



Universidad de Jaén

Escuela Politécnica Superior (Jaén)

JUEGO EDUCATIVO ADAPTATIVO BASADOS EN PREGUNTAS TIPO TEST

Alumno: Cristóbal Jiménez López

Tutora: Prof. D. Macarena Espinilla Estévez

Dpto: Informática

Area: Arquitectura y Tecnología de los
Computadores



UNIVERSIDAD DE JAÉN

Escuela politécnica superior de Jaén
Departamento de informática

Dra. D^a . Macarena Espinilla Estévez, tutora del proyecto Fin de Carrera Titulado: Juego educativo adaptativo basado en preguntas tipos test, que presenta D. Cristóbal Jiménez López, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Diciembre 2012

El alumno:

El tutor/a:

Cristóbal Jiménez López

Macarena Espinilla Estévez

Capítulo 1	5
Introducción.....	5
Propósito.....	7
Objetivos	7
Estructura de la memoria.....	8
Capítulo 2	11
E-Learning, M-Learning y tecnologías.	11
E-Learning.....	11
Características del E-Learning	12
Ventajas e Inconvenientes del E-Learning.....	13
Módulo pedagógico E-Learning.....	15
Tecnologías y plataforma de E-Learning	17
M-Learning.....	24
Características del M-Learning	24
Ventajas e inconvenientes M-Learning.....	25
Módulo pedagógico del M-Learning	26
Capítulo 3	29
Plataformas para desarrollo de juegos educativos móvil.	29
Andengine	34
Librería Jdom	39
Android.....	39

Ada Framework.....	45
Capítulo 4.....	47
Descripción del Proyecto.....	47
Especificación del Software	50
Estudio de viabilidad	51
Presupuesto.....	51
Especificaciones de requerimientos	54
Especificación de requerimientos de un juego Educativo.....	54
Diseño de software	57
Módulo de casos de uso del sistema de auto- entrenamiento y auto- evaluación.....	58
Escenarios del sistema.....	63
Diseño de la interfaz de la aplicación.....	68
Diseño de los datos de la Aplicación Móvil.....	73
Normalización en el modelo Entidad-Relación.....	76
Implementación	82
Capítulo 5	84
Anexo. Manual de instalación y usuario	87
Manual de instalación.....	87
Manual de la aplicación.....	88
Anexo I. Empaquetación de una aplicación y firmar digitalmente.	94
Anexo II, Instalación de la plataforma ILIAS	100

Anexo III, Creación e exportación de preguntas desde la plataforma Ilias....	103
Bibliografía.....	108

Capítulo 1

Introducción

El rápido crecimiento de Internet y de los dispositivos electrónicos móviles como tabletas electrónicas, Smartphones, Notebooks, etc., están causando un gran impacto en la sociedad que, cada vez demanda más información útil, en cualquier momento y en cualquier sitio. Este desarrollo tecnológico ha propiciado también el desarrollo de las TICs permitiendo el consumo de información, sin ataduras, a través de un ordenador personal o de una red cableada. De igual modo, el crecimiento de las TICs ha favorecido el desarrollo de muchos sectores gracias, a la gran velocidad que hay actualmente de intercambio de información, permitiendo al usuario de estas tecnologías ser el receptor y el creador de información por eso, sectores como la publicidad, el ocio o la educación han experimentado una gran evolución

En consecuencia, tanto la Unión Europea, como España han propuesto una serie de medidas con las que se pretende difundir las Tecnologías de la Información y llevarlas a todos los ámbitos, a fin de que, en la medida de lo posible, el acceso a estas tecnologías sirva para disminuir la brecha digital.

Dentro del ámbito educativo, gracias a la difusión e incorporación de las Tecnologías de la Información existe la posibilidad de acceder nuevos métodos de enseñanza o vías, consiguiendo de esta manera que la enseñanza se adapte al alumno y no al contrario, facilitando además al alumno consumir los contenidos de la materia cuando él quiera y como él quiera. Debido a este hecho, actualmente nuevas formas de enseñanza, basadas en Internet, como el E-Learning están surgiendo y teniendo gran éxito.

A continuación, definamos en más profundidad la metodología de enseñanza E-Learning, como una educación a distancia cuya difusión se fundamenta en los canales de comunicación electrónicos y medios digitales (Internet, otras redes de comunicación o soportes digitales como pueden ser los DVDs), empleando para dicha difusión herramientas o aplicaciones, como software informático, plataformas de formación, correo electrónico, foros de discusión, etc. sirviendo a su vez, todos ellos como soporte para la enseñanza.

Dentro de la metodología de enseñanza del E-Learning existe una variante denominada M-Learning que se apoya en los dispositivos móviles por ejemplo: Tablet-Pc, tabletas electrónicas, PDAs, Smartphones, etc. Tales dispositivos disponen de una conectividad inalámbrica, permitiendo servir contenidos pedagógicos, con independencia del lugar y del momento y que pueda planificar así, su propio aprendizaje.

Por lo tanto, lo que permiten estas dos metodologías es facilitar el aprendizaje autónomo del estudiante para obtener unos conocimientos en función de su meta. Este aprendizaje autónomo se puede definir según Roger como “proceso personal de aprender a aprender, a cambiar, a adaptarse” y cuyas características son la responsabilidad y claridad en sus objetivos.

Además, estas metodologías ofrecen también herramientas para evaluar el nivel de conocimiento que ha alcanzado el alumno. En lo que se refiere a la evaluación del alumno existen varios métodos como pruebas escritas (exámenes, tipo test), pruebas verbales, evaluación continua, etc. De las formas de evaluación anteriormente citadas, este proyecto fin de carrera se centrará en las pruebas tipo test, ya que son herramientas

que permite medir el nivel de conocimiento de forma objetiva, además de ser fáciles de implementar, aunque más adelante veremos que también tienen sus desventajas.

Propósito

El propósito de este proyecto de fin de carrera es diseñar y desarrollar un recurso educativo que sea ameno, divertido y entretenido para reforzar conocimientos (autoentrenarse) y autoevaluarse. A continuación, enumeramos las principales características de nuestro recurso:

- Permitirá desarrollar el proceso de aprendizaje autónomo (auto-entrenamiento) y una auto-evaluación de un modo lúdico y entretenido, que facilite que los alumnos no abandonen el proceso de auto-aprendizaje.
- Se nutrirá de una base de datos de preguntas en formato SCORM [1]: Formato extendido en la creación de contenidos en plataformas de aprendizaje. El uso de SCORM supondrá una flexibilización de utilización en las diferentes plataformas .
- Nuestro recurso educativo será implementado en dispositivos móviles, lo que permitirá que el alumno lleve a cabo ambos procesos, en cualquier lugar y en cualquier momento, como marca la demandada metodología *M-Learning*.

Objetivos

Para alcanzar la propuesta, nos propondremos las siguientes metas:

1. Búsqueda y revisión de bibliografía.

2. Estudio y repercusión de las metodologías basadas en E-Learning y M-Learning.
3. Estudio de la plataforma Android y herramientas disponibles para desarrollar aplicaciones para dicha plataforma.
4. Análisis, diseño e implementación de un prototipo software implementado en el sistema operativo Android, para poner en marcha el recurso educativo que se nutra de las preguntas importadas de los módulos SCORM.

Estructura de la memoria

A continuación, haremos una breve introducción de los diferentes capítulos en lo que se estructura esta memoria y los contenidos planteados en los mismos.

El primer capítulo es una introducción donde se indica el propósito y los objetivos que busca el presente proyecto fin de carrera.

En el capítulo segundo, se hará una breve revisión de los conceptos citados en el capítulo de introducción es decir, los conceptos E-Learning y M-Learning que están íntimamente ligados, ya que son metodologías educativas basadas en las Tecnologías de la Información. La metodología E-Learning se sirve de cualquier medio electrónico para transmitir el recurso educativo y el M-Learning es una variante del E-Learning que tiene características similares, pero se utiliza en dispositivos móviles. Comentaremos más adelante sus ventajas y sus desventajas. También, se describirán las principales plataformas virtuales que existen en el mercado, se comentarán algunas de sus características y se explicarán los módulos SCORM que nos permiten el intercambio de contenido entre plataformas.

El capítulo 3 se describe varias plataformas de desarrollo móvil, se detallará dichas plataformas y se justificará que plataforma se utiliza para el desarrollo del

recurso educativo, además de todas las herramientas que se utilizaran en nuestro desarrollo.

El capítulo 4 describe nuestro proyecto en sí. Al tratarse de un proyecto de ingeniería de software se describirán cada una de las etapas de elaboración de nuestro recurso educativo. Este proceso de ingeniería de software consta de varias etapas detalladas en este capítulo. Las etapas indispensables que hemos considerado en nuestro proyecto son: el estudio de viabilidad, el diseño y la implementación.

El capítulo 5, expone una serie de conclusiones generales derivadas del proyecto.

Por último se incluyen una serie de anexos.

En el anexo I se detallará todo el proceso de instalación y utilización de dicho recurso educativo, en cualquier dispositivo móvil.

En el Anexo II se indicará como se exportará las preguntas en formato SCORM desde la plataforma Ilias hasta nuestro recurso educativo.

En el Anexo III, se indicará como empaquetar la aplicación para distribuirla y para poder utilizarla en cualquier tableta con Android

En el Anexo IV, se explicará cómo se instala la plataforma Ilias en un servidor de pruebas.

Capítulo 2

En este capítulo se profundiza en los conceptos básicos de E-Learning y M-Learning, las plataformas disponibles, los módulos SCORM y algunos conceptos asociados a las metodologías anteriormente citadas como el auto-entrenamiento o la auto-evaluación. Además, describiremos los sistemas que llevan los dispositivos móviles en la actualidad.

E-Learning, M-Learning y tecnologías.

E-Learning.

Las Tecnologías de la Información han llegado a todos los aspectos de nuestra vida y uno de ellos es el ámbito de la educación. Este ámbito se ha revolucionado por completo, gracias al avance de las tecnologías dando flexibilidad a la educación, para adaptarla de esta manera, más rápidamente a las necesidades de formación demandadas por la sociedad actual.

Son muchas las definiciones dadas de esta metodología, pero las definiciones que más se adaptan a nuestro proyecto son estas:

- *"Formación a través de Internet es un programa instruccional hipertexto que utiliza los atributos y recursos de Internet para crear ambientes de aprendizaje significativo. [2]*
- *"Integración de las tecnologías de la información y las comunicaciones en el ámbito educativo con el objeto de desarrollar cursos y otras actividades educativas sin que todos los participantes tengan que estar simultáneamente en el mismo lugar."*
(Universidad Politécnica de Madrid)"

- *"Es un sistema de impartición de formación a distancia, apoyado en las TIC (tecnologías, redes de telecomunicación, videoconferencias, TV digital, materiales multimedia), que combina distintos elementos pedagógicos: instrucción clásica (presencial o autoestudio), las prácticas, los contactos en tiempo real (presenciales, videoconferencias o chats) y los contactos diferidos (tutores, foros de debate, correo electrónico [3]"*

En general, se puede decir que es una metodología que se apoya en las Tecnologías de la Información, en las redes normalmente en Internet, en los recursos multimedia para ofrecer una educación o tutorización personalizada a distancia de forma flexible y autónoma por parte del alumno, con la ayuda de unos tutores o un tutor.

Características del E-Learning

Para ver las características de esta metodología la compararemos con las metodologías tradicionales:

- **Desaparecen barreras espacio-temporales.** Los alumnos no dependen ni de horarios, ni de sitios determinados, permitiendo maximizar el tiempo dedicado a la formación en cualquier lugar. Por el contrario en la educación tradicional se prefijan el espacio y el tiempo.
- **Formación Flexible.** Gracias a la gran diversidad de métodos y recursos empleados, se facilita el personalizar los contenidos para los alumnos según sus necesidades.
- **El alumno decide en todo momento.** El alumno puede elegir el camino formativo que este más en consonancia con sus metas.

- **El tutor cambia el rol.** El tutor pasa de ser un simple transmisor de contenidos a un tutor que orienta, guía, ayuda y facilita los procesos formativos.
- **Contenidos actualizados.** Las novedades y recursos relacionados con el tema se introducen de manera casi instantánea, de esta forma siempre están actualizados.
- **Comunicación constante entre los participantes.** Gracias a las herramientas que incorporan las plataformas E-Learning (foros, chat, correo-e, etc.), siempre tutor alumno establecen un canal de comunicación asíncrono, que en cualquier momento puede ser consultado.
- **Multimedia.** Permite la combinación de diferentes materiales (textos, auditivos visuales y audiovisuales). Las metodologías de enseñanza tradicionales parten de la base de que el sujeto recibe pasivamente el conocimiento, con esta nueva metodología se tratan de generar actitudes innovadoras, críticas e investigadoras.

Ventajas e Inconvenientes del E-Learning.

Las ventajas se pueden deducir de sus características, pero también tienen sus propias desventajas. Citaremos primero las ventajas:

- Gran disposición de recursos multimedia.
- Flexibilidad horaria.
- Favorece la interacción gracias a las herramientas que se proporcionan en esta metodología como foros, correos, blogs, etc.

- Aumenta el número de destinatarios gracias a su flexibilidad horaria, permitiendo al alumno compatibilizar la formación con sus otras actividades.
- Permite al alumno formarse cuando lo necesite (*Just in Time/just in case*).
- Este tipo de formación es más barata, accesible y adaptable a las necesidades del usuario.
- Facilita una formación grupal y colaborativa, gracias a todas sus herramientas de colaboración como wiki, blog, chat, etc.
- Facilita la reutilización de recursos educativos, cualquier recurso educativo se puede reutilizar en varios cursos.
- Ahorra costo y desplazamiento, ya que se puede dar en cualquier lugar.
- Permite la deslocalización del conocimiento.

A pesar de todas las ventajas, el E-Learning presenta, también grandes inconvenientes, entre ellos se señalan:

- La motivación es un factor importante ya que el alumno tiene que asumir la responsabilidad de su propio aprendizaje, si no se encuentra motivación o el curso no está bien diseñado los alumnos suelen optar por renunciar. Por eso con este proyecto intentamos que aprender sea ameno y divertido, para así potenciar la motivación y reducir el abandono.
- Que el alumno no se sienta cómodo con las herramientas informáticas, o tenga reticencia de usar las nuevas tecnologías.
- Requiere más trabajo por parte del profesor.

- Requiere que el alumno tenga habilidades para el aprendizaje autónomo.
- Requiere que los profesores estén formados para dar un entrenamiento adecuado y habilidades para dar soporte correctamente al alumnado.
- Requiere una conexión de internet rápida.
- Tiene profesorado poco formado.
- Supone problemas de seguridad y además de problemas de autenticación del estudiante.
- En ocasiones, existen algunas temáticas que son difíciles ser impartidas de forma presencial

Estos inconvenientes irán desapareciendo conforme se adquiera mayor experiencia en su utilización y su presencia sea más usual en nuestro sistema educativo. Por ello, es fundamental la labor de formación de los docentes para incorporar este tipo de metodologías.

Módulo pedagógico E-Learning

Un modelo pedagógico es aquel que sirve como un instrumento de la investigación de carácter teórico creado para reproducir idealmente el proceso enseñanza-aprendizaje.

El modelo pedagógico en E-Learning debería basarse en los siguientes principios.

- Aprendizaje activo.
- Aprendizaje colaborativo.
- Aprendizaje autónomo.

- Opciones variadas de interactividad.
- Comunicación sincrónica y asincrónica.
- Actividades o tareas relevantes y creativas.
- Evaluación continua.

A partir de estos modelos pedagógicos surgen los conceptos de auto-entrenamiento, que es la capacidad de adquirir conocimiento para superar un objetivo determinado y el otro concepto que surge es la auto-evaluación que se define como la capacidad del alumno de medir su progreso por su propia cuenta, con las herramientas proporcionadas en las diferentes metodologías.

A lo largo del tiempo para la auto-evaluación y el auto-entrenamiento se ha utilizado la herramienta de evaluación pregunta tipo test, por una serie de ventajas como son la rapidez de corrección y realización de este tipo de pruebas y precisión en la medida del nivel del alumno siempre que las pruebas estén bien diseñadas. Aunque la elaboración de estos tests sea una tarea complicada para el profesor, pero no más que otro tipo de pruebas ya que el ahorro de tiempo en la corrección compensa ampliamente, el tiempo invertido en la elaboración de la prueba. Por otra parte, la utilización de este tipo de herramientas a largo-medio plazo suele ser tediosa, como se han dicho anteriormente, por ello el tutor tiene que esmerarse en que el usuario no se aburra o abandone y por ese motivo nuestro proyecto va intentar tener una interfaz cuidada y amena para el usuario.

Vamos hablar de las diferentes plataformas ya que nuestro proyecto fin de carrera, vamos a realizar un recurso educativo basado en preguntas tipo test que se obtienen de dichas plataformas educativas, para potenciar el auto-evaluación y el auto-entrenamiento.

Tecnologías y plataforma de E-Learning

En E-Learning, las herramientas puestas en práctica han de cumplir varios requisitos para otorgar calidad al proceso de enseñanza y aprendizaje, entre otras muchas características, hablamos de flexibilidad y capacidad de adaptarse al cambio.

Las herramientas más comunes en la formación E-Learning, son las plataformas de teleformación a través de Internet. Estas herramientas pueden usarse en combinación con otras metodologías. Citaremos algunas de sus características básicas, y casi imprescindibles, que cualquier plataforma debería tener:

- Deben ser interactivas, que de la sensación al usuario de que es el centro de sus formación.
- El sistema E-Learning deben adaptarse a la necesidades de los alumnos su contenido pedagógico y estilo
- Tienen que ser un sistema escalable, es decir que funcione igual para cualquier número de personas.
- Tienen que ser una plataforma estándar, es decir que pueda utilizar recursos educativos desarrollados por terceras personas y además que esos recursos puedan utilizarse porque no haya ningún problema de compatibilidad.

