

SISTEMAS DE EVALUACIÓN DE LOS ALUMNOS MEDIANTE TEST INFORMATIZADOS UTILIZAN- DO TELEFONÍA MÓVIL

Cristóbal Gálvez Espinar¹; Cristóbal Romero Morales²; Sebastián Ventura Soto³

UNIVERSIDAD DE CÓRDOBA

Departamento de Informática y Análisis Numérico

¹Alumno de Ingeniería Técnica Informática de Sistemas

²Profesor Colaborador

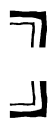
³Titular de Universidad

ESTUDIOS DE CALIDAD E INNOVACIÓN DE LA UNIVERSIDAD DE CÓRDOBA-AÑO 4º

RES NOVAE IV
CORDUBENSES



ÍNDICE



1. INTRODUCCIÓN

2. CREACIÓN DE TEST

3. EJECUCIÓN EN TELÉFONO MÓVIL

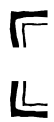
4. OPCIONES DE ADAPTABILIDAD

5. PRUEBAS DE UTILIZACIÓN

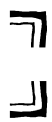
6. CONCLUSIONES Y FUTURAS MEJORAS

7. REFERENCIAS





TEXTO



SISTEMAS DE EVALUACIÓN DE LOS ALUMNOS MEDIANTE TEST INFORMATIZADOS UTILIZANDO TELEFONÍA MÓVIL

Cristóbal Gálvez Espinar (i12gaesc@uco.es)
Cristóbal Romero Morales ([cromero@uco.es](mailto: cromero@uco.es))
Sebastián Ventura Soto (sventura@uco.es)

RESUMEN

Este artículo presenta un sistema para la ejecución de test adaptativos en dispositivos móviles. Los test son generados en forma de ficheros XML a través de la herramienta autor Test Editor que se encuentra integrada en el sistema AHA! y son interpretados por nuestro sistema que puede mostrarlos a los usuarios en su dispositivo móvil, ya sea teléfono móvil o PDA. Nuestro motor de ejecución de test es un Midlet que es una pequeña aplicación Java orientada a ser ejecutada en dispositivos móviles. Esta aplicación permite realizar test clásicos y adaptativos de tipo actividad y de tipo examen. Si la prueba es de tipo examen, ésta sólo podrá ser ejecutada una sola vez, en cambio si es de tipo actividad se podrá ejecutar tantas veces como se quiera. Los estudiantes deben de identificarse con su login la primera vez que ejecuta el test, ya que el resto de veces éste queda almacenado en memoria RMS. En esta memoria también son almacenados los parámetros adaptables del interfaz, ya que los estudiantes podrán personalizar la configuración de el idioma, el tipo de fuente, el número de pantallas en que se muestran las preguntas, etc. Finalmente tras ejecutar un test los resultados obtenidos son mostrados a los estudiantes y son también enviados al servidor de AHA! a través de GPRS.

PALABRAS CLAVE: Test adaptativos, adaptabilidad, java, XML, telefonía móvil.

ABSTRACT

This paper describes a system for executing adaptive tests in mobile devices. Tests are generated in XML files using Test Editor that is a author tool integrated in AHA! system. These tests are interpreted by our system that can show to the user in their mobile devices such as mobile phone or PDA. Our test execution engine is a Midlet, that is a small Java application oriented to be executed in a mobile device. This application can execute classic or adaptive test of type activity (it can be executed all the times that students wants) or exam (it can be only executed one time). Students have to identify with their logins the first time that they execute a test. The next times is not necessary due to it is stored in the RMS memory. In this memory, we also store the interface adaptable parameters such as the language, the font type, the number of screens in which the question will be shown, etc. Finally, after executing a test the obtained results are shown to the students and they are also send it to the AHA! server by GPRS.

KEYWORDS: Adaptive tests, adaptability, java, XML, mobile telephony.



1. INTRODUCCIÓN

La educación a distancia ha sido uno de los sectores más beneficiados con el desarrollo de las nuevas tecnologías y especialmente con la expansión que han sufrido Internet y los dispositivos móviles. El aumento del uso de Internet y los dispositivos móviles como soporte para la educación a distancia hacen que se demanden nuevos modelos de enseñanza que se adecuen a este nuevo medio de comunicación y transmisión de información. Si no se adoptan nuevos modelos de educación se corre el riesgo de desaprovechar las ventajas que ofrecen estas nuevas tecnologías.

