámetros tales como: dificultad, discriminación, pseudoacierto, etc. [13]. La selección de los items puede hacerse de forma totalmente aleatoria o por algún otro método, siendo especialmente interesantes los dirigidos a la generación de tests adaptativos. Los exámenes tipo test así generados pueden administrarse en papel o usando también el ordenador —vía lógica en el caso de los adaptativos.

Un factor de gran importancia para la seguridad del sistema es el número de items en el banco. La probabilidad de exposición de un ítem es función inversa de la cantidad de items disponible (bien del total, bien de los de la misma clase cuando los items están clasificados para su selección mediante algún tipo de método adaptativo). Si el número de items disponible es bajo, los items se repetirán con frecuencia, lo que abre la posibilidad de superar los tests por pura capacidad memorística, más que por auténtica destreza en el campo objeto de evaluación.

La investigación de algoritmos para la generación automática de ítems, dentro de la cual destacan los estudios sobre items isomorfos, ofrece interesantes expectativas, cara a facilitar la disponibilidad de un mayor número de items con un menor esfuerzo. Claro está, que el éxito de la generación automática de items está ligado a la solidez del modelo conceptual subyacente. Los mejores resultados se han obtenido en campos

acotados como las matemáticas y el análisis lógico [13].

Los autores de este trabajo se han propuesto desarrollar un sistema completo de gestión de tests basado en su generación y evaluación automáticas. Los tests se crean a partir de bancos de metaitems, que se conciben como elementos capaces de generar una gran cantidad de items diferentes con un mínimo esfuerzo. Lo que se expone en este trabajo es el desarrollo de los esquemas para la representación de la información de los metaitems con claridad y sencillez y su aplicación para la definición de tests con items variables. Este sistema se ha aplicado al ámbito de los lenguajes de programación, aunque no se contempla, ni en los esquemas, ni en los programas, ninguna restricción que impida su aplicación inmediata a otras disciplinas.

2. Fundamentos del sistema

Un test se genera seleccionando sus items a partir de un banco de items. En muchos casos, los items se incorporan al test tal cual están en el banco o, a lo sumo, se altera el orden de las alternativas. El aspecto clave del trabajo que aquí se expone es la sustitución del banco de items clásico por un banco de metaitems, concepto con el que se pretende designar una estructura de información pensada para servir de generatriz al mayor número posible de items distintos basados en un concepto unitario.

El banco está organizado en temas que agrupan metaitems relacionados (Figura 1). Esta asociación en temas permite una organización estructurada de los metaitems acorde a la distribución conceptual de la materia, permitiendo un adecuado nivel de discriminación a la hora de generar y seleccionar los items. Esta característica podría utilizarse con posterioridad para generar tests sobre un tema o con preguntas de determinados temas en particular.

El metaítem (Figura 2) está compuesto de: un enunciado (opcional), y dos conjuntos, uno formado por una pregunta relativa al enunciado y una o varias respuestas correctas para la misma y otro, formado por un conjunto de respuestas incorrectas que pueden ir acompañados de una pregunta opcional, también relativa al enunciado. La naturaleza de esta segunda pregunta deber ser tal que las respuestas del primer conjunto sean