Hay tres tipos de plataformas.

- CMS (Content Management System). Es un sistema que nos permite gestionar contenidos. En líneas generales, un CMS permitiría administrar contenidos en un medio digital y para el caso particular que nos ocupa, un CMS permitiría gestionar los contenidos de una web.

Algunos de los gestores más conocidos en la actualidad son Drupal, Wordpress, Joomla, etc.

- LMS (Learning Management System) Este tipo de plataforma está orientada a la formación, proporcionando herramientas para la gestión de contenidos académicos, permitiendo mejorar las competencias de los usuarios de los cursos y su intercomunicación, en un entorno donde es posible adaptar la formación a las exigencias de la organización. Además, disponen de herramientas que permiten la distribución de cursos, recursos, noticias y contenidos relacionados con la formación general.

LCMS (Learning Content Management System) Este tipo de plataforma mezcla características de las dos anteriores solucionando los problemas de las otras dos. Las plataformas educativas al ser tan variadas y heterogéneas facilitan el intercambio de recursos educativos entre plataformas, si se da la doble condición de que los recursos educativos sigan un estándar y de que la plataforma implemente ese estándar.

Hoy en día no existe un único estándar en el mercado, pero todos intentan solucionar el problema de la independencia del recurso de la plataforma. Algunos de los estándares más conocidos LOM de IEEE LTSC [4], ampliamente aceptados que permiten describir el recurso educativo a través de metadatos.

Otro estándar, ampliamente utilizado y aceptado, es el formato SCORM de ADL, que permite la organización de contenidos soportando la descripción de itinerarios formativos, secuencias de contenidos, el empaquetamiento de los contenidos

de aprendizaje para su cómoda distribución. Detallaremos este estándar ya que es el que implementado en este proyecto fin de carrera.

Un objeto de aprendizaje son pequeñas unidades de materia en un soporte digital por ejemplo flash, páginas electrónicas, multimedia, etc. La idea de estos contenidos es que cualquiera puede crear un objeto de aprendizaje y empaquetarlo en único fichero comprimido. Este fichero estará compuesto por:

- Los contenidos de aprendizaje, que es en sí el recurso
- El manifiesto, que contiene la información para poder ejecutar el objeto de aprendizaje correctamente, normalmente es un fichero de XML estandarizado que almacena metadatos.
- Hojas de estilos que permiten poder presentarlo adecuadamente en cualquier navegador.

Los paquetes SCORM tienen las siguientes características.

- Pueden ser utilizados en múltiples contextos y aplicaciones.
- Se pueden utilizar indistintamente en cualquier plataforma pedagógica.
- Son capaces de resistir a la evolución de la tecnología sin necesitar una reconcepción, reconfiguración o rescritura del código
- Tienen la capacidad de personalizar la formación según las necesidades de las personas y organizaciones.
- Se pueden acceder a los módulos desde cualquier sitio, así como distribuirlos.

A parte de las plataformas de formación, existen otras herramientas para desarrollar esta metodología como foros, correo electrónico, chat, agenda blogs, etc. que pueden estar implementados dentro de la propia plataforma.

A continuación vamos a describir algunas plataformas más significativas en el último año.

Blackboard

BlackBoard Inc. inicialmente fue una consultora que se fusionó con CourseInfo LLC, una pequeña compañía proveedora de programas de administración de cursos y luego se volvió a fusionar de nuevo con la competencia WebCT, finalmente la empresa resultante retuvo el nombre de BlackBoard. Su producto Blackboard Learning System es utilizado en las universidades de México e instituciones públicas.

El paquete Blackboard Learning System ofrece una amplia variedad de características para facilitar la creación de contenido profesional y de alta calidad. Algunas de estas características o herramientas son las siguientes:

- Los docentes pueden especificar criterios para la divulgación programada de contenido.
- Nombramientos y navegación controlador por el usuario y basadas en texto para áreas de contenido del curso.
- Unidades de Aprendizaje permiten a los instructores establecer un camino estructurado para el progreso.
- Mover y copiar contenido, archivos y Unidades de Aprendizaje dentro de las áreas de contenido del curso y entre cursos.

Moodle

Moodle es una plataforma educativa virtual, hace una buena gestión de cursos, es de distribución libre, ayuda a los educadores a crear comunidades de aprendizaje en línea. La plataforma es del tipo LMS.

Asimismo, promueve una pedagogía constructivista social (colaboración, actividades, reflexión crítica, etc.). Su arquitectura y herramientas son apropiadas para clases en línea, así como también para complementar el aprendizaje presencial. Algunas de estas herramientas son encuestas, foros, wikis, tareas, etc. Además de esto, tiene una interfaz de navegador sencilla, ligera, y compatible.

Aparte tiene buena interoperabilidad cuyas características son:

- Autenticación, usando LDAP, Shibboleth, o varios métodos estándar (por ejemplo IMAP)
- Inscripción, usando IMS Enterprise entre otros métodos, o por interacción directa con una base de datos externa.
- Concursos, y preguntas, permitiendo la importación y exportación en diversos formatos como GIFT (el formato propio de Moodle), IMS QTI, XML y XHTML (NB, aunque consigue exportar con buenos resultados, la importación no está completada actualmente).
- Recursos usando IMS Content Packaging, SCORM, AICC (CBT), LAMS
- Integración con otros sistemas de administración de contenidos, como PostNuke (a través de extensiones de terceros)

- Sindicalización usando noticias RSS o Atom (las noticias de fuentes externas pueden ser mostradas en un curso. Los foros, blogs y otras características pueden ser puestas a disposición de otros como noticias).
- Moodle también tiene características de importación para uso con otros sistemas específicos, como la importación de preguntas o cursos enteros para Blackboard o WebCT.

Dokeos

Dokeos es una plataforma del tipo LMCS. Es software libre y tiene licencia GNU GPL, el desarrollo es internacional y colaborativo. También está certificado por la OSI y puede ser usado como un sistema de gestión de contenido (CMS) para educación y educadores. Esta característica para administrar contenidos incluye distribución de contenidos, calendario, proceso de entrenamiento, chat en texto, audio y video, administración de pruebas y guardado de registros.

Algunas de sus herramientas son Herramientas disponibles

- Lecciones SCORM.
- Producción de documentos basados en plantillas.
- Ejercicios: opción múltiple, llenado de espacios en blanco, cotejar alternativas, preguntas abiertas, hotspots.
- Interacción: foros, chats y grupos.
- Videoconferencia: vía Web (manual de instalación removido en la versión Free 1.8.6, paquete siempre disponible públicamente para descarga).

- Conversión de presentaciones en PowerPoint e Impress a cursos en SCORM (manual de instalación removido en la versión Free 1.8.6, paquete siempre disponible públicamente para descarga).
- Herramientas colaborativas (Blogs, Encuestas, Red Social, etc.).
- Trabajos.
- Agenda.
- Anuncios.
- Glosario.
- Notas personales.
- Encuestas.
- Autenticación vía LDAP y OpenID.
- Evaluaciones.
- Reserva de matrícula.
- Sesiones de usuario.

Ilias

ILIAS son las siglas (Integriertes Lern-,Informations- und Arbeitskooperations-System) que en español se podría traducir como Sistema Integrado de Cooperación, Información aprendizaje que es una plataforma del tipo LMS. Además, está disponible de forma gratuita con licencia GPL (General Public License) con lo que puede ser fácilmente adaptado a las necesidades de la organización. Se apoya en el aprendizaje basado en contenidos (soportando el estándar de 2004).

Algunas de las características o herramientas son: Mensajería, Chat, Foro, Aprendizaje, wiki, control y gestión de cursos, escritorio personalizado (Noticias internas, Calendario, Notas Personales, etc.). Además de tener un potente módulo de

evaluación, que permite tipo test de elección múltiple, rellenar huecos, numérica, se ponga en venta, preguntas aleatorias, etc.

M-Learning

Características del *M-Learning*

Se podría definir el M-Learning como un submodelo pedagógico del E-Learning, ya que su filosofía es la misma pero aplicada a cualquier dispositivo móvil con capacidad de conectarse red (Smartphone, redes, etc.). Algunas definiciones de M-Learning.

- Según García el E-Learning es una práctica que nace de combinar los cursos basados en la Web con otros medios, tales como CD-ROMS, videos, clases satelitales y clases presenciales. Esto se hace porque, a pesar de las múltiples ventajas y la versatilidad de la Web, sus capacidades son limitadas, hecho que en ciertas oportunidades obliga al diseñador a acudir a diferentes medios y formatos para la distribución de los contenidos del curso. Desde este punto de vista, García define el Mobile Learning (M-Learning) como el uso de la Web junto con tecnología móvil. [4].
- Se denomina aprendizaje electrónico móvil, en inglés, M-Learning, a una metodología de enseñanza y aprendizaje valiéndose del uso de pequeños y maniobrables dispositivos móviles, tales como teléfonos móviles, agendas electrónicas, tablets PC, pocket pc, i-pods y todo dispositivo de mano que tenga alguna forma de conectividad inalámbrica. [5]

Ventajas e inconvenientes M-Learning

Ventajas e inconvenientes de M-Learning son parecidas al de E-Learning, algunas de ellas son:

- Aprendizaje *cualquier sitio y a cualquier hora*: Se puede aprender en cualquier lugar gracias a los dispositivos móviles, y su conectividad a las redes de estos dispositivos. En consecuencia, el contenido será accesible desde cualquier lugar del planeta.
- La interacción entre el usuario y el dispositivo es casi automática, con lo que lo que conlleva una mejor retroalimentación alumno-dispositivo para la comprensión correcta de los contenidos.
- *Tecnología más barata*: El coste de adquisición de un dispositivo móvil es relativamente más barata que la de un PC o un Notebook.
- Mayor *accesibilidad a los contenidos*. Todos estos dispositivos móviles podrían estar conectados a redes y servicios de acceso a Internet, garantizando el acceso a los contenidos en cualquier momento..
- Aprendizaje *colaborativo*. La tecnología móvil favorece que los alumnos puedan participar en determinadas actividades con distintos compañeros, creando grupos, compartiendo respuestas, etc.
- Los dispositivos móviles facilitan el *aprendizaje exploratorio*, el aprender sobre el terreno, explorando, experimentando y aplicando a la vez que se aprende la lección.

Algunas desventajas del M-Learning son las siguientes:

- Los dispositivos móviles pueden tener problemas asociados de usabilidad ya que tienen pantallas pequeñas; en general, por lo tanto la información mostrada en pantalla puede ser limitada y puede obligar al usuario a que se tenga ir desplazando a través del texto para leerlo. Por consiguiente, la aplicación debe tener una interfaz muy clara para no confundir al usuario.
- Los usuarios pueden ser reticentes a la hora de utilizar dichas tecnologías móviles para formarse.

Módulo pedagógico del M-Learning.

La utilización de herramientas informáticas para el apoyo a la educación y aprendizaje colaborativo ha propiciado el desarrollo del proceso enseñanza-aprendizaje.

Para que los alumnos se involucren en este proceso de enseñanza-aprendizaje, los entornos de aprendizajes tienen que ser entendidos por los alumnos, ricos en conocimientos, recursos variados, permisivos y amigables.

Existen algunas variables que influyen en el desempeño de la labor de los estudiantes tanto en los entornos presenciales, como virtuales. Algunas de estas variables pueden ser:

- La orientación motivacional intrínseca: Ésta se da si la actividad es una meta que quiere alcanzar el usuario y no como un medio para alcanzar el objetivo final. En este caso, puede que lo más probable es que el alumno motivado seleccione y realice actividades por el interés, curiosidad y desafío de que están provocan.
- La orientación motivacional extrínseca: Ésta lleva al individuo a realizar una determinada acción para satisfacer otros motivos que no

están directamente relacionados con la actividad, sino con la consecución de otras metas. Desde el punto de vista de la educación no presencial esto es lo que desmotiva al estudiante.

El uso de los dispositivos móviles en los procesos de enseñanza y aprendizaje presenta una serie de ventajas pedagógicas que se suman a otras ventajas operativas y que pueden ser las siguientes:

- Mejorar sus capacidades para leer, escribir y calcular, y reconocer sus capacidades existentes.
- Ayudar a combatir la resistencia a las TICs y a tender un puente sobre la brecha entre la alfabetización tecnológica a través del teléfono móvil y la realizada a través de las TIC.
- Ayudar a eliminar algo de la formalidad de la experiencia de aprendizaje e involucrar a estudiantes renuentes. Por ejemplo, es importante atraer a quienes están familiarizados desde la niñez con máquinas de juegos como PlayStations o Gameboys, aprovechando el conocimiento que se tiene de la tecnología. Ayudar a elevar la autoestima y proporcionar una sensación de confianza en la medida que se brinda a los docentes y estudiantes la responsabilidad del cuidado de dispositivos tecnológicos propios del M-Learning.

Os presentaremos las distintas plataformas de desarrollo móvil, ya que nuestro proyecto se apoya en la tecnología móvil y desarrollaremos nuestro recurso educativo en una de ellas. Nuestro recurso educativo es un juego educativo basado en preguntas tipo test, para fomentar la auto-evaluación y autoaprendizaje.

Capítulo 3

En este capítulo hablaremos sobre las distintas plataformas de desarrollo para los distintos dispositivos móviles. Entre las distintas plataformas de desarrollo móvil justificaremos que plataformas de desarrollo y herramientas hemos escogido.

Plataformas para desarrollo de juegos educativos móvil.

Expondremos algunas de las principales plataformas de desarrollo para móviles. Hay gran variedad de plataformas de las cuales veremos las más actuales como iOS, Symbian, Windows Phone y Android. A continuación escogeremos una para el desarrollo de nuestra aplicación argumentando las razones de nuestra elección.

iOS

Es un sistema operativo móvil de Apple, fue diseñado principalmente para los iPhones pero después se trasladó a dispositivos Móviles tanto como iPod Touch o iPad. La interfaz está muy conseguida se basa en el concepto de interacción hombre-dispositivo que representa el objeto de interés, sobre el cual se puede hacer una retroalimentación rápida y reversible facilitando el uso de la aplicación.

Algunas de las características del sistema operativo de Apple son:

- Solo está preparado para los dispositivo móviles de Apple, así que solo se puede adaptar a unas pocas resoluciones iPad o iPhone
- Está sujeta a la licencia APSL y Apple EULA.
- Sus librerías esta escritas C, C++, Objective-C
- No soporta flash, ni java.
- Conectividad 3G, Bluetooth o Wifi de alta velocidad.
- Api de desarrollo dirigido y mucho más específico.

Symbian

Es un sistema operativo que fue producto de la alianza de varias empresas de telefonía móvil, fue creado para competir con Windows Mobile. Este sistema operativo está basado en un microkernel con lo que se asegura robustez, disponibilidad y capacidad de repuestapilares básicos de este sistema operativo. Symbian tiene las siguientes características:

- Se programa en C++ estándar utilizando QT o Symbian C++ que tiene una curva de aprendizaje bastante alto, porque utilizan técnicas especiales, tales como descriptores, los objetos activos y la limpieza de pila.
- Es un sistema basado en ROM lo que permite ahorrar batería.
- No es multihilo.
- Conectividad 3G y wifi.
- Soporta múltiples resoluciones, sus componentes están hecho en C++.

Windows Phone

Windows Phone es un sistema operativo móvil desarrollado por Microsoft. Está diseñado para propósito general. Se basa en el núcleo del sistema operativo Windows CE y cuenta con un conjunto de aplicaciones básicas utilizando la API de Microsoft Windows. Se parece estéticamente a versiones de escritorio de Algunas de características de este sistema operativo son:

- Api incluida con .NET Compact Framework, Visual C++ o se puede implementar.
- Conectividad 3G/Wifi.
- Hardware variado.

Android

Android es un sistema operativo bastante avanzado pensado para teléfonos móviles, al igual que los distintos sistemas operativos IOS, BlackBerry, Symbian, etc. Con una interfaz amigable nos permite hacer mucho más que llamar, que nos lleguen mensajes, escuchar música o tomar fotos. Es una plataforma en la cual, se pueden instalar aplicaciones para hacer cualquier tipo de funcionalidad donde el único límite es nuestra imaginación.

Otra característica interesante es que está basada en el núcleo de Linux que tiene como licencia GPL V2, también la mayor parte del código está bajo la licencia de Apache, que es una licencia libre y de código abierto.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework java sobre el núcleo de las bibliotecas java en una máquina Dalvik, que se compila de manera dinámica. Las bibliotecas están escritas en C en las cuales, se incluye un administrador de interfaz gráfica, una base de datos relacional SQLite, una Api para programación gráfica (OpenGL ES 2.0 3D), un motor de renderizado y un motor gráfico SGL.