Nuestra aplicación está basada en un sistema existente actualmente llamado AHA! (Adaptative Hypermedia Authoring), el cual intenta convertir en adaptativo toda clase de aplicaciones basadas en la Web, a través un simple pero poderoso motor de adaptación. Por lo que dejando de lado los test clásicos o lineales, los cuales está formados por un conjunto de preguntas y respuestas preestablecidas, nos centramos en todo lo que sea adaptativo y adaptable y todo lo que rodea a ambos términos.

Un test Adaptativo Informatizado se caracteriza por su capacidad de adaptación al usuario, debiendo ser capaz de determinar con la mayor precisión y rapidez posible cuál es su estado cognitivo, es decir, qué partes del dominio que pretendemos enseñarle son las que domina y cuáles son las que aún desconoce.

Un test adaptativo intenta imitar automáticamente lo que haría un examinador humano especialista en la materia, de forma que si se propone una pregunta y resulta ser demasiado fácil, la siguiente debería de ser más difícil y viceversa. Con su uso se consigue evitar la frustración de los estudiantes principiantes (al no saber las preguntas difíciles) y el aburrimiento de los alumnos avanzados (al saber las preguntas fáciles).

La principal ventaja de los test adaptativos informatizados (TAI), es que cada examinado recibe diferentes preguntas y su número es normalmente más pequeño que el número de preguntas necesarias en un test clásico, y siempre dando una estimación del nivel de conocimiento al menos tan buena como la de los test tradicionales. Esta doble mejora es gracias al uso de los algoritmos adaptativos.



Quando hablamos de sistemas adaptativos y de sistemas adaptables tenemos que tener en cuenta que ambos tienen la misma funcionalidad, que no es otra que adaptar el sistema a las características del usuario (la información presentada y las funcionalidades ofrecidas). En cambio, la diferencia entre ambos reside en que mientras que en el sistema adaptable es el propio usuario el que tiene que proporcionar información para hacer la adaptación (opciones de configuración: idioma, tipo de fuente, conexión, número de pantallas, etc), en el adap-

tativo el sistema decide qué información o servicios presentar de acuerdo con el comportamiento que el usuario está teniendo (las propias preguntas del test).

Por otra parte, y como comentamos anteriormente cuando hacíamos referencia a la educación a distancia, hemos de resaltar que m-learning (mobile learning) y u-learning (ubiquitous learning) han comenzado a emerger como potentes entornos educativos. De hecho, hoy en día hay varios sistemas orientados a ser usados no sólo para usuarios de PC's, sino también para usuarios de teléfonos móviles y PDA's; y hay algunos test interactivos específicamente desarrollados para su uso en teléfonos móviles. Además hay varias herramientas comerciales tales como Mobile EMT-B quiz, orientadas a dispositivos PDA's, y otros tales como Go Test Go's orientadas a teléfonos móviles con tecnología Java. Siguiendo esta misma línea, desde el departamento de Informática de la Universidad de Córdoba hemos desarrollado la herramienta autor Test Editor que permite generar test computerizados que pueden ser ejecutados tanto desde un PC como desde una dispositivo móvil con Java.

2. CREACIÓN DE TEST

Para facilitar la creación de test informatizados así como su mantenimiento, hemos desarrollado la herramienta Test Editor (ver Figura 1). Actualmente, la herramienta se encuentra integrada dentro del sistema AHA! que es una arquitectura adaptativa hypermedia muy conocida y que permite construir cursos basados en la Web, y además usa lenguajes como Java y XML. En concreto, Test Editor es un Applet Java, al igual que otras herramientas autor del sistema AHA! como son: Form Editor, Concept Editor y Graph Editor.