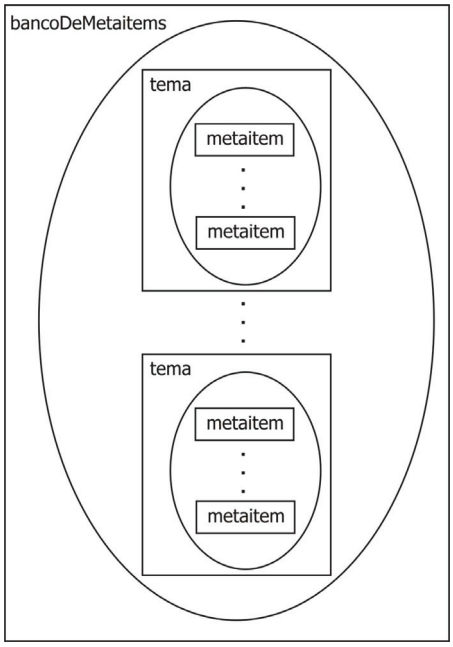

Figura 1. Estructura del banco de metaítems

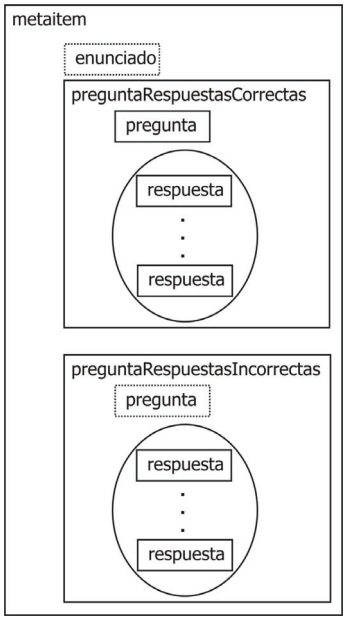

Figura 2. Estructura del metaítem

incorrectas para contestarla, mientras que cualquiera de las del segundo resulta adecuada.

Los dos conjuntos de repuestas tienen que ser disjuntos; de esta forma el sistema podrá generar numerosos items coherentes a partir de un metaítem sencillo.

Hay que tener en cuenta que es posible la existencia de una relación transversal entre respuestas correspondientes a cada uno de los dos conjuntos que las hagan incompatibles para aparecer juntas en un mismo ítem. La incompatibilidad estaría motivada por la presencia de distractores que aporten información relevante que facilite descubrir la respuesta correcta. Por tanto, durante la generación de un ítem se rechazan como respuestas erróneas las que pertenezcan al mismo grupo de incompatibilidad que la seleccionada como correcta.

2.1. Formato de representación

Con la finalidad de que el material que se elabore sea fácilmente trasladable a otros sistemas, el banco de items se escribe usando un formato XML (eXtensible Markup Language) [18]. El uso del XML requiere la definición de la representación aceptable para el banco de metaitems. En la especificación de esta representación se ha usado un Document Type Definition (DTD) [18]. El DTD (Tabla 1) es el núcleo del sistema ya que establece la forma que debe tener la información. Se ha desarrollado persiguiendo dos factores fundamentales: sencillez y claridad, con objeto de generar con facilidad el banco de items, además de mejorar su comprensibilidad y legibilidad por un lector humano.

La etiqueta *bancoDeMetaitems* hace de raíz del documento. La etiqueta *tema* permite definir un grupo de al menos un metaítem. Tanto, las etiquetas *bancoDeMetaitems* como *tema* tienen un atributo *título*, que se usa como identificador.

La etiqueta *metaítem* es la esencia del sistema. Está compuesta de un enunciado (opcional) y dos etiquetas *preguntaRespuestasCorrectas* y *preguntaRespuestasIncorrectas*. La etiqueta *preguntaRespuestasCorrectas* está formada por una etiqueta *pregunta* y una o más respuestas correctas. La etiqueta *preguntaRespuestasIncorrectas* está formada por una etiqueta *pregunta* (opcional) y una o más respuestas incorrectas. Las etiquetas *enunciado*, *pregunta* y *respuesta* son, en cierta forma, terminales ya que su contenido es su representación. Esta representación es textual aunque se ha añadido la posibilidad de usar las etiquetas XHTML [17] *br*, *i*, *b* y *pre* como elementos de presentación para un formateo más versátil que permite, por ejemplo, la inclusión de algoritmos y código de programas manteniendo un formato adecuado. Cada metaítem tiene un atributo identificador único.

Las respuestas asociadas a la etiqueta *preguntaRespuestasIncorrectas* constituyen los distractores del ítem construido con la pregunta asociada a la etiqueta *preguntaRespuestasCorrectas* y viceversa.

```
<!ELEMENT bancoDeMetaitems (tema*) >
<!ATTLIST bancoDeMetaitems título CDATA #REQUIRED>
<!ELEMENT tema (metaitem*) >
<!ATTLIST tema título CDATA #REQUIRED>
<!ELEMENT metaitem (enunciado?, preguntaRespuestasCorrectas,
                              preguntaRespuestasIncorrectas)>
<!ATTLIST metaitem identificador ID #REQUIRED>
<!ELEMENT preguntaRespuestasCorrectas (pregunta,respuesta+)>
<!ELEMENT preguntaRespuestasIncorrectas (pregunta?,respuesta+)>
<!ELEMENT enunciado (#PCDATA|b|i|pre|br)*>
<!ELEMENT pregunta  (#PCDATA|b|i|pre|br)*>
<!ELEMENT respuesta (#PCDATA|b|i|pre|br)*>
<!ATTLIST respuesta idIncompatibilidad CDATA #IMPLIED>
<!ELEMENT b (#PCDATA|br|i)*>
<!ELEMENT i (#PCDATA|br|b)*>
<!ELEMENT pre (#PCDATA|br|b|i)*>
<!ELEMENT br EMPTY>
```

Tabla 1. DTD del banco de metaitems

Las respuestas disponen de un atributo *idIncompatibilidad* que se emplea para formar grupos transversales entre ellas. Un grupo de respuestas incompatibles estará formado por elementos de los dos conjuntos de respuestas de un metaítem que tengan el mismo valor en el atributo *idIncompatibilidad*.