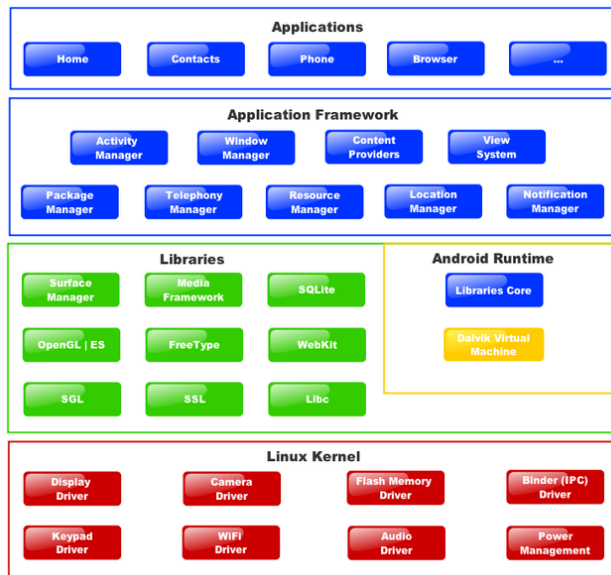


Ilustración 1. Arquitectura de Android

Algunas de las características de la plataforma de desarrollo Android son:

- La plataforma es adaptable a cualquier tipo de pantalla, tanto para un teléfono, una tableta digital o una televisión.
- Se puede almacenar datos en una base de datos embebida SQLite.
- Soporte mensajería SMS o MMS.
- Navegación web con motor renderizado de WebKit.
- Soporte multimedia.
- Videollamada.
- Multitarea.
- Tethering compartir conexión.
- Conectividad vía 3G o wifi.

Para conseguir nuestro objetivo, en este proyecto fin de carrera utilizaremos Android por los siguientes motivos.

Posee una cuota de mercado bastante amplia según un estudio realizado por International Data Corporation (IDC) del primer trimestre de 2012. Además, Android posee una cuota de mercado del 59% con 89.9 millones de teléfonos vendidos, seguidos de sus competidores Apple con Ios con 23%, BlackBerry con 6.4% y el resto otros sistemas como Linux o Windows/Mobile.

Otro motivo es porque este sistema operativo posee una gran cantidad de recursos como una documentación muy extensa, ejemplos, recursos online, una cantidad de frameworks para tareas específicas como framework Andengine (motor de OpenGL para realización) o framework Adaframework (gestión de bases de datos SQLite).

Otra ventaja de utilizar este sistema operativo es que ofrece herramientas potentes de programación como el simulador, el registro de errores que ayuda con la depuración con LogCat y el lenguaje programación que se basa en Java.

Finalmente señalaremos que cada vez hay más dispositivos que se utilizan con el sistema operativo de Google, actualmente existen más de 190 millones de dispositivos

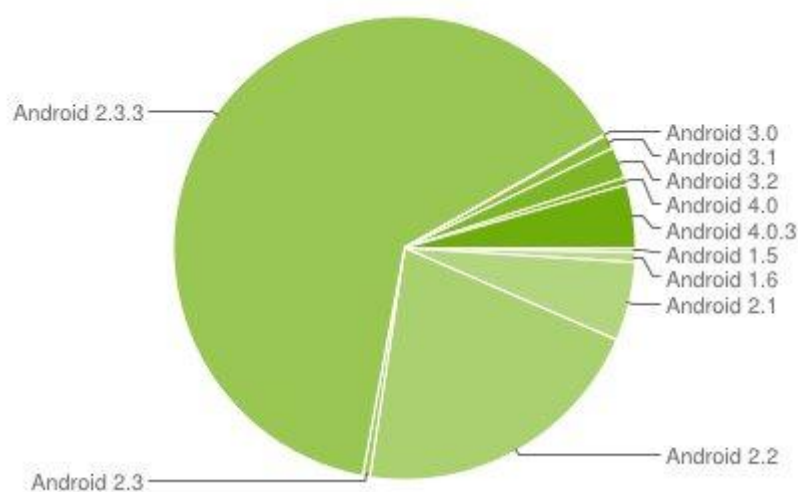


Ilustración 2. Cuota de mercado de las diferentes versiones de Android

No todo son ventajas hay un grave problema que acucia al sistema operativo de Google que es la fragmentación. Como se puede observar en la gráfica hay diferentes versiones de Android que conllevan el problema problemas de rendimiento y de compatibilidad, porque todas las aplicaciones de Android más antiguo tienen que ejecutarse en la última versión, con lo que puede llevar una pérdida de rendimiento por mantener la compatibilidad. Asimismo se puede dar la situación contraria que se haga para una versión moderna que aprovecharía los recursos de los dispositivos pero no sería compatible con otras versiones más antiguas. Por eso vamos a desarrollar nuestra aplicación para Android 2.3.3 para llegar a más usuarios y no tener tanta pérdida de rendimiento.

Andengine

Una vez decidida la temática del proyecto, el primer paso que se debía llevar a cabo era recolectar toda la información posible sobre el desarrollo de videojuegos para la plataforma Android. Después de un proceso de investigación, llegamos a la conclusión de que había dos formas de hacerlo, o bien usando las API's nativas de Google o bien usando algunas API's externas.

Llevar a cabo el desarrollo mediante el uso de las API's nativas de Android, implicaba un gran número de horas de trabajo e investigación, sin saber que el resultado sería el deseable, dado que su complejidad es elevada y la documentación al respecto escasa.

El uso de API's externas facilitaba la tarea, ya que la documentación era mucho más extensa y el grado de complejidad de éstas, se reducía de forma significativa. De forma que se decide hacer uso de alguna API orientada al diseño de videojuegos.

Dada la gran cantidad de alternativas, una primera selección la hicimos en función de la actividad del proyecto y de su madurez. Las opciones que más atractivas nos parecieron AndEngine y Rokon.

	Rokon2	AndEngine
FPS (one sprite)	52	58
FPS (50 physics)	46	60
FPS (100 physics)	42	36
Physics Engine	box2d	box2d
Particle System	No	Yes
License	BSD	LGPL
Examples	?	Yes
Game Implement	Tetronimo Drop Block	Snake AlienStars

Ilustración 3 Comparativa entre Rokon2 y AndEngine

Comparando el rendimiento y las características de ambas, podemos observar que AndEngine puede sufrir una caída de rendimiento en función de los objetos del juego algo que Rokon parece que es capaz de mantener constante, aunque para nuestro propósito ambos serían válidos.

Los dos frameworks utilizan el motor de física box2d, algo fundamental en nuestro caso ya que tanto la versión Android como iOS deben usar el mismo motor, para que los juegos sean idénticos en comportamiento, y nos resulte más sencillo calibrarlos.

Era un requisito indispensable que la API elegida fuera OpenSource, por lo que después de una ardua tarea de investigación, la escogida fue la librería AndEngine [6], ya que cumplía con todos los requisitos previamente establecidos, nos proporcionaba gran cantidad de funcionalidades relacionadas directamente con el desarrollo de videojuegos, y sobre todo es de código libre.

En los siguientes apartados detallamos las principales funcionalidades que nos ofrece esta librería.

Actividad Principal

Es parte básica del desarrollo de cualquier juego, una actividad que se encarga de gestionar todo el ciclo de vida del juego, sistema de puntuación, temporización en caso de ser necesaria, avance de los mapas, etc., en resumen todo aquello que hace que la experiencia de juego del usuario goce de cierta fluidez.

Es en este punto en el que usamos la llamada BaseGameActivity del AndEngine, que contiene las clases principales del flujo de juego. Para poder controlar este flujo, debemos crear una actividad que derive de BaseGameActivity y extender la implementación de sus métodos principales para adaptarlos a nuestras necesidades.

Siendo algunos de los eventos más típicos que deberemos controlar, el arranque del juego, la pantalla de inicio, la selección de nivel, personajes y demás configuraciones, en caso de que éstas puedan llegar a existir, pausar el juego, etc.

Motor

El motor es la pieza encargada de hacer que el juego pueda llegar a funcionar. Dispone de un hilo de ejecución que refresca la pantalla, es decir aquello que el usuario está viendo, cada un intervalo de ciertos milisegundos, previamente definidos por el programador del juego.

En cada tick se encarga de sincronizar los refrescos de la pantalla, con la nueva situación de todos los elementos que forma parte de ella, y en definitiva de actualizar la escena.

Escenas

La escena es el contenedor de todos y cada uno de los sprites que se deben visualizar por pantalla, y para esto AndEngine nos facilita una clase, con una serie de métodos base que debemos completar para adaptarlos a las necesidades de nuestro juego.

Las escenas pueden estar formadas por diferentes capas y entidades, en las cuales podemos diferenciar entre mapas, sistemas de puntuación e información al jugador, objetos con los que interactuar, personajes del juego, etc.

Texturas

Las texturas son imágenes en memoria. Estas imágenes serán utilizadas para visualizar sprites o fondos, con los que componer la escena de nuestro juego. Es por esto que en el momento en el que se inicia el desarrollo de un juego, por una parte se gestiona la información lógica de los objetos y por otra parte el aspecto visual que finalmente tendrá, o lo que es lo mismo, las texturas que mostraremos.

Sprites

Los sprites son objetos que se visualizarán en el juego y en la mayoría de los casos serán interactivos o incluso animados. Existen diversos tipos de sprites, tiles en los cuales los juegos tienen forma de matriz, animados en los que el sprite se compone de varias imágenes o fotogramas, que se irán intercambiando a lo largo de la acción del juego, para simular un movimiento o animación, o background los que serán usados para dibujar fondos.

Físicas y Colisiones

Adicionalmente, disponemos de las funcionalidades necesarias para detectar colisiones e incluso aplicarles algunos efectos de física, mediante los cuales simulamos movimientos provocados por la interacción de diferentes elementos del juego, como serían

el no poder atravesar una pared, o detectar que una bala ha impactado en un enemigo, provocando así nuestra victoria, consiguiendo así comportamientos mucho más reales de los objetos durante la acción del juego.

Música y Efectos de Sonido

Otra característica de este framework es la posibilidad de reproducir algunos tipos de ficheros de sonido. Permittiéndonos, de esta manera controlar ciertas características de la reproducción como serían subir y bajar el volumen, repetir el sonido, saber si está siendo reproducido en un momento concreto, escoger un punto concreto de la pista de audio, entre otras funciones adicionales.

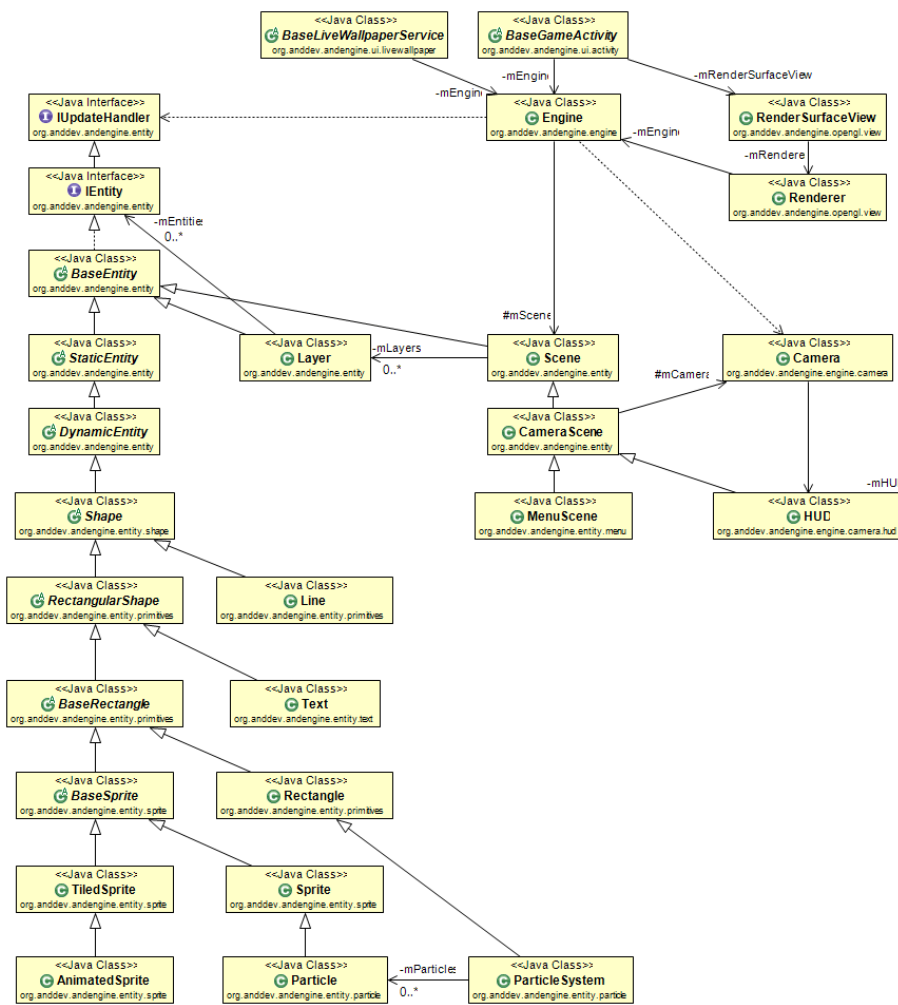


Ilustración 4 Jerarquía de clases de Andengine

Librería Jdom

JDOM es una API desarrollada específicamente para Java y da soporte al tratamiento de XML: parseo, búsquedas, modificación, generación y serialización. Es un modelo similar a DOM, pero no está creado ni modelado sobre DOM. Se trata de un modelo alternativo. La principal diferencia es que DOM fue desarrollado para que fuera independiente del lenguaje, mientras que JDOM está creado y optimizado específicamente para Java. Esto imprime a JDOM las ventajas inherentes a Java, dando lugar a una API más eficiente y más natural de usar para el desarrollador Java y por tanto, requiere un menor coste de aprendizaje.

JDOM genera un árbol de nodos al parsear un documento XML. En cuanto a los tipos de nodos, son similares a los de DOM, aunque algunos cambian el nombre ligeramente.

Android

Android es un sistema operativo diseñado por capas. Utiliza el kernel de Linux 2.6 que le proporciona el acceso a la parte hardware de los dispositivos. A continuación, se muestra un diagrama donde se puede observar la arquitectura que se desarrolla, posteriormente en los siguientes apartados se detallará y explicará en qué consiste cada una de las capas.

A continuación, se detallan cada una de las capas que constituyen el sistema Android:

Aplicaciones

Forman la capa superior, esta capa la controla plenamente el usuario. Está formada por las aplicaciones que vienen instaladas de base y por aquellas aplicaciones que el usuario desee instalar. Todas las aplicaciones están escritas en el lenguaje de programación Java.

Framework de aplicaciones

El marco de aplicaciones da acceso completo a los programadores a las mismas APIs utilizadas por las aplicaciones básicas. La arquitectura está diseñada para facilitar el rehúso de los componentes; cualquier aplicación puede publicar sus capacidades con el objetivo de que, otra aplicación pueda más tarde hacer uso de ellas. Esto se permite siempre y cuando la seguridad impuesta por el framework no se vea comprometida. En el gráfico anterior se pueden observar las librerías más importantes de esta capa.

Bibliotecas

El sistema incluye un set de librerías en C y C++ que proporcionan la mayor parte de las funcionalidades y que son utilizadas por varios componentes del sistema. Estas capacidades se exponen a los desarrolladores a través del framework de aplicaciones de Android.

Entre las librerías más importantes destacan:

- System C library: Librería estándar de C, optimizada para dispositivos móviles.
- SGL: Motor gráfico 2D.
- 3D Libraries: Esta librería utiliza hardware para la aceleración 3D.
- SQLite: Sistema gestor de base de datos.
- FreeType: Librería para fuentes de texto.

AndroidRunTime

Constituye el entorno de ejecución, compuesto por las librerías y la máquina virtual. Cada vez que se ejecuta una aplicación crea su propio proceso con su propia instancia de la máquina virtual Dalvik, que ha sido desarrollada para poder albergar varios procesos a la vez.

Kernel de Linux

Constituye la capa más próxima al hardware del dispositivo. Basado en el kernel de Linux 2.6, este kernel es utilizado para abstraerse del hardware de cada terminal, conteniendo los driver necesarios para poder comunicarse con el dispositivo. El sistema operativo es el encargado de gestionar los servicios del sistema como seguridad, gestión de memoria, de procesos, la pila de red, etcétera, a través de este módulo.

Componentes

Durante este apartado se van a describir los diferentes componentes que puede tener una aplicación Android. Dichos componentes pueden colaborar entre ellos de cualquier manera, no siendo necesaria la implementación obligatoria de alguno de ellos, para el correcto funcionamiento de la aplicación.

Activity

El componente Activity es considerado el componente principal de la interfaz gráfica de una aplicación Android. Los Activity suelen ir asociados con al menos un layout y entre ambos crean la interfaz visual de la aplicación. Para implementar una clase Activity debe heredar de la clase base Activity. Este componente lleva asociada la interfaz de usuario, representada por la clase View y sus derivados.

La mayoría de las aplicaciones disponen de varias pantallas, esto se suele conseguir usando varias Activitys. Cuando el usuario pasa de una pantalla a otra, la primera actividad queda en estado de pausa dentro de la pila. Estos aspectos serán detallados más adelante en el apartado “Ciclo de vida de una aplicación Android”.

Los Intent están íntimamente relacionados con las Activitys, se consideran un objeto mensaje que describe que quiere hacer la aplicación. Un Intent tiene dos partes básicas:

- La Acción que se desea realizar en ese momento.
- Los parámetros que se le otorgan al objeto por medio de la clase Bundle.

Las principales capacidades que desarrollan los objetos Intent son: Solicitar información de un contacto, escribir un mensaje de texto, realizar una llamada o abrir una aplicación.

Otra clase relacionada con los Intent son los Listeners, estos objetos son utilizados para reaccionar a eventos externos (por ejemplo, tocar la pantalla, una llamada, recibir un mensaje, etcétera).

Service

Un Servicio está formado por un código que se ejecuta durante largo tiempo en segundo plano y sin necesidad de interfaz gráfica. Los servicios pueden realizar cualquier tipo de acción que requiera la aplicación como lanzar notificaciones, mostrar eventos, actualizar la base de datos.