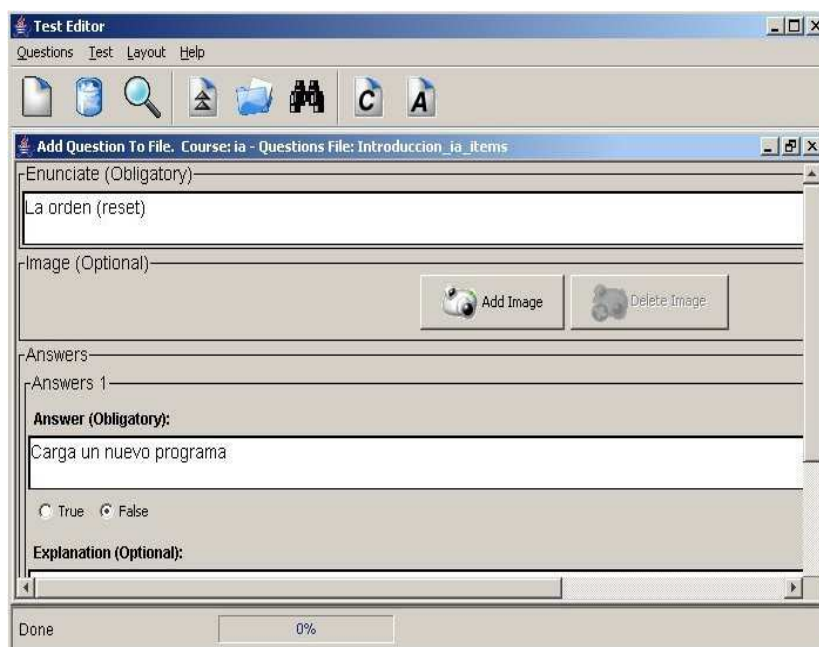


Figura. 1. Test Editor: Parámetros obligatorios de una pregunta.

Utilizando Test Editor, el profesor puede construir test de una manera fácil e intuitiva, con sólo ir rellenando los diferentes formularios sobre los distintos parámetros tanto obligatorios (ver Figura 1) como optativos del test. Finalmente cada test es almacenado en un fichero XML, aunque para dispositivos móviles además se genera un fichero .jar donde se almacena tanto dicho fichero XML con los datos del test creado (preguntas, respuestas, parámetros de ejecución, etc.), como las clases Java de la propia aplicación “móvil”. El test generado puede ser por tanto o bien descargado a un teléfono móvil o bien puede ser directamente usado (a través de un “navegador web”) en un curso del sistema AHA!.

Cuando un test se ejecuta dentro de AHA!, éste es presentado en un Applet (ver Figura 2) con un interfaz similar al que presenta la versión móvil. Nuestro motor de test basado en Web es un Applet de Java (ver Figura 2) que usa Servlets de Java para comunicarse con AHA!. Pueden ejecutarse test informatizados clásicos y adaptativos, con preguntas simples o multi-respuesta. Un test convencional (clásico) es una secuencia de preguntas y respuestas preestablecidas. El algoritmo que controla la ejecución de un test clásico es muy simple: muestra una secuencia de preguntas hasta que no hay más, o el examinado ha sobrepasado el tiempo máximo permitido.

Por otra parte, un TAI, es un test basado en PC donde la decisión sobre qué pregunta mostrar y cuándo finalizar el test depende del conocimiento del examinado en la pregunta anterior.