2.2. Comparativa con otros esquemas

La característica principal del DTD propuesto es que no tiene más de diez etiquetas relevantes, alcanzando el objetivo inicial buscado de simplicidad (Tabla 1). Nótese la diferencia con el estándar Question & Test Interoperability QTI 2.1 desarrollado por el IMS *Global Learning Consortium* [8] el cual dispone de más de 100 etiquetas, sin contar las de presentación. Esta sencillez no supone grandes restricciones en cuanto a la construcción del banco de items. La principal diferencia con respecto a otros planteamientos como el del estándar QTI es la aceptación de un único tipo de ítem de opción múltiple (multiple choice question) —otras formas de ítem pueden adaptarse a este formato sin pérdida de generalidad.

El esquema es válido para expresar items clásicos formados por una pregunta, una respuesta correcta y varias incorrectas (Tabla 2).

La simplicidad del esquema no merma las posibilidades de expresar muchas opciones que permitan una mayor variabilidad de items generables partiendo de un metaítem. En la versión lite 1.2 del estándar QTI los items son fijos y sólo se permite desordenar las distintas alternativas que se muestran al usuario.

3. Construcción del banco de metaitems

Aunque el sistema permite representar situaciones clásicas como se ha mostrado en la Tabla 2, su potencialidad se manifiesta al plantear metaitems con un gran número de respuestas válidas a cualquiera de las dos preguntas que pueden componerlos.

Para construir el banco de metaitems se pueden seguir las siguientes pautas:

- Cada tema a evaluar debe quedar reflejado en los temas del documento XML.
- Se debe elaborar una lista de objetivos detallados y seguirla, apoyándose en el material didáctico, de tal forma que cada apartado o concepto del tema quede plasmado en un metaítem.
- El enunciado del metaítem debe establecer el concepto sobre el que versa el ítem.
- El nivel de abstracción, tanto del enunciado como de la pregunta, debe ser tal que permita varias respuestas correctas.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE bancoDeMetaitems SYSTEM "bigt.dtd">
<bancoDeMetaitems título="Lenguaje de programación C">
  <tema título="Léxico">
    <metaitem identificador="tb_hexadecimal">
      <enunciado>
        Es posible escribir literales de un entero en hexadecimal.
      </enunciado>
      <preguntaRespuestasCorrectas>
        <pregunta>¿Indique cuál es el formato correcto?</pregunta>
        <respuesta>comienzan por 0x ó por 0X</respuesta>
      </preguntaRespuestasCorrectas>
      <preguntaRespuestasIncorrectas>
        <respuesta>comienzan por HEX</respuesta>
        <respuesta>comienzan por x0 ó por X0</respuesta>
        <respuesta>finalizan con x0 ó por X0</respuesta>
      </preguntaRespuestasIncorrectas>
    </metaitem>
  </tema>
</bancoDeMetaitems>
```

Tabla 2. Descripción XML de un ítem clásico

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE  bancoDeMetaitems SYSTEM "bigt.dtd">
<bancoDeMetaitems título="Lenguaje de programación C">
  <tema título="Léxico">
    <metaitem identificador="id3">
      <enunciado>Las palabras reservadas son aquellas que
                 no se pueden usar como identificadores.</enunciado>
      <preguntaRespuestasCorrectas>
        <pregunta>Indique cuál de los siguientes términos
                  <b>es una palabra reservada</b> en C</pregunta>
        <respuesta>auto</respuesta>
        <respuesta>extern</respuesta>
        <respuesta>short</respuesta>
        <respuesta>signed</respuesta>
        <respuesta>static</respuesta>
        <respuesta>default</respuesta>
        <respuesta>register</respuesta>
        <respuesta>continue</respuesta>
        <respuesta>break</respuesta>
        <respuesta>do</respuesta>
        <respuesta>case</respuesta>
        <respuesta>switch</respuesta>
      </preguntaRespuestasCorrectas>
      <preguntaRespuestasIncorrectas>
        <pregunta>Indique cuál de los siguientes términos
                  <b>no es una palabra reservada</b> en C</pregunta>
        <respuesta>import</respuesta>
        <respuesta>dynamic</respuesta>
        <respuesta>heap</respuesta>
        <respuesta>exit</respuesta>
        <respuesta>repeat</respuesta>
        <respuesta>fault</respuesta>
        <respuesta>until</respuesta>
        <respuesta>with</respuesta>
        <respuesta>record</respuesta>
        <respuesta>loop</respuesta>
        <respuesta>jump</respuesta>
        <respuesta>package</respuesta>
        <respuesta>use</respuesta>
      </preguntaRespuestasIncorrectas>
    </metaitem>
  </tema>
</bancoDeMetaitems>
```