Content provider

Las aplicaciones desarrolladas para Android pueden compartir información con otras aplicaciones utilizando el componente Content Provider. Consiste en una clase que

implementa un conjunto estándar de métodos que permite a otras aplicaciones guardar y obtener la información que maneja dicho Content Provider.

Dicho de otro modo, este servicio nos permite compartir datos de nuestra aplicación sin mostrar detalles sobre su almacenamiento interno o su estructura.

Broadcast receiver

Es el componente encargado en detectar llamadas o mensajes de otros servicios generados por el sistema. Dichos servicios pueden ser recibir una llamada, un e-mail, sincronización de datos, actualización disponible, etc.

Este componente a diferencia de las Activity no tiene una interfaz gráfica asociada. Pero pueden utilizar el API Notification Manager para comunicar al usuario que se ha producido una notificación en el sistema.

Ciclo de vida

Como se ha comentado anteriormente cada aplicación Android ejecuta su propio proceso, el cual se crea en el momento de ejecutar la aplicación y permanece hasta que ya no sea requerido o el sistema reclame su memoria para otras aplicaciones.

Uno de los aspectos más importantes de Android es que el tiempo de vida [7] de un proceso no está controlado por la misma aplicación que lo creó, sino que lo determina el sistema a partir de la información que dispone. Éste utiliza parámetros como, que prioridad tienen para el usuario y cuanta memoria queda disponible en él. Android crea una jerarquía de importancia que se detalla a continuación:

1. Se considera un proceso de primer plano si:

i. El proceso que se ejecuta pertenece a una actividad con la que el usuario está interactuando.

ii. Está ejecutando un Broadcast Receiver.

iii. Esta ejecutándose un servicio.

2. Los procesos visibles son aquellos que pueden aparecer en la pantalla del dispositivo pero no en primer plano, ya que su actividad se encuentra pausada. Estos procesos solo serán eliminados en caso de que el sistema necesite sus recursos para mantener en ejecución los procesos de primer plano.

3. Un proceso de servicio es aquel que contiene un servicio del sistema. Se mantendrán en ejecución a no ser que el sistema no pueda mantener a los dos anteriores.

4. Un proceso en background no es visible por el usuario. Es un proceso que aloja una activity que no es actualmente visible para el usuario (su método `onStop ()` ha sido llamado). El sistema puede eliminarlos para dar memoria a cualquiera de los 3 servicios anteriores.

5. Proceso nulo, es un proceso que no alberga nada, lo usa Android como cache para cuando se crea un proceso nuevo.

Android controla las actividades como una pila, las nuevas se sitúan en lo más alto, relegando a la actividad que ocupaba la primera posición a la segunda y así consecutivamente con todas las actividades. A continuación se puede observar el ciclo de vida de una aplicación, así como los métodos que se utilizan para cambiar el estado de los

procesos.

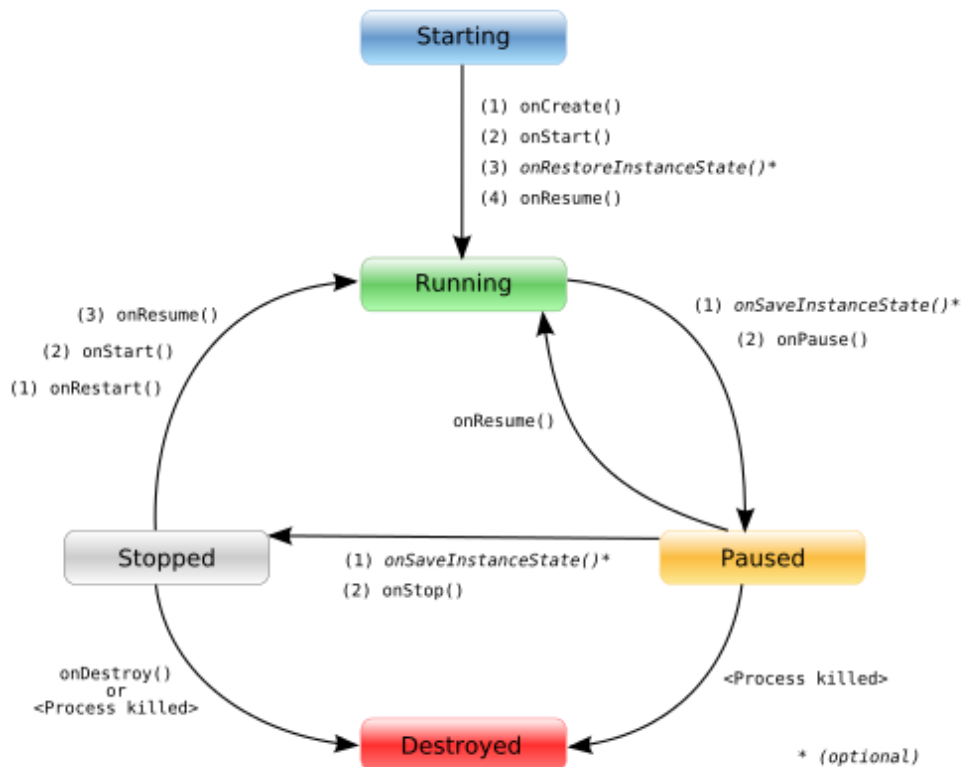


Ilustración 5. Ciclo de vida de Android

Ada Framework

Es un motor de persistencia para Android basada sobre SQLite, se asemeja al DAO (Data Access Object). Un DAO no es más que un componente que nos proporciona los mecanismos para comunicarnos con el almacenamiento de datos. Normalmente los DAO contienen dos tipos de métodos:

- De consulta: nos permiten recuperar un objeto o una lista de objetos de la base de datos aplicando patrones de búsqueda, ordenación, etc.
- De escritura: nos permiten guardar, actualizar o borrar un objeto en la base de datos

ADA Framework [3] nos aporta el concepto de **ObjectSet**. En el sentido estricto, un **ObjectSet** es un DAO, sin embargo lo que lo hace especial es que, como su nombre indica, representa un conjunto de objetos y además implementa la interfaz List, proporcionando los métodos de las listas (get, size, remove, etc.).

La diferencia del **ObjectSet** con los DAO tradicionales es que en los segundos encontramos métodos para recuperar una lista de objetos mientras que con **ObjectSet** lo que tenemos son métodos para “rellenar” el propio **ObjectSet** con los objetos de la base de datos. Además, como acabamos de comentar tenemos todos métodos de una lista.

Capítulo 4

En esta sección describiremos el proceso de ingeniería de software que hemos seguido, para llevar a cabo nuestra propuesta que consiste en un recurso educativo para realiza proceso de Auto-Entrenamiento y Auto-Evaluacion a través de 1 juego educativo. Asimismo, describiremos todas las fases de análisis, diseño e implementación de nuestro proyecto, aunque diferenciaremos dos partes en nuestra aplicación la inclusión preguntas tipos test y el recurso educativo.

Descripción del Proyecto.

Nuestro proyecto es una aplicación destinada a dispositivos electrónicos bajo el sistema operativo de Android tabletas electrónicas que va a constar de varias funcionalidades.

- Una interfaz que permita de manera agradable al usuario pasar las preguntas del módulo SCORM a nuestras bases de preguntas tipo test, pudiendo ser multirepuesta sobre una materia en concreto.
- Un juego con la dinámica de preguntas tipo test obtenidas del módulo anterior.
- Un ranking para ver las puntuaciones de los jugadores y sus preguntas falladas.

En nuestro sistema para la incorporación de las pregunta a la bases de datos es necesario que el fichero este comprimido con formato ZIP y tenga la estructura del

módulo en formato SCORM, que se almacenará en la base de datos interna de Android (SQLite).

En este punto, vamos a describir el juego por el que nos hemos decantado que es el juego de las serpientes y escaleras, básicamente esta diferenciado en dos zonas:

En el marcador se mostrarán elementos como un dado y un máximo de cuatro jugadores y las posiciones de los diferentes jugadores.

Por otro lado, se mostrará el tablero numerado que tiene dibujado serpientes y escaleras, el jugador está representado por fichas. Cuando un jugador cualquiera tira el dado, avanza la ficha tantas posiciones como indique el dado, si esa casilla tiene pintado el inicio de una escalera avanzará, la ficha hasta el final de la escalera. Si por el contrario, cae en una cola de la serpiente baja la ficha hasta la cabeza de la serpiente.

Para este proyecto hemos hecho algunas adaptaciones del original, cada vez que se tira el dado se hace una pregunta tipo test, si se acierta se avanza y si no se detiene en la misma posición. El objetivo del juego es llegar hasta la última posición del tablero, al jugador que gana el juego es el que llegó antes. Finalmente, es de destacar que todas las preguntas falladas se almacenarán en un historial para que el jugador lo pueda consultar en cualquier momento.

A continuación, debido a que nuestro proyecto se basa en el desarrollo software, vamos a definir lo que es Ingeniería del Software y detallar las distintas etapas que la componen.

Como la ingeniería de software se puede entender de muchas formas diferentes, con estas definiciones pretendemos a facilitar su comprensión:

- Ingeniería de software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software. [4].
- Ingeniería de software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de ordenador y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software [2].

La ingeniería de software es un proceso que suele constar de muchas actividades dependiendo del tipo de software que se está desarrollando, describiremos las fases que hemos utilizado para la consecución de nuestro proyecto:

- Especificación de software o ingeniería de requerimientos, ésto se puede definir como el proceso de comprensión y definición de que servicios se requieren en nuestro sistema y las restricciones que tenemos. Esta fase de desarrollo no es lineal, ya que se puede realimentar en las fases que tiene esta etapa, ésto puede producir revisiones constantes de todo lo anterior.
- Diseño de software, que es una descripción de la estructura del software que se va implementar, los datos que son parte del sistema, los algoritmos utilizados, etc.
- Implementación del sistema, se traduce en el diseño a código fuente en un lenguaje específico, es la tarea más compleja ya que requiere conocimientos técnicos sobre el lenguaje y la plataforma hardware sobre la que se va a montar.

- Pruebas: Básicamente consiste en verificar y validar el sistema, comprobando que esté exento de errores.

A continuación, describiremos estas etapas en mayor profundidad, aplicadas siempre a nuestro juego educativo cuya evaluación del alumnado está basada en preguntas tipo test.

Especificación del Software

Como se describió anteriormente en esta etapa se define la utilidad o funcionalidad de nuestro sistema, teniendo en cuenta una serie de restricciones y recursos para la consecución del mismo. Esta etapa se puede dividir en varias subfases o actividades:

- Estudio de viabilidad: En esta subfase se estima si el proyecto es rentable o no, teniendo en cuenta las limitaciones de recursos de hardware, de software, de tiempo, de dinero, etc. La realización de esta etapa deber ser económica y rápida.
- Obtención y análisis de requerimientos. En esta subetapa se obtienen las necesidades del sistema, mediante la observación de sistemas ya existentes, o de charlas con los usuarios de nuestro sistema. Esta etapa puede implicar desarrollo de varios prototipos para ayudar la comprensión del sistema.
- Especificación de requerimiento: En esta actividad se transcribe la información recogida anteriormente en la obtención y análisis de requerimientos, dando lugar a un documento que puede constar de requerimientos del usuario, es decir, declaraciones de necesidades del cliente y el usuario final del sistema y requerimientos del sistema, que

es una descripción más detallada de la funcionalidad que se tiene que proporcionar.

Empezaremos con la primera actividad de la especificación de software, vamos a ver si nuestro proyecto es viable teniendo en cuenta que las limitaciones existentes, a la hora de desarrollar nuestro proyecto, son del tipo económico y de tiempo.

Estudio de viabilidad

Previamente antes de recoger los requisitos funcionales, hay que averiguar si el proyecto va ser rentable o no, qué es lo que se pretende obtener con el software; especificando los recursos disponibles y las restricciones, la inversión que se va a realizar y si esta inversión va ser útil para la organización o la empresa.

Especificaremos que lo que queremos conseguir con nuestro proyecto es realizar un recurso educativo, que ayude a los usuarios de dicha aplicación a entender dicha materia y autoevaluarse, para que puedan adquirir y afianzar conocimientos en cualquier momento y lugar, con cualquier dispositivo móvil con el sistema operativo Android.

Presupuesto

En esta sección veremos la parte económica que conlleva el proyecto, se van a detallar los recursos que se disponen y las limitaciones de tipo económicas para la realización de dicho proyecto.

Desarrollar una aplicación en Android es muy económico y no hacen falta herramientas de pago, el SDK para Android es totalmente gratuito y además es muy fácil encontrar información sobre dicha plataforma.

Los recursos que tenemos disponibles son:

	Material Informático	Coste €	Total €
Software	Ubuntu 12.04	0,00	0,00
	Eclipse	0,00	
	Sdk Android	0,00	
	Visual Paradigm	0,00	
	Community Edition		
Hardware	Ordenador para desarrollo con las siguientes características	1600,00	1600,00
	Microprocesador I7		
	Gigabyte EX58-UD4		
	6 GB de Ram		
	500 GB de disco duro		
	NVidia GTX-260		
	2 Monitores de 22		

Un equipo informático se amortizara en 5 años linealmente, con esto nos vamos a calcular el coste por día que nos cuesta el equipo de desarrollo.

Se estiman para la realización del proyecto 8 días de trabajo.

$$\text{Coste Hardware} = \frac{\text{Precio Adquisición}}{\text{Años Amortización}} = \frac{1600 \text{ €}}{5 \text{ años}} = 320 \text{ €/año}$$

$$\text{Coste por día} = \frac{\text{Coste por año}}{365} = \frac{320 \text{ €}}{365} = 0.87 \text{ €/día}$$

$$\text{Coste para el proyecto} = 0.87 \text{ €/día} * 30 \text{ días} = 26.10 \text{ €}$$

Estos son los recursos que tenemos de todas formas necesitaríamos una Tablet por lo menos para comprobar la aplicación en un dispositivo externo. Esto es aconsejable que aunque dispongamos de un simulador de Android.

Cristóbal Jiménez López

Mancha Real

Jaén

Presupuesto

Código	Descripción	Precio Unidad	Cantidad	Total
	Desarrollar un juego educativo para la plataforma de Android, horas estimadas para el desarrollo de la aplicación (Análisis, Desarrollo, Implementación)	10 €	100	1000€
	Tablet con Android 2.3	87.29 €	1	87,29€
	Uso equipo desarrollo durante 8 días	0.87 €	30	26,1€
			Total	1113,39

El juego educativo implementado para la plataforma Android aprovecha el estándar SCORM de las plataformas de docencia virtual, para nutrir de preguntas al juego educativo lo que beneficiará a los docentes, ya que no tendrán que volver a meter las preguntas en el sistema con lo que se ahorrarán tiempo y dinero.

Aparte de estas ventajas, se puede sacar rendimiento económico mediante dos vías: publicarla en la Play Store de Google y hacerla de pago, o poner publicidad en dicha aplicación gratuita.

Como podemos concluir el proyecto es viable y aporta grandes beneficios a los usuarios.

Especificaciones de requerimientos

En este documento se redacta los servicios o funcionalidades que van ofrecer el sistema y las restricciones asociadas a su funcionamiento. En este tipo de documento podemos encontrarnos con requerimientos funcionales y no funcionales.

- Requerimientos funcionales, son aquellos que expresa la naturaleza del funcionamiento del sistema (como interacciona el sistema con su externo y cual a ser su funcionamiento).
- Requerimientos no funcionales, son aquellos que no tiene que ver con el funcionamiento del sistema, sino que son restricciones a la solución dada.

Especificación de requerimientos de un juego Educativo

Requisitos funcionales:

- El usuario podrá crear preguntas.

El usuario podrá introducir preguntas tipos test que cumpla el estándar SCORM, que se almacenara en el dispositivo propiamente. Estas preguntas tipo test son las que se utilizara en el juego.

- Ver ranking

La aplicación mostrará todos los jugadores las preguntas acertadas, falladas y partidas ganadas. Así como un resumen de las preguntas falladas y cuantas veces.

- Seleccionar temática

La aplicación pedirá al usuario cada vez que inicie un juego elegir la temática de las preguntas tipos test.

- Selección de jugadores

El sistema deberá de permitir la selección de varios jugadores, estableciendo como máximo cuatro jugadores, para empezar la partida.

- Añadir jugador

El sistema debe de permitir al usuario dar de alta un nuevo jugador.

- Jugar

El sistema mostrará al usuario, información sobre la partida como el jugador activo, tirada que ha hecho. Por otro lado mostrará un tablero numerado, con serpientes y escaleras, con lo que el usuario puede interactuar tirando el dado. El sistema seleccionará una pregunta y el usuario responderá. Si responde correctamente, el usuario avanzara tantas posiciones se haya sacado en el dado. Si cae en una cola serpiente

descenderá hasta la cabeza, por lo contrario si se encuentra el principio de una escalera subirá hasta el final de dicha escalera. El juego terminará y ganará el que llegue al final del tablero.

Requisitos no funcionales

Principalmente los requisitos no funcionales vendrán determinados por el hardware y el software de los dispositivos móviles, aunque estos dispositivos van recortando distancias con los PCs de sobremesa, ahora la limitación más importante es la batería.

Requisitos Software

- Sistema operativo. Android 2.3 o superior ya que todas las Api que se manejan son compatibles con esa versión.

Requisito Hardware

- Disponer entre 7 y 10 Mb de almacenamiento. Para almacenar la aplicación y la bases de datos relacionadas.
- Ofrecer bajo consumo, como comentábamos antes es la principal limitación de los dispositivos móviles.
- Procesador mínimo de 600 MHz
- Pantallas mínimas de 7". Ya que la aplicación ha sido desarrollada pensando en las tabletas digitales para salvar la limitación de dispositivos móviles con pantallas más reducidas.