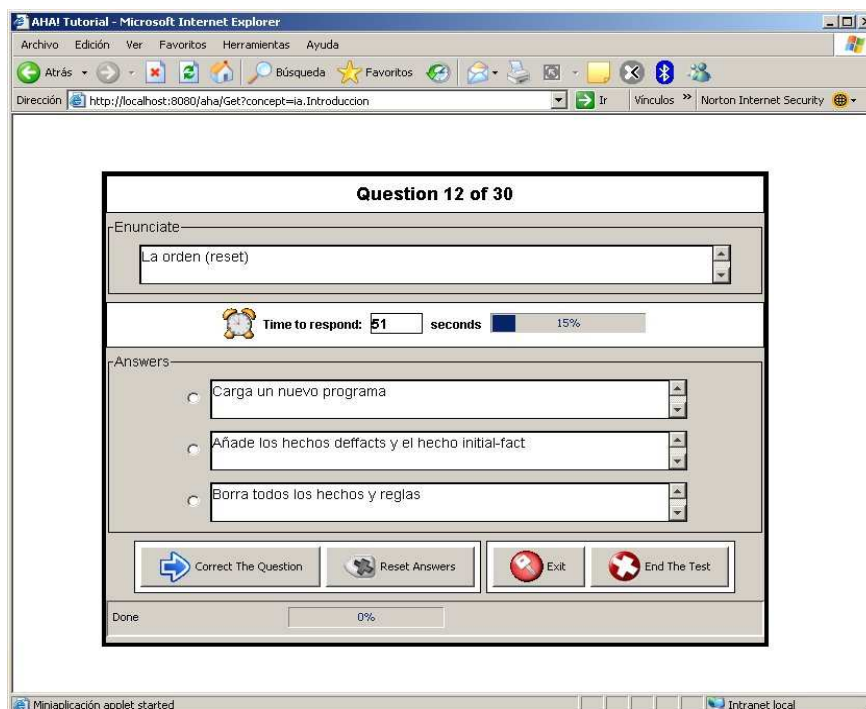


Figura. 2. Interfaz del motor de test basado en Web con una pregunta.

3. EJECUCIÓN EN TELÉFONO MÓVIL

Nuestro motor de test adaptable móvil es un Midlet de Java con una interfaz diseñada para pequeños dispositivos móviles. Los Midlets de Java son pequeñas aplicaciones que pueden ser ejecutadas en un teléfono móvil. Esto tiene ventajas importantes con respecto a la tecnología WAP (Wireless Application Protocol) y aplicaciones basadas en navegadores Web. Por ejemplo, se pueden ejecutar sin necesidad de estar permanentemente conectado a Internet, reduciendo el coste de conexión. Tiene además una interfaz interactiva y muy intuitiva, lo que permite un fácil manejo incluso para gente que no tenga conocimientos móviles o de informática.



El usuario puede descargar e instalar el fichero .jar (generado por el Test Editor) en el dispositivo móvil usando Bluetooth, Infrarrojos, cable serie, etc. Después de instalarlo, la ejecución del test es totalmente fuera de línea y trabaja como muestra en la Figura 3.

Funcionalmente, nuestro “motor móvil” puede leer los ficheros de test XML generados tal y como explicamos en el punto anterior. Presenta información inicial sobre el test, donde se especifica el concepto que se evalúa, el tiempo para cada pregunta, el número de preguntas que componen el test, y si las preguntas incorrectas o sin contestar tienen algún tipo de penalización. Además, muestra los enunciados de las preguntas con sus posibles respuestas, comprueba las respuestas elegidas, muestra una explicación de la respuesta elegida, muestra el resultado y puntuación final incluyendo una gráfica, etc., en definitiva todas las opciones que se pueden realizar en el sistema AHA!

Al tratarse de test adaptativos se establecen unos criterios de parada que vienen determinados por:

- la superación del tiempo máximo disponible para la ejecución del test,
- así como obtener un error menor al indicado
- o se llegue a mostrar el número total de preguntas sino se cumplen ninguno de los anteriores.

En ambos casos el test finalizaría y mostraría la puntuación en base a las preguntas contestadas. Una vez finalizado, ofrece desde el menú principal la posibilidad de enviar el resultado del test al sistema AHA! a través de GPRS (General Packet Radio Service).

A continuación vemos un ejemplo en el que se muestra una pregunta de un test y una gráfica que refleja los resultados de las respuestas, indicando el color verde el porcentaje de preguntas acertadas, el rojo de incorrectas, y el azul de preguntas sin contestar.