Tabla 3. Ejemplo de metaítem

- Las respuestas asociadas a la etiqueta *preguntaRespuestasIncorrectas* deben ser alteraciones verosímiles de las respuestas correctas.

La Tabla 3 muestra un ejemplo de metaítem en el que se aplican estas pautas. Suponiendo items con una respuesta correcta y tres distractores, este metaítem, con 12 respuestas en el primer conjunto y 13 en el segundo, es capaz de generar 6292 items que difieren en al menos una respuesta.

4. Variabilidad

En base a la definición dada del banco de metaitems, el número de items distintos que se pueden generar depende de la cardinalidad de los dos conjuntos de respuestas de cada metaítem, sujeta a las restricciones debidas a la incompatibilidad, y del número de opciones a considerar en un ítem.

Suponiendo que se generan ítems con cuatro opciones [12] y que dos items son distintos si difieren al menos en una respuesta, si se denota por *nrc* al número de respuestas correctas de un metaítem y por *nre* al de incorrectas, para conjuntos de respuestas que no presentan ningún tipo de incompatibilidad el número de ítems distintos, *nid*, que se pueden generar a partir de un metaítem viene dado por la siguiente expresión:

$$\text{nid} = \text{nrc} * C^3_{\text{nre}} + \text{nre} * C^3_{\text{nrc}}$$

Para calcular el *nid* de un metaítem con respuestas incompatibles, se considera cada uno de sus dos conjuntos de respuestas formado por grupos de respuestas incompatibles. Si *ngrc* representa el número de grupos incompatibles de respuestas correctas y *ngre* el de incorrectas, y se denota por ngc_i al cardinal del i-ésimo grupo del conjunto de respuestas correctas y por nge_i al cardinal del i-ésimo grupo del conjunto de incorrectas, y se dispone de dos funciones tales que, para cada grupo de un conjunto, nos devuelve el número de respuestas compatibles del otro conjunto

$$\text{NCC(i)} = \begin{cases} \text{nre} & \text{si no existen incompatibles} \\ \text{nre} - \text{nge}_i & \text{si existen incompatibles} \end{cases}$$

$$\text{NCE(i)} = \begin{cases} \text{nrc} & \text{si no existen incompatibles} \\ \text{nrc} - \text{ngc}_i & \text{si existen incompatibles} \end{cases}$$

el número de ítems distintos que se pueden generar a partir de un metaítem es:

$$\text{nid} = \sum_{i=1}^{\text{ngrc}} \text{ngc}_i * C^3_{\text{NCC(i)}} + \sum_{j=1}^{\text{ngre}} \text{nge}_j * C^3_{\text{NCE(j)}}$$

Si se denota por nid_k a la cardinalidad del número de ítems distintos que se pueden generar a partir del k-ésimo metaítem. El número de ítems distintos, *nidbm*, que se pueden generar a partir de un banco de metaítems de cardinalidad *nbm* es:

$$\text{nidbm} = \sum_{k=1}^{\text{nbm}} \text{nid}_k$$

Si se considera que dos ítems son distintos cuando difieren al menos en una respuesta o cuando el orden de las respuestas es diferente, para conjuntos de respuestas que no presentan ningún tipo de incompatibilidad el número de ítems distintos considerando el orden, *nidco*, que se pueden generar a partir de un metaítem viene dado por la siguiente expresión:

$$\text{nidco} = 4 * \left(\text{nrc} * V^3_{\text{nre}} + \text{nre} * V^3_{\text{nrc}} \right)$$

Se pueden obtener expresiones análogas a las mostradas anteriormente para el número de items distintos que se pueden generar a partir de un metaítem con incompatibilidades y de un banco de metaitems al incorporar el orden como factor diferencial.