Requisitos no funcionales interfaz.

Este tipo de requerimientos persigue tener una interacción fluida y efectiva. Para que una interfaz sea usable se tiene que basar en los siguientes principios:

- **Facilidad de Aprendizaje:** Que sea fácil de utilizar para nuevos usuarios, para tener una interacción efectiva con el sistema.
- **Facilidad de uso:** Facilidad que el usuario hace el uso del recurso educativo, esto se facilita con el uso de las metáforas.
- **Flexibilidad:** relativa a la variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información. También abarca la posibilidad de diálogo, la multiplicidad de vías para realizar la tarea, similitud con tareas anteriores y la optimización entre el usuario y el sistema.
- **Robustez:** es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos. Está relacionada con la capacidad de observación del usuario, de recuperación de información y de ajuste de la tarea al usuario.

Diseño de software

Una vez conocido el propósito del proyecto software, las funciones que debe cumplir y las restricciones a las que debe someterse, llega el momento de analizar el sistema y crear un modelo del mismo que sea correcto, completo, consistente, claro y verificable. Para conseguir esto se crearán y definirán casos de uso en base a los requerimientos previamente obtenidos. Por último, se describirán ciertos escenarios de acción de dichos casos de uso

Módulo de casos de uso del sistema de auto-entrenamiento y auto-evaluación.

Un caso de uso representa una clase de funcionalidad dada por el sistema como un flujo de eventos. También se puede definir como la representación de una situación o tarea de interacción de un usuario con la aplicación.

Los casos de uso son tareas con significado, coherentes y relativamente independientes, que los actores realizan en su trabajo cotidiano. En un caso de uso concreto puede participar más de un actor.

Los casos de uso describen cómo se realiza una tarea de manera exacta y constan de los siguientes elementos:

- Nombre único e unívoco.
- Actores participantes.
- Condiciones de entrada.
- Flujo de eventos.
- Condiciones de salida.
- Requerimientos especiales.

Por lo tanto, es necesario determinar cuáles son los actores que van a participar en cada uno de los casos de uso. Un actor modela una entidad externa que se comunica con el sistema, es decir, es un tipo de usuario del sistema. Un actor, al igual que un caso de uso, debe tener un nombre único y puede tener una descripción asociada.

En nuestro sistema de auto-entrenamiento van a aparecer los siguientes actores.

Usuario: Se refiere al que utiliza la aplicación

BBDD (Bases de datos móvil): Es la base de datos interna que almacena los datos de las preguntas, así como los de la puntuaciones y rankings en los distintas partidas.

Una vez definidos los actores que se relacionan con nuestro sistema, vamos a pasar a los distintos casos de uso.

Un paso previo a la creación y descripción de los distintos casos de uso es la obtención de los diversos diagramas de caso de uso en nuestro sistema. El primero es el diagrama frontera que describe completamente la funcionalidad del sistema y se presenta en la siguiente figura.

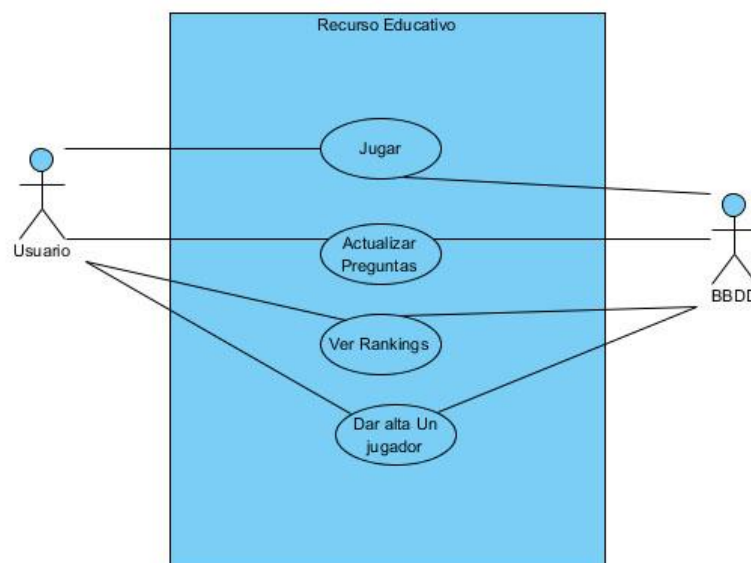


Ilustración 6 Diagrama frontera del sistema

Los casos de uso mostrados en un diagrama frontera pueden ser suficientemente precisos o necesitar ser explicados en mayor detalle. A la hora de detallar un caso de uso se pueden emplear dos tipos de relaciones:

- <<extend>>: Es una relación cuya dirección es hacia el caso de uso que detallamos y que representa comportamientos excepcionales del caso de uso.
- <<include>>: Es una relación cuya dirección es contraria a la de la relación <<extend>> que representa un comportamiento común del caso de uso.

En nuestro caso, se da la circunstancia de que sólo un caso de uso debe ser descrito con mayor profundidad, porque presenta una mayor complejidad, tratándose del caso de uso “Jugar”. Los demás son tan simples que no lo requieren en un mayor detalle, aunque también los describiremos, de manera breve.

A continuación, se expone en la figura 3.4 el diagrama que se identifica con el caso de uso Jugar:

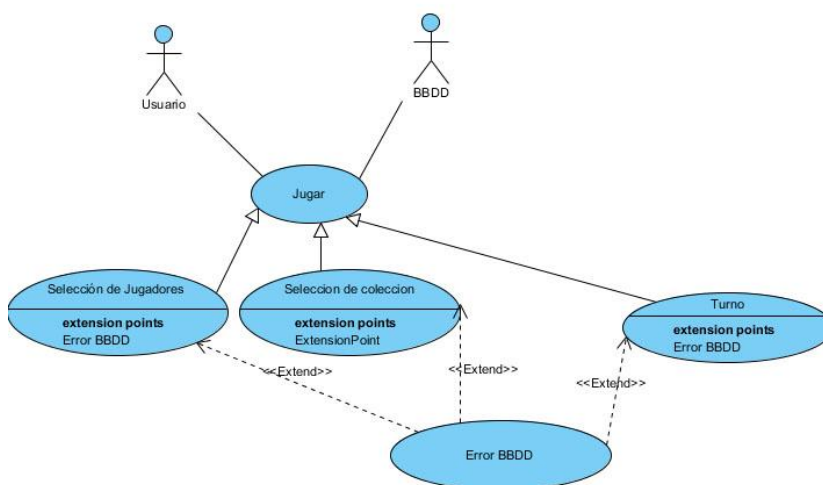


Ilustración 7 Casos de uso de Jugar

Caso de Uso 1: Seleccionar jugadores

Actores participantes: Usuario, BBDD.

Condiciones de entrada: Que el alumno haya iniciado la aplicación, haya elegido la opción de Jugar y la base de datos no esté vacía.

Flujo de eventos:

1. El sistema mostrará por pantalla los jugadores que hay disponibles en el sistema.
2. El usuario seleccionará su nombre.
3. El sistema pasará su nombre de Jugadores a Jugadores Seleccionados. Repetir el paso 2 como máximo 4 veces.
4. El usuario pulsará el botón siguiente.
5. El sistema mostrará la siguiente pantalla.

Condiciones de salida: Que el usuario haya seleccionado por lo menos 2 jugadores y le haya dado siguiente

Flujo alternativo:

En cualquier momento el usuario pueda añadir un nuevo jugador.

Caso 2. Seleccione las colecciones

Actores participantes: Usuario, BBDD.

Condiciones de entrada: Que el usuario haya elegido los jugadores y que le haya dado al botón de siguiente.

Flujo de eventos:

1. El sistema muestra todas las colecciones de preguntas que están disponibles.
2. El usuario elige una de las colecciones.
3. El sistema muestra por pantalla los jugadores que hay disponibles en el sistema.

Condiciones de salida: Que el jugador seleccione una colección.

Caso 3. Turno

Actores participantes: Usuario, BBDD.

Condiciones de entrada: Que el usuario haya elegido los jugadores y le haya dado al botón de siguiente.

Flujo de eventos:

1. El sistema mostrará todos los jugadores y las piezas en la posición 1.
2. El usuario pulsará sobre el tablero del juego.
3. El sistema le mostrará una pregunta con sus diferentes opciones.
4. El usuario seleccionará una respuesta correcta.
5. El sistema lanzará el dado y moverá la ficha, teniendo en cuenta si cae una escalera sube hasta la posición y si se encuentra con una serpiente baja a la posición. Cambia turno del jugador. Repetir desde el paso 2.

Condiciones de salida: Que un jugador cualquiera llegue hasta el final del tablero o el usuario se salga de la partida.

Flujo alternativo:

En el paso 4 que el usuario seleccione una pregunta incorrecta. En el paso 5 la ficha no se moverá y cambiara de jugador. Repetir desde el paso 2.

Caso de uso 4: Actualizar preguntas.

Actores participantes: Usuario, BBDD.

Condiciones de entrada: Que el usuario haya iniciado la aplicación, haya elegido la opción de Explorar.

Flujo de eventos:

1. El sistema muestra los archivos que hay en el sistema.
2. El usuario selecciona el fichero comprimido válido.
3. El sistema descomprime el fichero, añadiendo las preguntas a la bases de datos y muestra un mensaje de que ha convertido el fichero.
4. El usuario pulsa aceptar.
5. El sistema vuelve al menú principal.

Condiciones de salida: Que seleccione un fichero correcto pulse atrás.

Flujo alternativo:

En el paso 2 que el usuario seleccione un archivo incorrecto, en el paso nueve se mostrar un aviso de que es un archivo inválido.

Caso de uso 3: Ranking

Actores participantes: Usuario, BBDD.

Condiciones de entrada: El usuario haya iniciado la aplicación, haya elegido la opción de ver Ranking.

Flujo de eventos:

1. El sistema muestra todos los jugadores registrados en el sistema, mostrando su nombre, numero de partidas, preguntas acertadas y falladas.
2. El usuario selecciona un jugador.
3. El sistema muestra las preguntas falladas y cuantas veces falladas.

Condiciones de salida: Pulse botón para atrás

Caso de uso: dar de alta un jugador

Actores participantes: Usuario, BBDD.

Condiciones de entrada: El usuario haya iniciado la aplicación, haya elegido la opción de jugar.

Flujo de eventos:

1. El sistema muestra todos los jugadores registrados en el sistema, mostrando su nombre.
2. El usuario pulsa la tecla menú.
3. El sistema muestra añadir jugador.
4. El usuario pincha en añadir jugador.
5. El sistema muestra un formulario para dar de alta.
6. El usuario rellena el formulario y le da aceptar.
7. El sistema registra el usuario en el sistema.

Condiciones de salida: Pulse botón para atrás o cancelar.

Escenarios del sistema

Un caso de uso es una representación abstracta de una funcionalidad para realizar por el sistema. La representación concreta de un caso de uso se realiza mediante la creación de uno o más escenarios que muestren todas las interacciones posibles, entre el sistema y sus usuarios.

Los escenarios son historias ficticias que describen posibles interacciones con una interfaz. Permiten a los diseñadores anticiparse a los problemas. Aunque son historias ficticias deben hacerse lo más detalladas posibles, así por ejemplo, los personajes deben

tener nombres, motivaciones para usar la interfaz, deben encontrarse en entornos reales con las restricciones que ello conlleva, etc. De esta manera se facilita a los diseñadores la discusión sobre la interfaz ya que a las personas nos cuesta más trabajo discutir sobre una situación abstracta.

Esta forma de proceder fuerza a los diseñadores a considerar el rango de usuarios que van a usar el sistema y el rango de actividades por las que lo van a usar. Los escenarios permiten hacer diferentes combinaciones de usuarios y actividades de forma que se tengan en cuenta todas las posibilidades.

Un escenario está formado por los siguientes elementos:

- Un nombre único y unívoco.
- Una descripción.
- Los actores participantes.
- El flujo de eventos.

Como se ha indicado, para cada caso de uso puede haber varios escenarios. Para nuestro caso, vamos a definir los escenarios más importantes que presentan nuestros sistemas.

Escenario 1: Seleccionar Jugadores

Nombre: Seleccionar_Jugadores

Descripción: El usuario de la aplicación seleccionara los jugadores que van a jugar la partida, seleccionando como máximo 4 y mínimo 2. Cuando hayas seleccionado los jugadores pulsa el botón siguiente.

Actores: Cristóbal, Macarena y las bases de datos.

Flujos de Eventos:

1. El sistema muestra todos los jugadores, con sus partidas ganadas.
2. Cristóbal selecciona su nombre.
3. El sistema cambiara el nombre del Jugador a Jugadores Seleccionados.
4. Macarena seleccionará su nombre.
5. El sistema cambiará el nombre de Jugador a Jugadores Seleccionado.
6. El usuario pulsaría siguiente.
7. El sistema mostraría el tablero con Cristóbal Y Macarena.

Escenario 2: Importar preguntas**Nombre:** Importar_Preguntas

Descripción: El usuario seleccionará el paquete de preguntas que están almacenadas en un fichero zip, para almacenarlas en nuestra aplicación.

Actores: Cristóbal las bases de datos.

Flujos de Eventos:

1. El sistema muestra todas las carpetas y archivos que están en la raíz.
2. Cristóbal selecciona el directorio SDCARD
3. El sistema mostrará el contenido
4. El usuario seleccionará un zip que es el generado por la plataforma.
5. El sistema mostrará un mensaje de que las preguntas ya se habrá importado y cerrar la ventana.

Escenario 3: Jugar

Nombre: Jugar

Descripción: Los usuarios jugarán al juego de la serpientes y escaleras, por turnos y en cada turno si aciertan bien la pregunta avanzan y si fallan se queda en la posición donde están. Cuando la ficha del jugador cae en una escalera

Actores: Cristóbal, Macarena y las bases de datos.

Flujos de eventos:

1. El sistema mostrará una lista con todos los bancos de preguntas importados desde recursos SCORM.
2. El usuario elegirá un banco de preguntas concreto.
3. El sistema mostrará el tablero para jugar es el turno de Cristóbal.
4. Cristóbal pulsa en cualquier lado del tablero.
5. El sistema formulará una pregunta
6. Cristóbal responderá la pregunta.
7. El sistema moverá la ficha donde corresponda.
 - 7.1. Si la respuesta es incorrecta, el sistema no moverá la ficha del jugador activo.
 - 7.2. Si la repuesta es correcta, el sistema tirara el dado y la ficha avanzará. Si la posición tiene una escalera avanzará sobre la escalera y si es una serpiente descenderá.
 - 7.3. El sistema cambiara el turno y se lo dará turno al siguiente jugador.
8. Repetir el proceso desde el punto 2, con los jugadores restantes hasta que alguno de ellos llegue al final del tablero.

9. Cuando alguna de los jugadores (Cristóbal o Macarena) lleguen al final el sistema mostrará que quien es el ganador.
10. El usuario presiona aceptar.
11. El sistema mostrará la pantalla selección de jugadores.

Escenario 4: Añadir Jugadores

Nombre: Añadir Jugadores

Descripción: El usuario no se encontrara en la lista del usuario del sistema, y se registrará en el sistema para poder jugar

Actores: Cristóbal y bases de datos.

Flujo de datos:

1. El sistema nos mostrará la pantalla de selección de alumnos.
2. Cristóbal vera que no está en los jugadores disponibles y presionará el botón menú del dispositivo Android.
3. El sistema mostrará las opciones disponibles.
4. El usuario seleccionará la opción de añadir usuario.
5. El sistema mostrará un formulario de registro.
6. El usuario introducirá el nombre y seleccionara el color de su ficha.
Cuando Cristóbal lo haya rellene su nombre y escoja el color. El usuario presionará el botón Aceptar.
7. El sistema mostrará la pantalla de selección de jugadores.

Diseño de la interfaz de la aplicación

Estilo

El estilo trata acerca de la forma en que el contenido debe ser presentado al usuario, como es la fuente del texto, colores, alineado, cabeceras, etc. Para ello, se definen guías de estilo, para mantener una consistencia en el estilo para toda la interfaz de la aplicación. Cuando van a participar en el diseño de una interfaz varios diseñadores, éstos deben tener definida una guía de estilo con el objeto de evitar incoherencias en la interfaz.

Sin embargo, a pesar de lo que pueda parecer en un principio, también es de gran utilidad definir una guía de estilo cuando sólo hay un diseñador encargado de la interfaz. Esto es debido a varias razones:

- En ocasiones es posible que mantener la coherencia y consistencia de una interfaz, sea complicado incluso si sólo hay un diseñador si esta es muy grande o muy ambiciosa.
- El diseñador primitivo puede, por las más diversas razones, abandonar el diseño. Por tanto, es de gran utilidad para sus sustitutos contar con una guía de estilo predefinida para no tener que empezar desde cero de nuevo. Esto es también aplicable cuando no es el diseñador original el que se encarga de la actualización o el mantenimiento de la interfaz.

En nuestro caso, como se ha comentado, la mayor parte de los aspectos del estilo de la interfaz dependerá de la guía de estilo propia de la interfaz del dispositivo móvil sobre el que se ejecuta la aplicación, como es la fuente del texto, el color e incluso muchas veces el alineamiento sobre la pantalla. Sólo nos quedan algunos detalles que podemos definir, que son:

- Menús fuente calibri siempre en mayúscula.
- Tono pasteles para los colores de los jugadores y fichas.
- En todas las pantallas tiene que ser el fondo igual excepto la pantalla del explorador ya que dificultaría su comprensión.