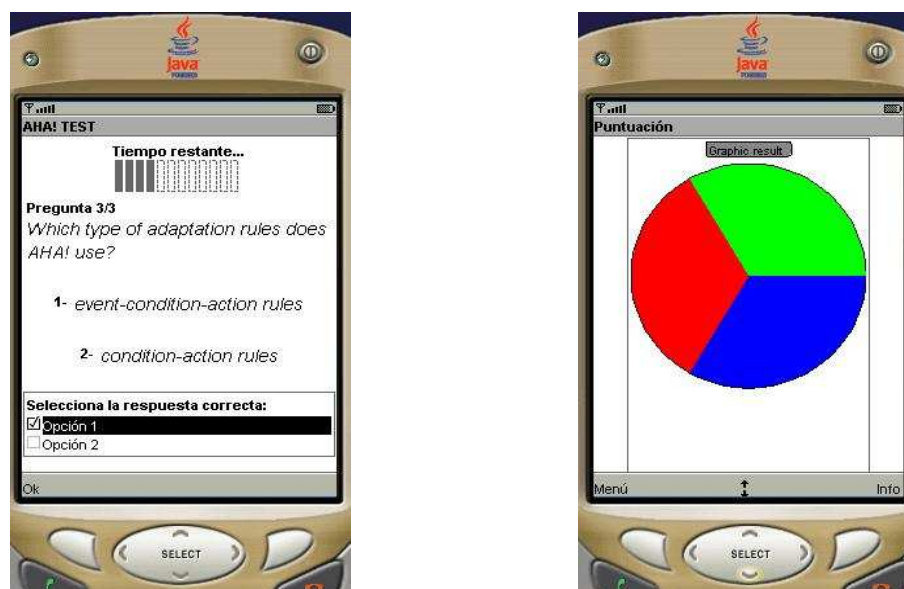


Figura 3. Enunciado, respuestas y opciones de una pregunta y gráfica de resultado.

El motor de test móvil tiene las siguientes características relacionadas con la individualización de la ejecución:

- Cuando el usuario ejecuta la aplicación por primera vez, éste debe identificarse introduciendo su nombre de usuario o login (los mismos que utiliza en AHA!). El resto de ocasiones que se acceda a esa misma aplicación todos los parámetros de configuración serán los mismos que la última vez que se abandonó la aplicación, quedando almacenados en memoria RMS (Record Management System) a nivel de login.
- A la hora de generar los test, se podrá decidir si se trabaja en modo actividad o en modo examen. Si es de tipo actividad el test podrá ejecutarse tantas veces como quiera el usuario, en cambio, si es de tipo examen, sólo podrá ser ejecutado una vez, aún saliendo y entrando de nuevo en la aplicación. En este caso nos mostrará un mensaje de información advirtiéndonos de que el test ya ha sido ejecutado, y únicamente podremos ver la puntuación y enviar los resultados. También a la hora de generar los test mediante la herramienta test-editor, se podrán configurar una serie de parámetros como el modo de ejecución (clásico o adaptativo), el número total de preguntas, los criterios de parada en caso de que la ejecución sea adaptativa (tiempo máximo y error Standard), etc, aunque todo esto corre a cargo del autor de

los cursos.

- Cuando el usuario finaliza el test, el resultado es mostrado en pantalla tanto en formato numérico como gráfico (ver Figura 3), además de ser físicamente almacenados en la memoria “permanente” del dispositivo usando RMS junto al resto de parámetros de configuración.
- Se permite ver la puntuación cada vez que se deseé desde el menú principal. Si el test ha sido ejecutado, mostrará la puntuación de la última vez que se ejecutó el test, ya fuera de tipo actividad o de tipo examen, en cambio, si desde que se instaló la aplicación no hemos realizado ningún test, nos mostrará un mensaje de información indicándonos que primero debemos ejecutar el test.
- El usuario puede enviar el resultado obtenido al sistema AHA! (para actualizar su perfil) a través de conexión GPRS (General Packet Radio Service).

4. OPCIONES DE ADAPTABILIDAD

Uno de los puntos fuertes de nuestro sistema son las características que lo convierten en adaptable. En este tipo de sistemas el usuario adquiere un papel principal, ya que es él mismo quien decide qué le interesa cambiar o cómo configurar las distintas opciones adaptándolas a sus necesidades. En nuestro sistema de test-móvil se muestra un claro ejemplo de adaptabilidad en las opciones que se presentan, tales como:

- **Adaptar Imágenes:** el usuario podrá decidir si quiere que las imágenes se adapten al tamaño de su pantalla. Esto evitará que imágenes con un gran tamaño que aparezcan en el enunciado de alguna pregunta, queden cortadas debido a unas reducidas dimensiones de la pantalla del dispositivo móvil, dificultando así la comprensión de la pregunta.