5. Aplicación para la generación de test

Se ha construido una aplicación desarrollada en Java [9] que procesa documentos XML con el DTD anterior. La aplicación tiene como entrada un banco de metaitems, pudiendo generar test autoevaluables en ordenador o imprimibles para evaluación manual.

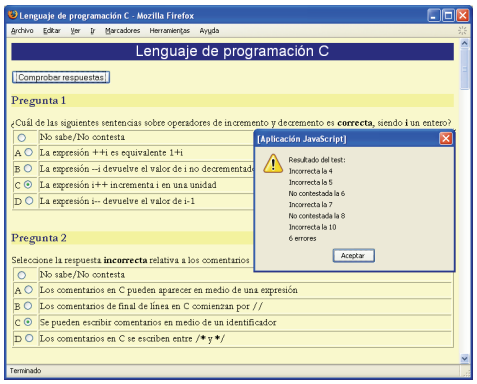

Figura 3. Ejemplo de test en HTML

Como parámetros generales de los test a generar, se puede indicar el número items por test y el número de opciones a elegir en cada uno (respuesta correcta más distractores) (Figura 4).
Para generar tests autoevaluables se emplea HTML [7] y javascript [5]. En este caso un test es un formulario en HTML con botones de tipo radio para cada ítem. Al pulsar el botón de enviar el

formulario las opciones seleccionadas son revisados por un código en javascript incrustado en el mismo documento HTML. El resultado de la revisión se muestra en pantalla (Figura 3).
Para realizar exámenes escritos el programa es capaz de generar un documento en formato rtf con una serie de test distintos añadiendo al final un solucionario. El formato rtf permite que los tests sean fácilmente personalizables, pudiéndose añadir encabezados, cambiar el tipo o tamaño de letra, etc. En la primera página de cada test aparece un recuadro donde el alumno debe dar las respuestas. Este recuadro encaja con el solucionario para facilitar la posterior revisión. Se puede llegar al extremo de generar un test distinto para cada alumno a evaluar.

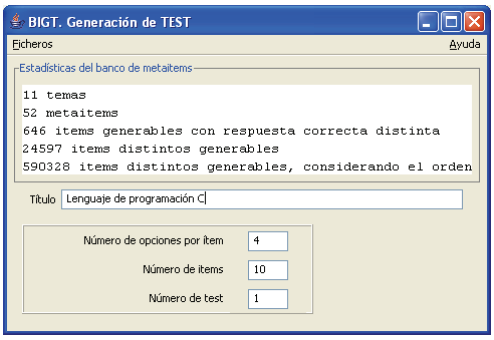

Figura 4. Estadísticas del banco de metaitems

Como experiencia de uso y campo de pruebas para el DTD se ha construido un banco de metaitems que permite generar tests sobre el lenguaje de programación C. Para editar el documento XML se empleó Eclipse 3.1 [4] con la extensión de herramientas webs (aunque existe un gran número de aplicaciones específicas que permiten la edición de documentos en este formato). El banco de metaitems se construyó empleando la metodología descrita anteriormente, resultando una base de 52 metaitems, capaces de generar 646 items de cuatro opciones con respuestas correctas distintas y 24.597 items distintos —difieren, al menos, en una respuesta— (Figura 4). La elaboración se realizó con cierta facilidad, debido, sobre todo, al seguimiento estricto del material didáctico disponible y al tipo de materia sobre la que versa el banco. Este banco se ha comenzado a emplear en la generación de tests que se utilizan como una parte de la evaluación de los alumnos de segundo curso de estudios universitarios en Informática. Para favorecer el conocimiento del nivel adquirido en esta materia, se ponen vía web a disposición de los alumnos tests autoevaluables con exactamente las mismas características que los que se van a emplear en la evaluación sumativa. Estos tests se componen de veinte items, cada uno con una respuesta correcta y tres distractores. La evaluación sumativa se realiza utilizando tests impresos en papel. La revisión de los tests realizados por parte de los profesores consume un tiempo medio estimado de 30 segundos por test.