Metáforas

Según la definición de RAE una metáfora es la aplicación de una palabra o expresión a un objeto o concepto, al cual no denota literalmente, con el fin de sugerir una comparación para facilitar su comprensión, aplicando este concepto al campo de las interfaces utilizar un objeto conocido por el usuario para que su utilización facilite parte del dominio de la interfaz o funcionalidades. Por ejemplo, la flecha de deshacer de cualquier editor de textos, es vuelta a un estado anterior. Las metáforas que hemos utilizado son:

- Un dado que representa un número aleatorio del 1 al 6.
- Ficha que representa a los jugadores y en qué posición están del tablero.
- Tablero que representa un tablero de la realidad.
- Escaleras representa algo positivo y asciende posiciones.
- Serpientes que representa algo negativo y desciende posiciones.

Principales pantallas de la aplicación móvil



Ilustración 8 Pantalla Principal



Ilustración 9 Pantalla Selección de Jugadores



Ilustración 11 Pantalla Creación de un jugador

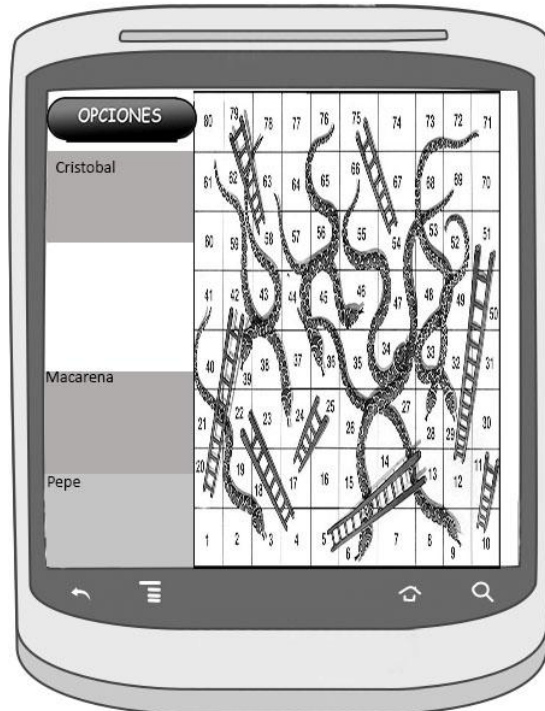


Ilustración 10 Pantalla del Juego

Pantalla menú principal y selección de jugadores.



Ilustración 12 Pantalla para ver estadísticas y Ranking

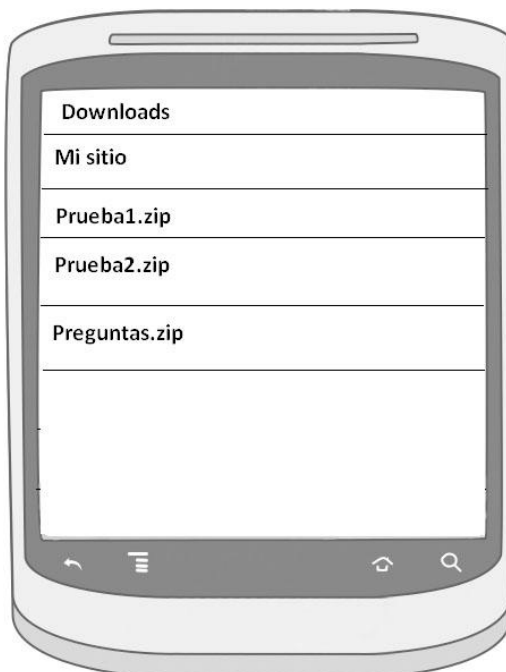


Ilustración 13 Pantalla para importar preguntas

Storyboard

A continuación vamos a presentar las acciones más comunes de nuestro recurso educativo, que son jugar, ver ranking e importar preguntas en formatos SCORM.

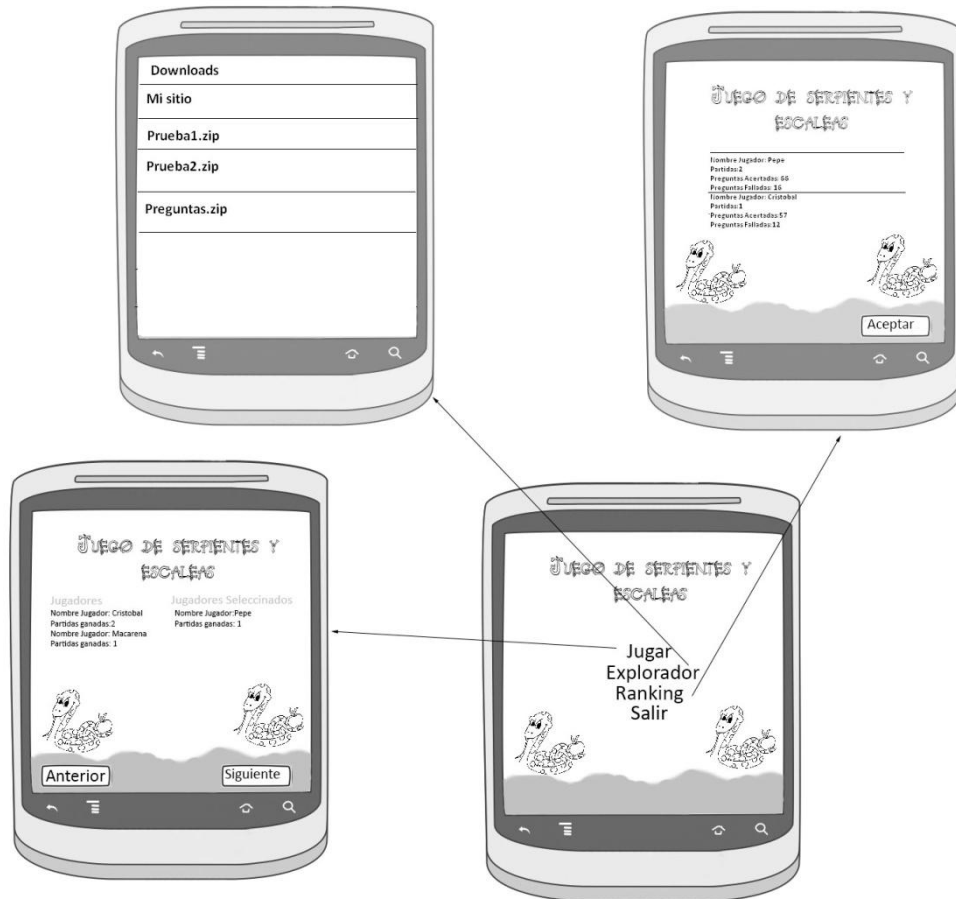
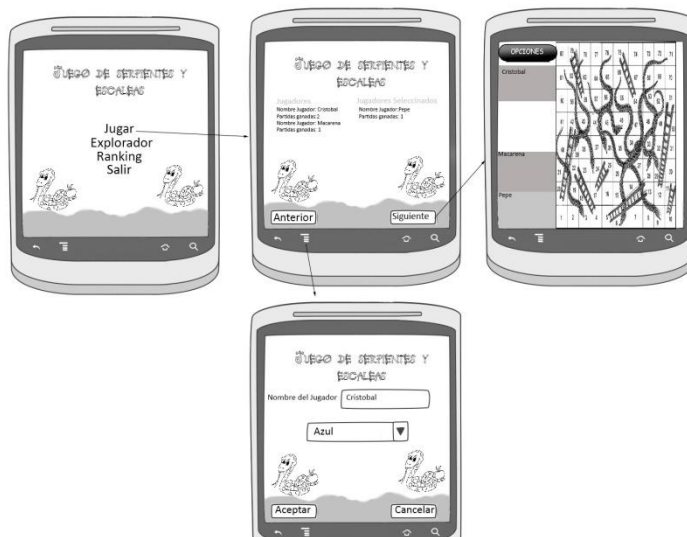


Ilustración 14 Pantallas Principales



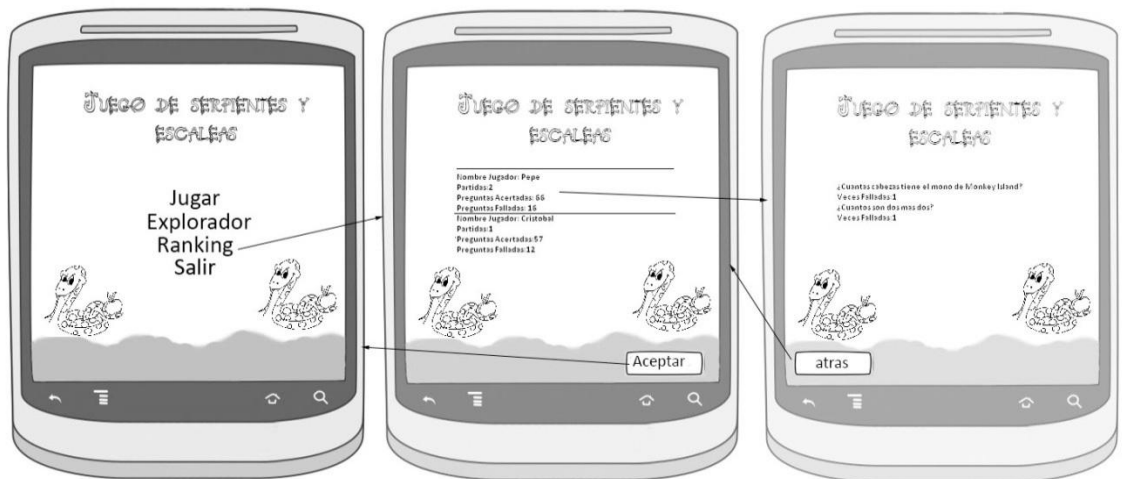


Ilustración 16 Pasos para ver ranking



Ilustración 17 Pasos para importar preguntas

Diseño de los datos de la Aplicación Móvil

El objetivo de esta fase del diseño software es determinar la estructura que poseen cada uno de los elementos de información del sistema, es decir, la estructura de los datos sobre los que se va a trabajar.

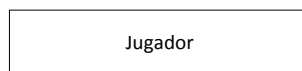
Para efectuar y realizar el diseño adecuado de los datos utilizaremos el modelo Entidad-Relación.

El modelo Entidad-Relación (también conocido por sus iniciales: E-R) es una técnica de modelado de datos que utiliza diagramas entidad-relación.

No es la única técnica de modelado pero si es la más extendida y utilizada.

Un diagrama entidad-relación está compuesto por tres tipos de elementos principales:

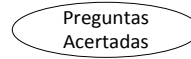
- Entidades: Objetos (cosas, conceptos o personas) sobre los que se tiene información. Se representan mediante rectángulos etiquetados en su interior con un nombre. Una instancia es cualquier ejemplar concreto de una entidad.



- Relaciones: Interdependencias entre uno o más entidades. Se representan mediante rombos etiquetados en su interior con un verbo. Si la relación es entre una entidad consigo mismo se denomina reflexiva, si es entre dos entidades se denomina binaria, ternaria si es entre tres y múltiple si es entre más.



- Atributos: Características propias de una entidad o relación. Se representan mediante elipses etiquetados en su interior con un nombre.



En los diagramas entidad-relación también hay que tener en cuenta otros aspectos como pueden ser:

- Entidades débiles: Son aquellas que no se pueden identificar unívocamente sólo con sus atributos, sino que, necesitan estar relacionadas con otras entidades para existir. Se representan con dos rectángulos concéntricos de distinto tamaño con un nombre en el interior del más pequeño.
- Cardinalidad de las relaciones: Existen tres tipos de cardinalidades de una relación según el número de instancias de cada entidad que involucren:
 - Uno a uno, una instancia de la entidad A se relaciona solamente con una instancia de la entidad B. (1:1)
 - Uno a muchos: cada instancia de la entidad A se relaciona con varias de la entidad B. (1:*)
 - Muchos a muchos: cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B. (*:*)

Claves: cada entidad de un diagrama entidad-relación debe tener una clave, debe estar formada por uno o más de sus atributos.

Una vez conocidos los elementos que forman parte de un diagrama entidad-relación podemos empezar a desarrollar el modelo entidad-relación.

Los pasos a seguir son los siguientes:

1. Convertir el enunciado del problema en un Esquema Conceptual del mismo.

2. Convertir este Esquema Conceptual (o EC) en uno más refinado conocido como Esquema Conceptual Modificado (ECM).

3. Obtener las tablas de la base de datos a partir del Esquema Conceptual Modificado.

Normalización en el modelo Entidad-Relación

La normalización es un proceso consistente en imponer a las tablas ciertas restricciones mediante una serie de transformaciones consecutivas. Con ello, se asegura que las tablas contengan los atributos necesarios y suficientes para describir la realidad de la entidad que representan, separando aquellos que pueden contener información cuya relevancia permite la creación de otra nueva tabla.

Para asegurar la normalización, Codd estableció tres formas normales, las cuales hacen que una base de datos (si las cumple) esté normalizada.

Estas formas normales son:

Primera forma normal (F<1): Una tabla está en FN1 si todos los atributos no clave, dependen funcionalmente de la clave, o lo que es lo mismo, no existen grupos repetitivos para un valor de clave.

Segunda forma normal (F<2): Una tabla está en FN2 si está en FN1 y además todos los atributos que no pertenecen a la clave dependen funcionalmente de forma completa de ella. De esta definición se desprende que una tabla en FN1 y cuya clave está compuesta por un único atributo está en FN2.

Tercera forma normal (F<3): Una tabla está en FN3 si está en FN2 y además no existen atributos no clave que dependan transitivamente de la clave.

En nuestro caso y teniendo en cuenta que se trata de un prototipo de juego educativo orientado a dispositivos móviles, donde dichos dispositivos poseen unas limitaciones tanto en memoria como en capacidad de computación reducidas, no debemos abordar una excesiva complejidad en los datos que se van a manejar, ya que por un lado no nos lo permitirían estos dispositivos (no soportan todos los tipos de datos que suelen soportar un PC) y además sufriríamos un detrimento de las prestaciones en cuanto a velocidad de reacción y refresco del dispositivo. Por lo tanto, lo que intentaremos es simplificar al máximo y quedarnos con los datos realmente importantes o que más interesen para conseguir nuestro objetivo.

Teniendo en cuenta que nuestros recursos son limitados, hemos tenidos en cuenta las siguiente elementos.

- Los Jugadores, son los datos del jugador y sus estadísticas.
- Las Colecciones, son colecciones de preguntas extraídas de un recurso SCORM que de la colección solo extraemos el nombre.
- Preguntas, son las preguntas que pertenecen a las colecciones, pudiendo tener varias opciones indeterminadas.
- Respuestas, son las repuestas a la pregunta almacenando su peso, porque puede haber una respuestas más verdadera que otra. Por eso, cero se considera incorrecta y las correctas son mayor que cero considerando su veracidad.

Esquema conceptual.

Los elementos previamente citados, hay que convertirlos en entidades, atributos y relaciones. En nuestro caso tendremos varias entidades como Jugador, Preguntas, Respuestas, las relaciones que se establecen son las siguientes:

R1: Guarda, cuya cardinalidad es de N:N, donde se indica el historial de preguntas falladas.

R2: Una colección, es una relación de cardinalidad de 1:N, donde queremos decir que una colección tiene múltiples preguntas.

R3: Esta relación tiene cardinalidad de 1:N para almacenar las respuestas de las preguntas.

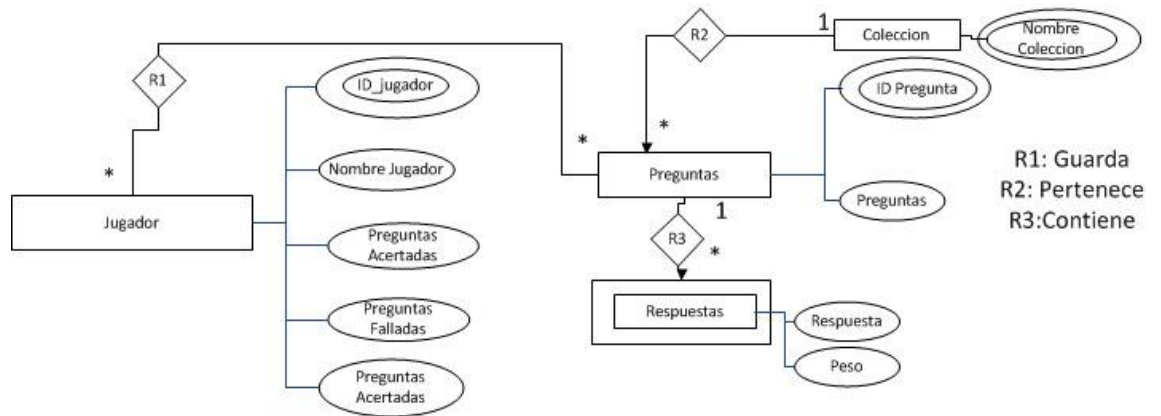


Ilustración 18 Esquema conceptual

Esquema conceptual modificado

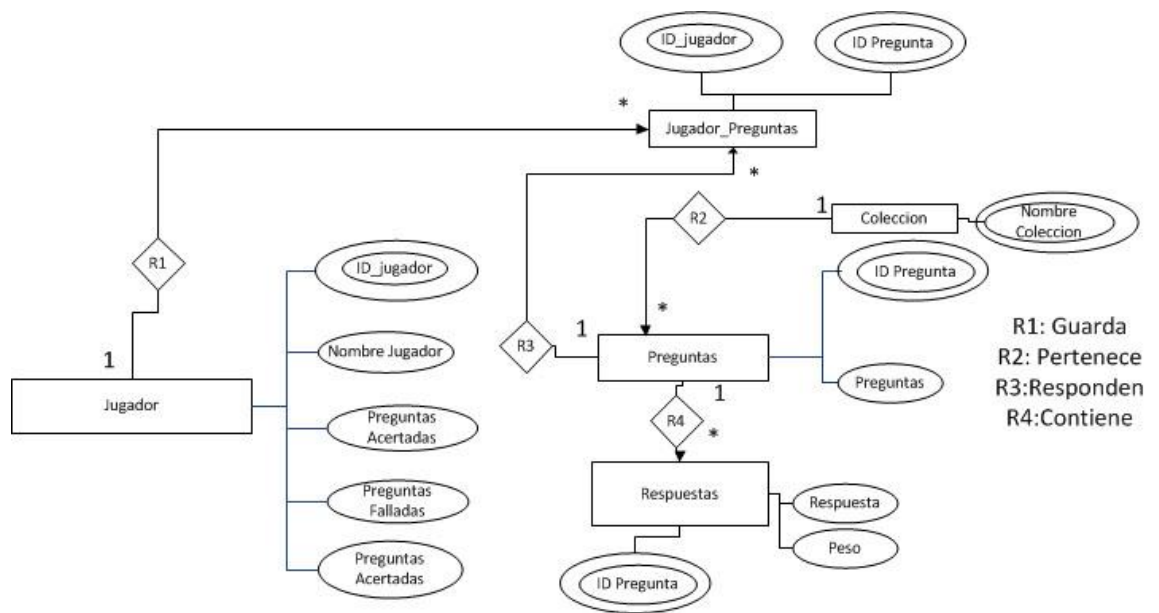


Ilustración 19 Esquema conceptual modificado

Para obtener el Esquema Conceptual Modificado a partir del Esquema Conceptual se deben hacer los cambios que siguen a continuación:

- Eliminar todas las entidades débiles.
- Eliminar las relaciones de muchos a muchos.
- Eliminar las relaciones con atributos que hayan en nuestro sistema.