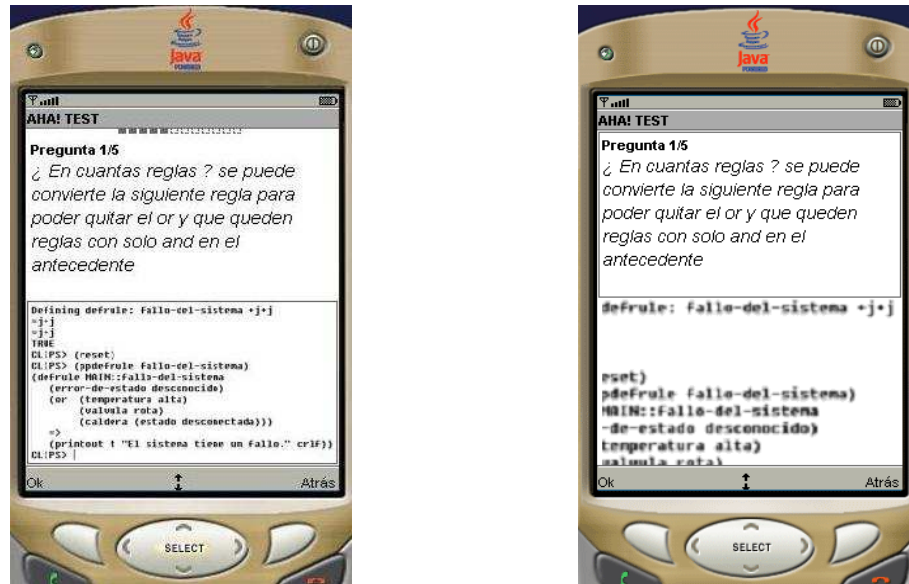


Figura 4.- Imagen adaptada en primer lugar, e imagen no adaptada

Podemos apreciar la diferencia, ya que como explicamos anteriormente la imagen queda cortada lo que lo convierte en una opción muy útil ya que algunos dispositivos de los usuarios pueden tener reducidas dimensiones.

- **Conexión:** el usuario podrá determinar la dirección del servidor y el puerto a través del cual se conecta para enviar los resultados del test. Si el usuario no modifica nada en esta opción, la dirección por defecto es 127.0.0.1 y el puerto de conexión 8080.
- **Cambiar fuente:** el usuario podrá elegir entre los distintos tipos de fuente preestablecidos, donde hay diferentes tipos y tamaños, para proporcionar una mejor legibilidad del texto en la pantalla del dispositivo.



Figura 5.- Tipos de fuentes y ejemplo de fuente número 4 e.

- **Número de pantallas:** el usuario podrá elegir si cada pregunta prefiere mostrarla en una sola pantalla e ir bajando a través de una barra o scroll, o bien dividirla en 2 o 3 pantallas e ir avanzando y retrocediendo por cada una de ellas a través de un botón de Ok y Atrás.

En el caso de que sean 2 pantallas, en la primera de ellas aparecerá el enunciado, y en la segunda las respuestas y las opciones correspondientes. En cambio, si el usuario elige mostrar cada pregunta en 3 pantallas, en la primera de ellas mostrará el enunciado, en la segunda las respuestas y en la tercera opciones.

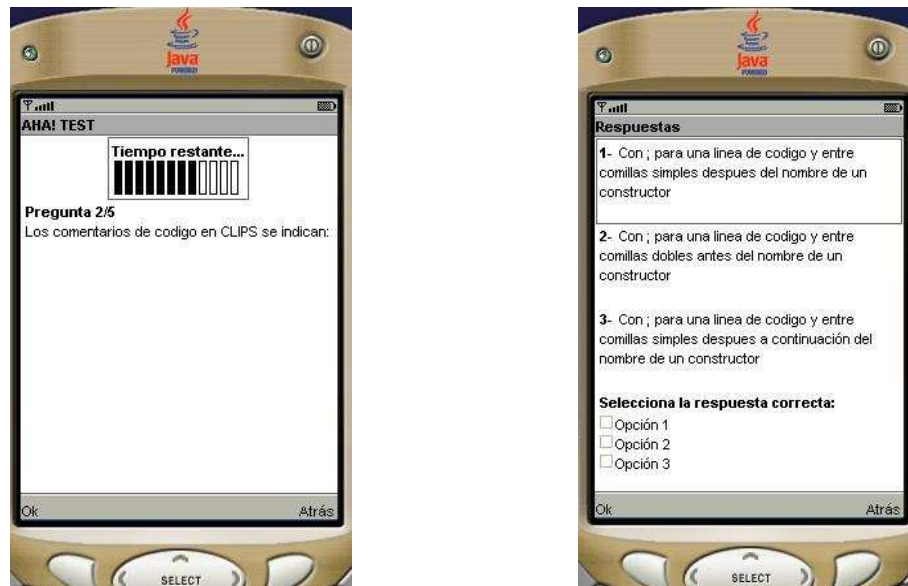


Figura 6.- Ejemplo de pregunta mostrada en 2 pantallas.