6. Conclusiones y perspectivas futuras

Se desarrolla un sistema de generación y evaluación de tests de forma automática haciendo hincapié en la claridad y sencillez del esquema de representación de la información, y en minimizar el esfuerzo de creación de un gran número de items distintos. El núcleo del sistema lo constituye la definición de un DTD para representar bancos de metaitems a partir de los cuales se obtienen los items que constituyen los tests. Además, se ha desarrollado un programa escrito en Java que permite usar con facilidad los datos para generar tests, tanto para su aplicación en un ordenador como en papel, con la posibilidad de autoevaluación.

Como posibles líneas de evolución futura a corto plazo se tiene: el desarrollo de una aplicación que ofrezca una interfaz amigable, adecuada para la construcción de los bancos de metaitems; la posibilidad de incorporar gráficos e imágenes como parte de los enunciados y/o las respuestas; la exportación, tanto de los tests como del propio banco de metaitems, a otros formatos estandarizados como el QTI y su integración en otros sistemas de apoyo a la docencia como el moodle [11] o el GAP [14, 15, 16]. Muchos de estos sistemas posibilitan el uso de tests pero sin la versatilidad que aquí se muestra. Las ventajas de la integración consistirían en poder obtener resultados personalizados de las pruebas realizadas por cada alumno al automatizar tanto la evaluación diagnóstica como el desarrollo de la acción formativa y la evaluación sumativa final.
Otras posibilidades a más largo plazo entran dentro de la teoría de la respuesta al ítem [1, 6, 10]

y el calibrado automático de los items generados, que requieren el estudio de las respuestas dadas.

Referencias

[1] Baker, F. *The Basics of Item Response Theory*. Second edition Published by the ERIC Clearinghouse on Assessment and Evaluation, 2001

[2] Benford, S.; Burke, E.; Foxley, E.; Gutteridge, N. y Mohd Zin, A. *Ceilidh: A course administration and marking system*, Proceedings of the International Conference on Computer Based Learning in Science, Vienna, December 1993.

[3] Bordas, M.; Cabrera, F. *Estrategias De Evaluación De Los Aprendizajes Centrados En El Proceso*. Revista Española de Pedagogía. Año LIX, 2001, enero-abril, n.218. pp.25 a 48.

[4] Eclipse Project Release Build: 3.1. http://www.eclipse.org/eclipse/

[5] ECMAScript Language Specification. http://www.ecma-international.org/

[6] Hambleton, R.; Swaminathan, H. y Rogers, H. *Fundamentals of Item Response Theory*. SAGE, 1991

[7] HTML 4.01 Specification. W3C Recommendation 24 December 1999. http://www.w3.org/TR/html4/

[8] IMS Global Learning Consortium, Inc. Public Dispach. February 2005. http://www.imsglobal.org/

[9] Java 2 Platform Standard Edition 5.0 (J2SE 5.0). http://java.sun.com/j2se/1.5.0/

[10] Lord, F, M. *Applications Of Item Response Theory To Practical Testing Problems*. Lawrence Erlbaum Associates, 1980. Hillsdale. New Jersey (USA)

[11] MOODLE. Moodle is a course management system (CMS). http://moodle.org/

[12] Muñiz, J. *Introducción a la teoría de respuesta a los ítems*. Ediciones Pirámide Madrid. 1997

[13] Ponsoda Gil, V.; Hontangas, P.; Olea, J.; Revuelta, J.; Abad, F. y Ximénez, C. *Los tests adaptativos informatizados: investigación actual*. Metodología de las Ciencias del Comportamiento, Suplemento 2004, 505-510

[14] Rodríguez del Pino, J. C. *GAP. Gestión Automática de entrega de Prácticas vía web*. Ix Congreso Iberoamericano de Educación Superior en Computación. Mérida-Venezuela, Septiembre 2001.

[15] Rodríguez del Pino, J. C. 2002. *Gestión Automática de entrega de Prácticas vía web (GAP)*. Actas de las VIII Jornadas de enseñanza Universitaria de la Informática. Cáceres, Julio 2002, pp 53-58.

[16] Rodríguez del Pino, J. C.; Hernández Figueroa, Z.; González Domínguez, J. D.; Díaz Roca, M. *Experiencia de uso y características del programa Gestión Automática de Prácticas (GAP)*. Conferência IADIS Ibero-Americana. Lisboa, Octubre 2005, pp 35-42.

[17] XHTML™ 1.0 The Extensible HyperText Markup Language. http://www.w3.org/

[18] XML (Extensible Markup Language). http://www.w3.org/