Para convertir nuestro esquema conceptual a esquema conceptual modificado en nuestro caso, hay que Eliminar la relación R1 porque es de N:N, dan lugar a dos nuevas relaciones y a una nueva entidad. Se crearía la entidad Jugador_Preguntas cogeriámos de atributos las claves principales de Jugador y Preguntas, aparte de tener una relación de Jugador a Jugador_preguntas y Preguntas a Jugador_preguntas, eso por un lado.

Por otro lado, para quitar las entidades débiles las respuestas se deben poner como la clave de las preguntas, así lo identificarán unívocamente la respuesta a que pregunta corresponde.

Tablas de la aplicación móvil.

Una vez obtenido el ECM, podemos pasar a identificar las tablas que se va a componer las bases de datos, considerando que:

- Cada entidad del ECM se transforma directamente en una tabla.
- La entidad se convierten en los campos de las tablas respectivas.

Así obtenemos las siguientes tablas.

- Jugador.
- Preguntas.
- Jugador_Preguntas.
- Respuestas.
- Colección.

Para aclarar el contenido de cada tabla de las bases de datos vamos a detallar cada tabla con sus los campos correspondientes:

Jugador		
Campo	Tipo	Descripción
Id	Entero Largo	Identificador del jugador
NombreJugador	Cadena	Nombre del jugador
Puntuación	Entero	Número de partidas ganadas
PreguntasFalladas	Entero	Número de preguntas falladas
PreguntasAcertadas	Entero	Número de preguntas acertadas

Colección

Campo	Tipo	Descripción
IdColeccion	Entero Largo	Identificador de la colección
NombreColeccion	Cadena	Nombre de colección

Pregunta		
Campo	Tipo	Descripción
IdColeccion	Entero Largo	Identificador de la colección a que pertenece
IdPregunta	Entero Largo	Identificador de la pregunta
Pregunta	Cadena	Indica la pregunta

Jugador_Preguntas		
Campo	Tipo	Descripción
IdPregunta	Entero Largo	Identificador de la pregunta
IdJugador	Cadena	Identificador del jugador

Respuestas		
Campo	Tipo	Descripción
Id_pregunta	Entero Largo	Identificador de la pregunta
Respuesta	Cadena	La repuesta de la cadena
Peso	Entero	El valor de la repuesta

Implementación

La implementación es la actividad final de la Ingeniería del software, aquella en el que el modelo obtenido en las actividades anteriores, debe de transformarse en el código fuente. Puesto que nosotros estamos programando para la plataforma Android hemos elegido eclipse, ya que es fácil instalar la herramienta de desarrollo de Android.

Capítulo 5

En este capítulo, expondremos las conclusiones derivadas de nuestro proyecto fin de carrera y formularemos posibles ampliaciones del proyecto.

Actualmente las Tecnologías de la Información y Comunicación (TIC) tienen un importante peso en nuestra vida cotidiana, propiciando el desarrollo de nuevas metodologías de E-Learning y de M-Learning. El uso de las metodologías M-learning y E-learning, permite mayor flexibilidad en el aprendizaje porque el alumno es el centro de esta metodología de enseñanza, teniendo en cuenta que el usuario sabe lo que quiere aprender y hasta donde quiere llegar, y no se ve limitado ni en lugar y ni en el tiempo para poder aprender.

En nuestro proyecto hemos desarrollado un recurso educativo para dispositivos móviles con el sistema operativo.

En este proyecto fin de carrera nos hemos decantado por desarrollar el recurso educativo en el sistema operativo Android.

Nuestro recurso educativo consta de tres partes diferenciadas.

Un módulo para pasar preguntas tipo test de cualquier plataforma educativa que sea compatible con SCORM estándar, ya que utiliza un código de marcado XML (eXtensible Markup Language). Para ello, utilizamos la librería jdom que nos permite extraer información de cualquier XML. Además, hemos utilizado ADA Framework que es un Framework para Android que nos facilita la utilización de SQLite, permitiendo hacer el intercambio de forma casi automática de XML a SQLite.

Otro módulo en sí es el recurso educativo que es el juego de la serpientes y escaleras que hemos utilizado de nuevo varios framework.

Andengine un framework gratuito pensado para juegos de la plataforma Android además está muy completo tiene soporte OpenGL, sonido, implementa sistema de colisiones, incluye física, posibilidad de programar varios elementos de entrada, posibilidad de hacer el juego multijugador, implementa sistemas de partículas. En definitiva una api muy completa para el desarrollo de juegos en Android.

De nuevo el ADA Framework que es una api para manejo de SQLite, esta api optimiza al máximo casi todas las operaciones con SQLite, con una interfaces sencillas de utilizar.

Por último el sistema de Ranking implementado simplemente con la API de Android.

Para la publicación y distribución de la aplicación Google nos da la posibilidad de publicarla en la Play Store.

Con nuestro proyecto se consigue una aplicación intuitiva, amena, atractiva que ayude al alumno a medir sus conocimientos de una manera fácil y rápida consiguiendo que el alumno se divierta aprendiendo. Además, la aplicación aprovecha los recursos educativos de las distintas plataformas educativas compatibles con SCORM, para transferir las preguntas de las que se nutre nuestro recurso educativo, con lo que se consigue no gastar más recursos en generar las preguntas que ya estaban hechas.

Entre los trabajos futuros que nos proponemos destacan los siguientes.

- Incorporar juegos educativos a la aplicación como el ahorcado o el trivial.
- Incorporar otro tipo de preguntas que facilitan los bancos de preguntas del formato SCORM.

- Implementar el multijugador desde diferentes dispositivos móviles.

Anexo. Manual de instalación y usuario

Manual de instalación

Para poder instalar el recurso educativo en un dispositivo móvil bajo Android hay que seguir los siguientes pasos.

1. Transferir fichero .apk a la tarjeta de memoria, este proceso se puede hacer de varias formas con la aplicación Airdroid o con un cable de datos como si fuera una unidad de datos más.
2. Ir al menú del dispositivo Android. Seleccionamos Ajustes, después seguridad y habilitamos el origen desconocido de datos para la versión 4 y para versiones anteriores elegimos aplicaciones habilitamos.



Ilustración 20 Menú principal

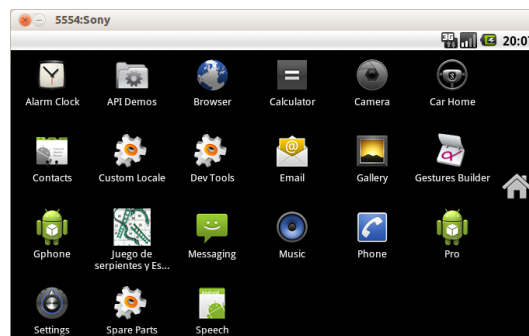


Ilustración 21 Aplicaciones instaladas



Ilustración 22 Menú de ajustes

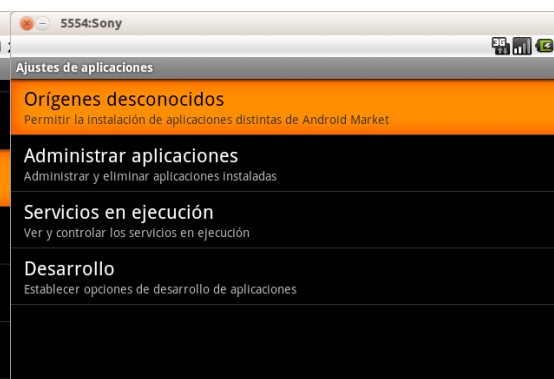


Ilustración 23 Ajustes de aplicación

3. Con cualquier explorador de archivos en nuestro caso utilizamos uno gratuito llamado Es Explorador de Archivos. Nos dirigiremos donde está el paquete de nuestra aplicación en nuestro caso está en el directorio /Sdcard/. Seleccionaremos el archivo que hemos copiado en el paso 2. Pinchamos sobre él y nos dirá que si deseamos instalar la aplicación. Con eso ya habremos instalado la aplicación.

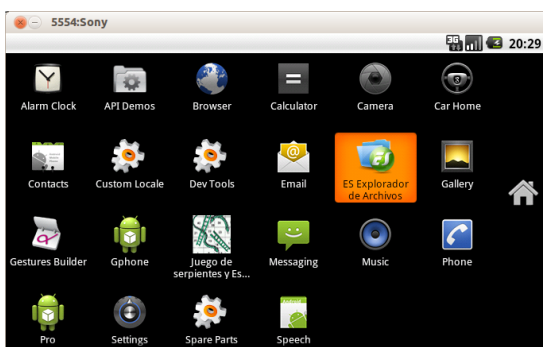


Ilustración 24 Explorador de archivos

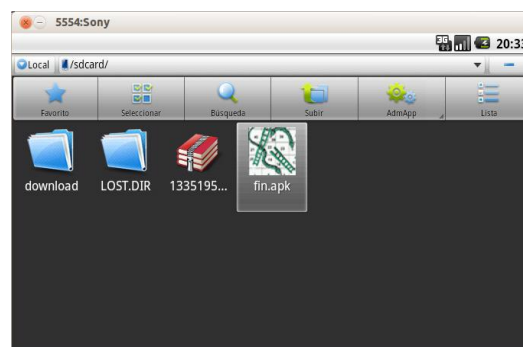


Ilustración 25 Archivo de .apk

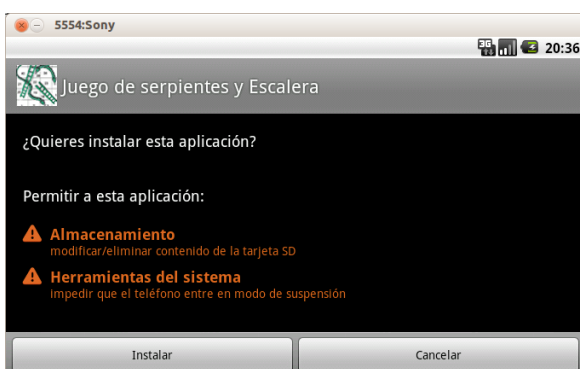


Ilustración 26 Cuadro de dialogo para instalar la aplicación

Manual de la aplicación

Este manual de usuario pretende familiarizar al usuario con el recurso educativo y resolver las posibles dudas que pueda tener al utilizarlo.

Explicaremos el funcionamiento del recurso y las opciones que posee.

El primer paso es localizar la aplicación y ejecutarla. En este momento se inicia una pantalla de presentación que indica el autor de la aplicación. Después de esa pantalla de presentación muestra la pantalla principal con las diferentes opciones.



Ilustración 27 Menú principal del recurso educativo.

Jugar

Si el usuario desea jugar, desde el menú principal pulsa Jugar. Seguidamente aparecerá la pantalla de selección de Jugadores.

En esta pantalla se mostrarán todos los jugadores registrados en el sistema con su correspondiente puntuación. El jugador buscará su nombre y pulsará sobre él, como mínimo tiene que haber dos jugadores y máximo puede haber hasta cuatro jugadores.



Ilustración 28 Pantalla selección de jugadores

Si el jugador en cuestión no se encuentra presione la tecla menú de su dispositivo para agregar al jugador en cuestión y rellene el nombre y elige el color deseado.



Ilustración 29 Pantalla para añadir el jugador

Presiona aceptar y vuelve a la pantalla de seleccionar jugadores. Cuando haya seleccionado los jugadores pulse siguiente y aparecerá un cuadro de dialogo, eligiendo colección de pregunta, pero si no hay colección de preguntas el sistema dará un aviso, lo que el usuario tendrá que irse a la opción de explorar e importar las preguntas, para que el juego le deje jugar. En el caso de que se haya introducido preguntas, aparecerá todas las colecciones de preguntas, el usuario seleccionara la colección de pregunta que se quiera y finalmente se mostrara el tablero.

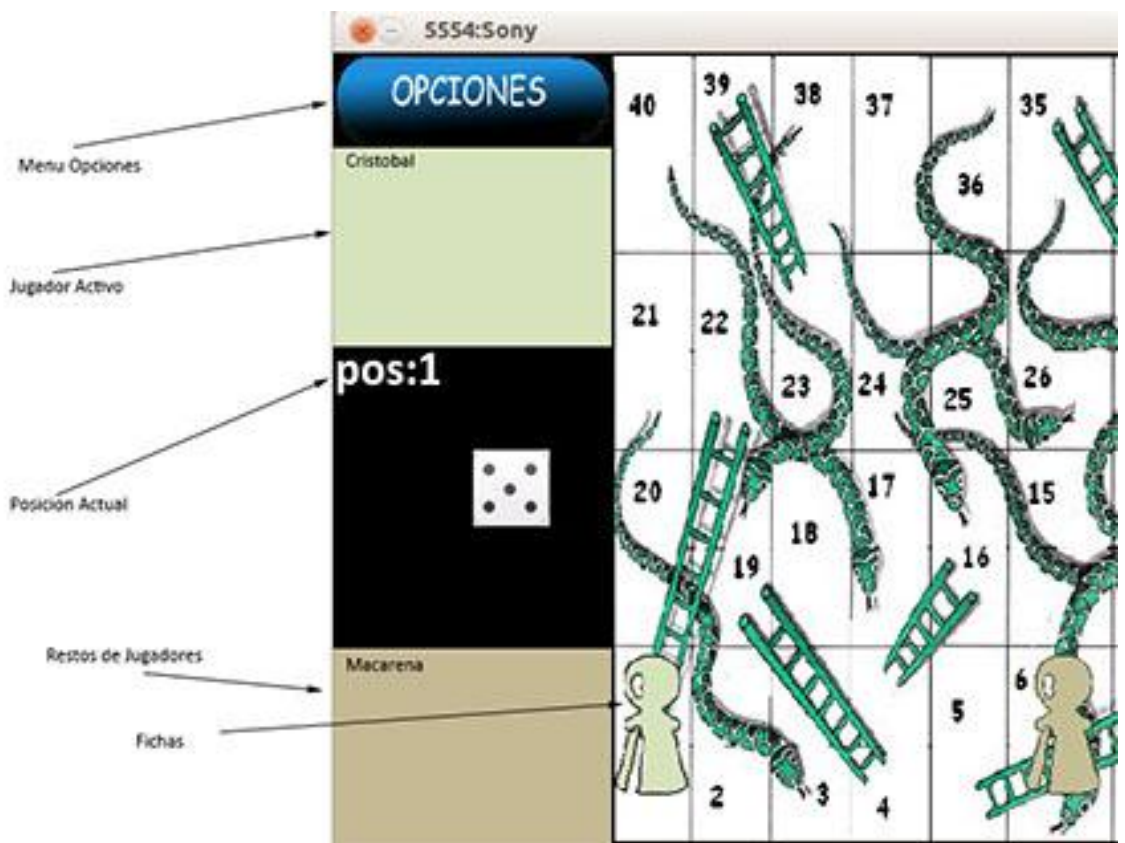


Ilustración 30 Descripción tablero

En el tablero se pueden diferenciar dos partes. La primera de ella es información y la segunda es el tablero propiamente dicho. Para empezar a jugar el

jugador activo presionará sobre cualquier parte del tablero para que formule su pregunta, si la acierta avanzará y si no se quedará en el mismo lugar. Si se cae en una casilla que hay una escalera avanza posiciones y si se cae en una serpiente desciende posiciones. En cualquier momento, se puede pulsar el botón de opciones. Saldrá un menú con las siguientes opciones Reiniciar, Ayuda y Salir.

Reiniciar Vuelve a empezar el juego.

Ayuda. Muestra una ayuda del juego de serpientes y escaleras.