- **Idioma:** el usuario podrá seleccionar el idioma que se muestra en la interfaz, ya sea español o inglés.



Figura 7.- Ejemplo de Menú Principal en Español e Inglés.

5. PRUEBAS DE UTILIZACIÓN

Tras haber realizado durante varios meses pruebas típicas de detección de errores, actualmente se están llevando a cabo pruebas de funcionamiento real del sistema con alumnos de varias asignaturas de la Universidad de Córdoba. Un ejemplo concreto se utiliza en la asignatura de Informática Aplicada de 2º curso de Ingeniería Técnica Informática. La experiencia ha sido muy positiva, aunque se propuso como principal desventaja el tamaño de la pantalla, ya que la mayoría de los usuarios preferían ver las preguntas en una sola pantalla y sin necesidad de una barra de desplazamiento. Este problema además, será muy difícilmente solucionable debido a las dimensiones que presentan el tipo de dispositivos hacia el que va dirigido nuestra aplicación.

Con la información recogida en estas pruebas se pretende detectar nuevos y posibles errores no recogidos en fases anteriores y estudiar mejoras del sistema. Por otro lado, se desea conocer el interés que despierta este tipo de sistemas en nuestra Universidad a través de las opiniones de una pequeña representación de alumnos.

6. CONCLUSIONES Y FUTURAS MEJORAS

Se ha desarrollado una aplicación que cubre la ejecución de test adaptativos informatizados de tipo actividad y examen en un dispositivo móvil, manteniendo de forma independiente la información relativa a los parámetros de configuración del test y la propia información del test. Además permite la evaluación de los usuarios de los cursos incluidos en el sistema AHA!, mediante la ejecución de test adaptativos por parte de dichos usuarios, calculando los resultados de la ejecución y permitiendo enviarlos al sistema AHA! mediante GPRS.

Por otro lado, cabe destacar que la aplicación ha sido diseñada y desarrollada en módulos, lo que permite futuras ampliaciones de forma independiente en cada uno de ellos, al igual que la construcción de una interfaz gráfica sencilla, intuitiva y fácil de utilizar para todo tipo de usuarios.

Como futuras mejoras más necesitadas tal vez se encuentren la actualización/descarga de nuevas preguntas y exámenes desde la propia aplicación Java a través de GPRS, de forma que no se necesite descargar un nuevo .Jar, o establecer mejoras respecto a la seguridad y privacidad, utilizando algoritmos de encriptación en el caso en que sea necesario.

Después de la prueba realizada, la primera impresión es que los estudiantes están a favor del uso de tecnologías móviles en entornos educativos y además es posible el desarrollo de aplicaciones móviles educativas a pesar de algunas limitaciones de J2ME (Java 2 Micro Edition) como son el tamaño reducido de pantalla, el tamaño de la aplicación y el no soporte de números en punto flotante.

7. REFERENCIAS

ASSESSMENT SYSTEMS CORPORATION: Microcat and Fasttest, <http://www.assessment.com> (2006)

DE BRA, P., AERTS, A., BERDEN, B., DE LANGE, B., ROUSSEAU, B., SANTIC, T., SMITS, D., STASH, N.: AHA! The Adaptive Hypermedia Architecture. Proc. of the ACM Hypertext Conference (2003) 81-84

Go test go: <http://www.gotestgo.com> (2006)

GOH, T., KINSHUK, LIN, T.: Developing an adaptive mobile learning system. Proc of International Conference on Computers in Education, Hong Kong, (2003) 1062-1065

HALONEN, T.; ROMERO, J.; MELERO, J., GSM, GPRS and EDGE Performance: Evolution Towards 3G/UMTS. Wiley. (2003)

Juntao, Enterprise J2ME: Developing Mobile Java Applications. Prentice Hall PTR. Madrid. (2003)



KINSHUK: Adaptive Mobile Learning Technologies. GlobalEducator.com, (2003)

Mobile EMT-B Quiz: <http://www.emszone.com/mobilequiz> (2006)

YUAN, M.J.: Enterprise J2ME: Developing Mobile Java Applications. Prentice Hall, New Jersey (2003)