Salir. Sale del juego y va la pantalla de selección.

Cuando se sale de la aplicación o se reinicia el juego, las preguntas acertadas o falladas se quedan almacenadas.

Explorar

Esta opción te permite importar preguntas de los módulos SCORM. Desde el menú principal pulsamos explorador, se muestra la estructura de directorios del dispositivo Android, pulsamos Sdcard que es la tarjeta externa, buscamos el fichero comprimido del módulo que hemos importado desde la plataforma. Pulsaríamos sobre ese fichero y le importaríamos.

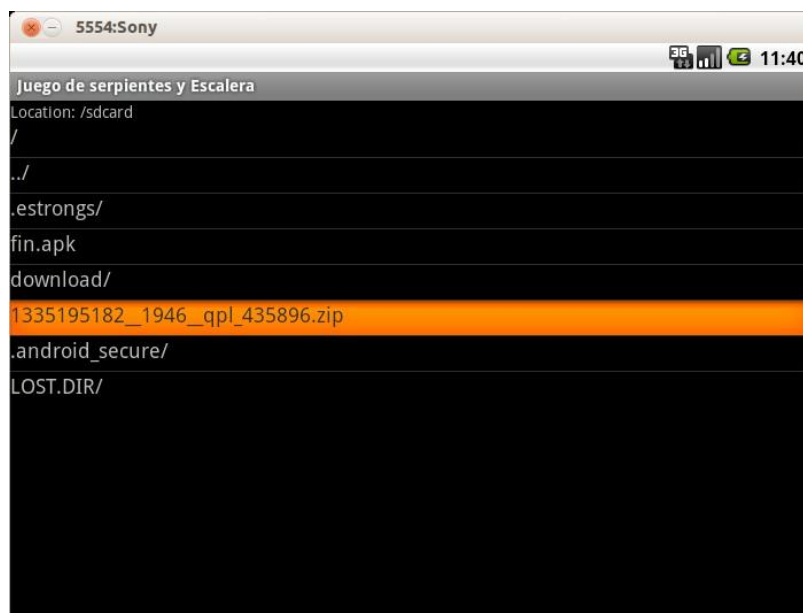


Ilustración 31 Transferir preguntas módulos SCORM a nuestro Recurso educativo

Ver Ranking

En esta opción se muestra el ranking de los jugadores. Cuando se presiona aparece un listado de todos los jugadores del sistema con su nombre, preguntas acertadas y falladas.

Si pulsamos en un jugador saldrán todas las preguntas falladas del jugador.

Anexo I. Empaquetación de una aplicación y firmar digitalmente.

Este proceso es necesario, ya que nuestra aplicación está desarrollada para Android y solo funcionará para un dispositivo real en el cual podamos probar nuestra aplicación. Sin embargo, si queremos distribuirla, tenemos que generar el paquete y firmarlo digitalmente, con esto conseguimos dos cosas. Por un lado que nuestra aplicación funcione en todos los dispositivos a los que va dirigida y segundo, tener la seguridad de que la aplicación pertenece a alguien en concreto. [3]

Para empaquetar la aplicación Android hay que seguir varios pasos:

1. Preparar y ejecutar la aplicación, comprobar que está exenta de errores.
2. Configurar correctamente el archivo manifest de Android. En este paso habría que comprobar correctamente el número de versión, si el modo debuggable está desactivado, si está definido el nombre que debería ser corto para así facilitar, el icono de la aplicación debería de estar a varias resoluciones. Comprobar que se usa la etiqueta uses SDK para ver que versión se va utilizar de esta plataforma. Meter restricciones con cuidado para no restringir el uso de la aplicación en los dispositivos móviles. Verificar todos los permisos que requieran la aplicación.
3. Empaquetar y firmar digitalmente la aplicación, en este paso se empaqueta la aplicación en un fichero .apk. que se firma digitalmente con una clave privada que corresponde unívocamente al programador se utiliza para generar confianza entre usuarios y programadores,

Google actualmente para publicarlo en la playstore la firma necesita que sea válida hasta después de octubre de 2033.

4. Cuando esté empacado pues instalarlo en los dispositivos y probarlo exhaustivamente.

Vamos a verlo con el ejemplo de nuestra aplicación.

Primer paso lo hemos compilado con eclipse y no ha salido ningún error. En el siguiente paso configuraremos el Android-Manifest, para su posterior publicación. En esta pantalla es donde definimos las actividades que forman la aplicación, así como su orientación y otros parámetros. Se recomienda parámetro debuggeable que este a falso para impedir que se descompile y se pueda copiar código de la aplicación.

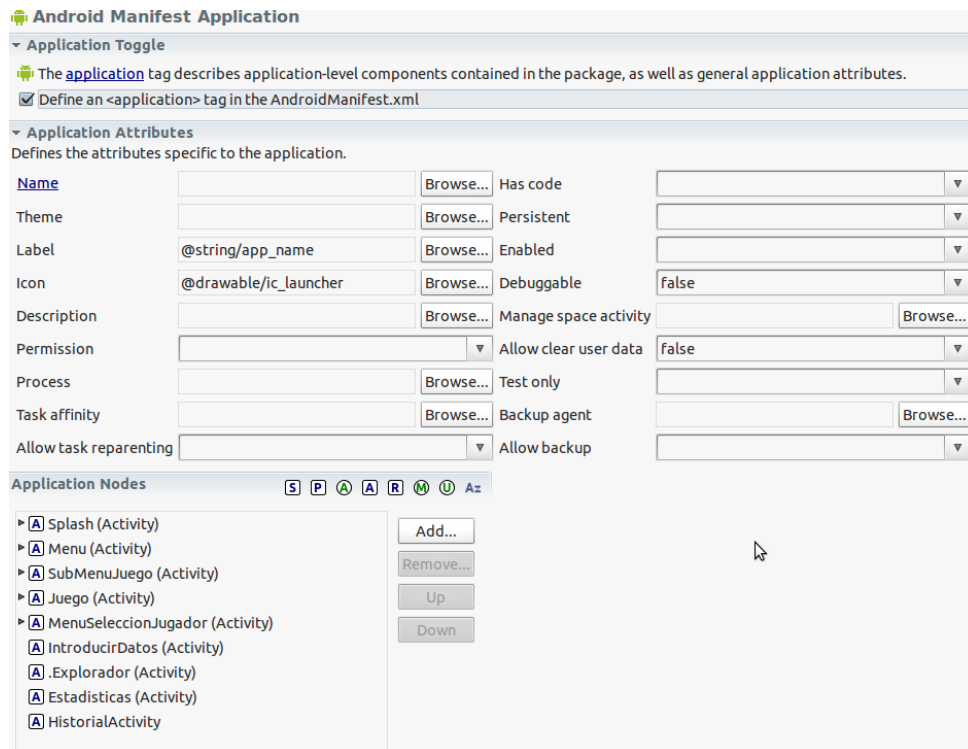


Ilustración 32 Configuración del archivo de manifest

Configuramos los permisos que tiene la aplicación sobre el terminal, que nuestro caso van a ser dos Wake_Lock que consiste en que evita el terminal que entre modo suspensión y external_storage este permiso es para escribir en un dispositivo de almacenamiento masivo.

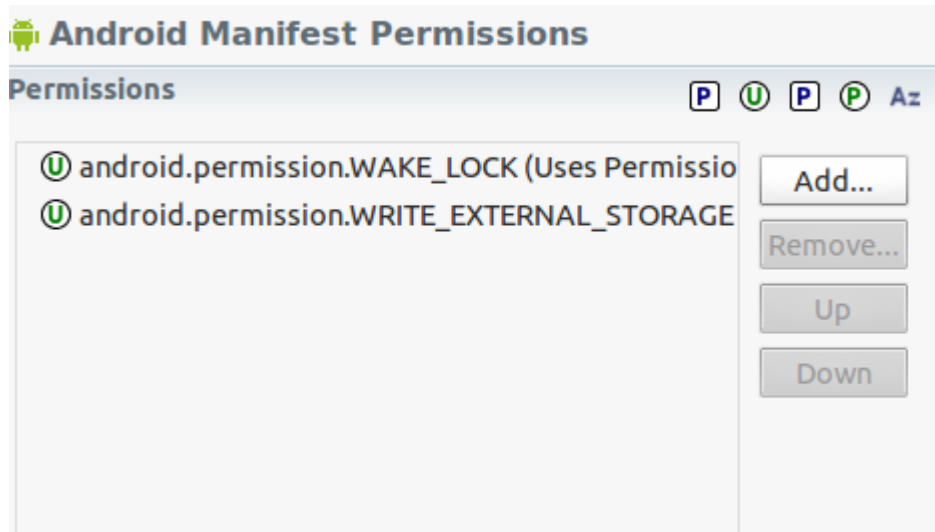


Ilustración 33 Permisos de la aplicación

En el siguiente paso firmaremos digitalmente la aplicación con eclipse lo que nos permitirá instalar la aplicación en cualquier terminal Android. Seleccionamos el proyecto y clicamos sobre el botón derecho del ratón nos saldrá un menú parecido a éste.

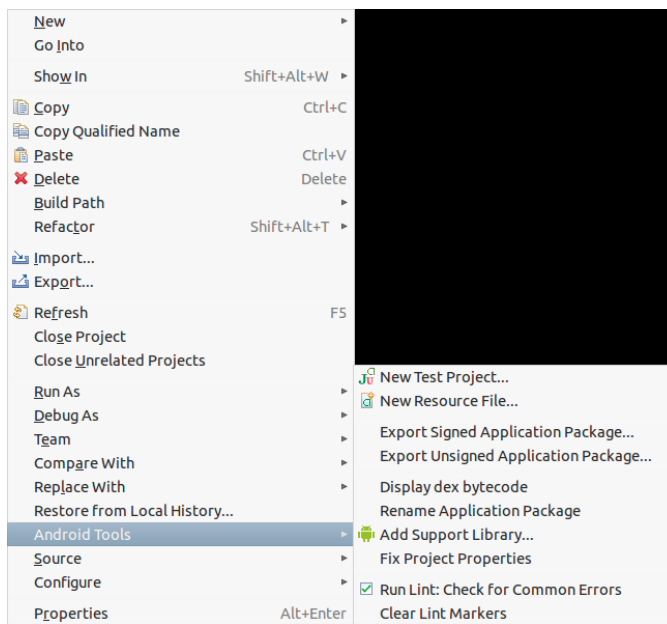


Ilustración 34 Menú Android Tools

Seleccionamos Export Signed Application Package..., aparecerá un cuadro de Dialogo pidiéndonos el proyecto que queremos firmar, seleccionamos el proyecto y damos siguiente aquí nos aparecerá.

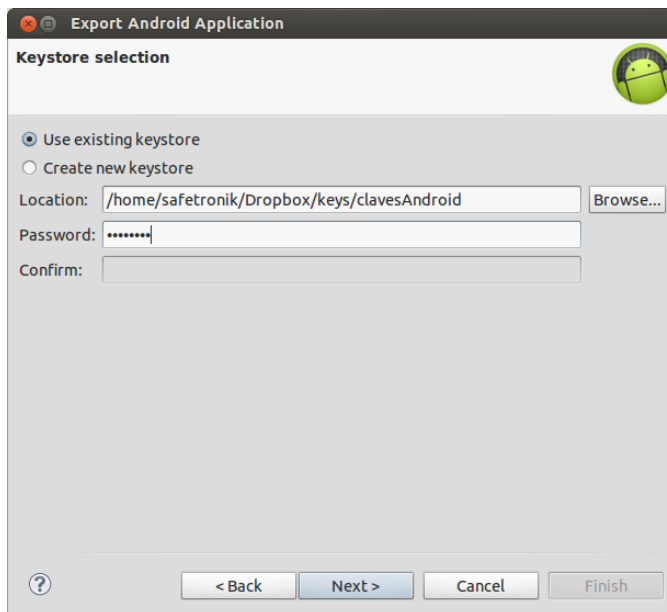
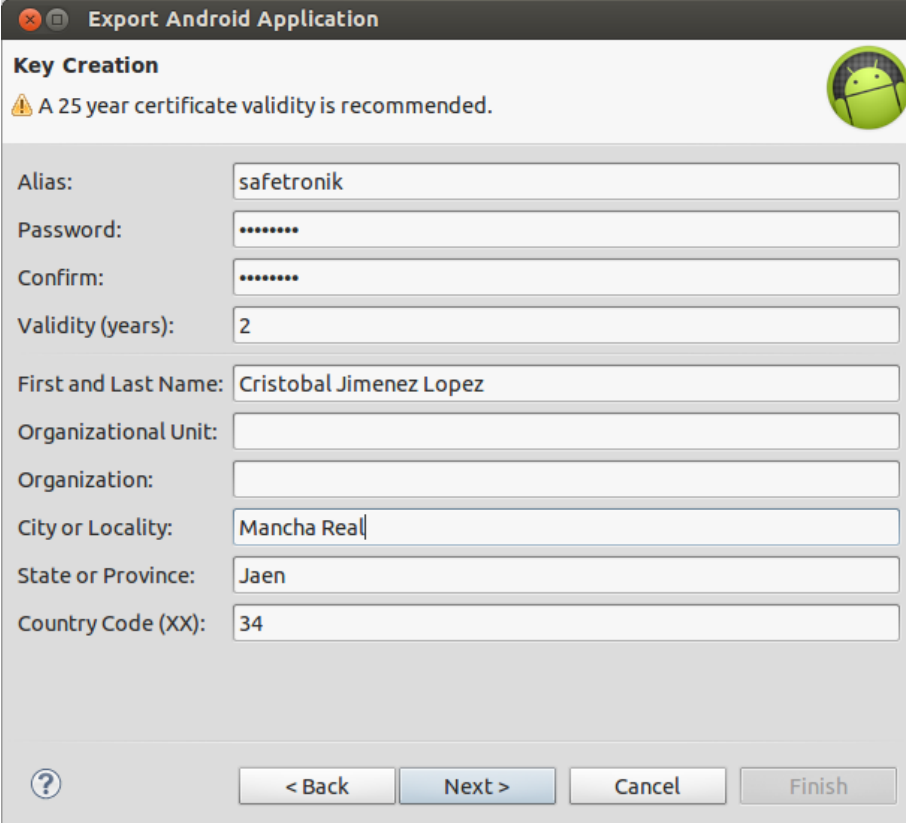


Ilustración 35 Selección de firma

Aquí seleccionamos la clave privada previamente creada, sino seleccionaremos create new keystore, rellenaremos los campos y automáticamente se nos crea la clave privada, si esta clave cayera en otras manos podría suplantar perfectamente la entidad del programador y publicar actualizaciones como malware o si se perdiera no se podría actualizar la aplicación en cuestión. Por eso se aconseja mantener la clave privada a salvo.



Export Android Application

Key Creation

⚠ A 25 year certificate validity is recommended.

Alias: safetronik

Password:

Confirm:

Validity (years): 2

First and Last Name: Cristobal Jimenez Lopez

Organizational Unit:

Organization:

City or Locality: Mancha Real

State or Province: Jaen

Country Code (XX): 34

? < Back Next > Cancel Finish

Ilustración 36 Creación de llave para publicación

Después de esto nos generara un fichero.apk que se puede instalar en cualquier dispositivo y no habrá problema ninguno.

Anexo II, Instalación de la plataforma ILIAS

Para la instalación de la plataforma ILIAS en cualquier servidor web es necesario que el servidor que tenga instalado Apache2, php 5 con soporte XSL y XML.

Para este propósito se ha creado una máquina virtual con VirtualBox Oracle, el sistema operativo elegido es Ubuntu 12.04 con una dirección ip estática (192.168.1.10) apache 2, MySql, Php 5 y Vsftpd.

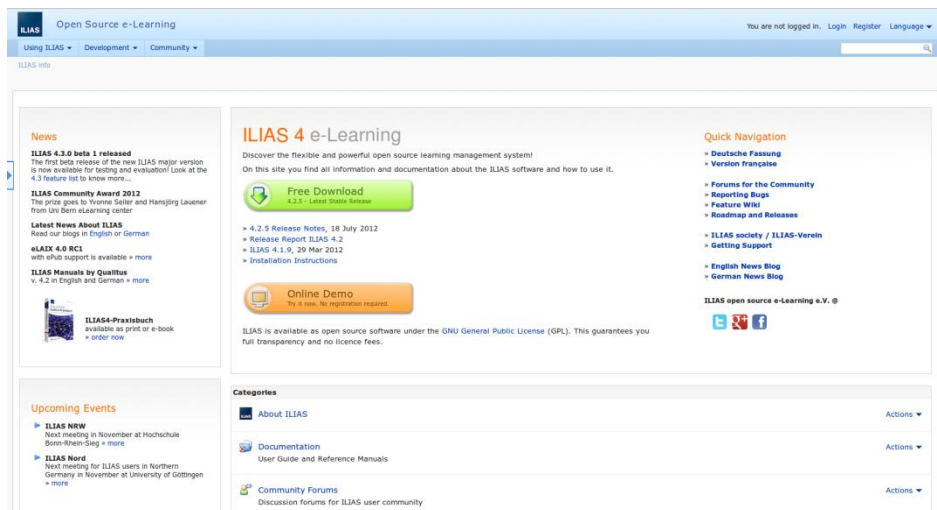


Ilustración 37 Página oficial de Ilias

De la página <https://www.ilias.de/> nos bajamos la plataforma ilias en un fichero comprimido en nuestro caso es la versión 4.2.5 la versión estable.

Descomprimimos el fichero descargado anteriormente en nuestro caso sería ilias-4.2.5.zip.

Subimos la instalación al servidor.

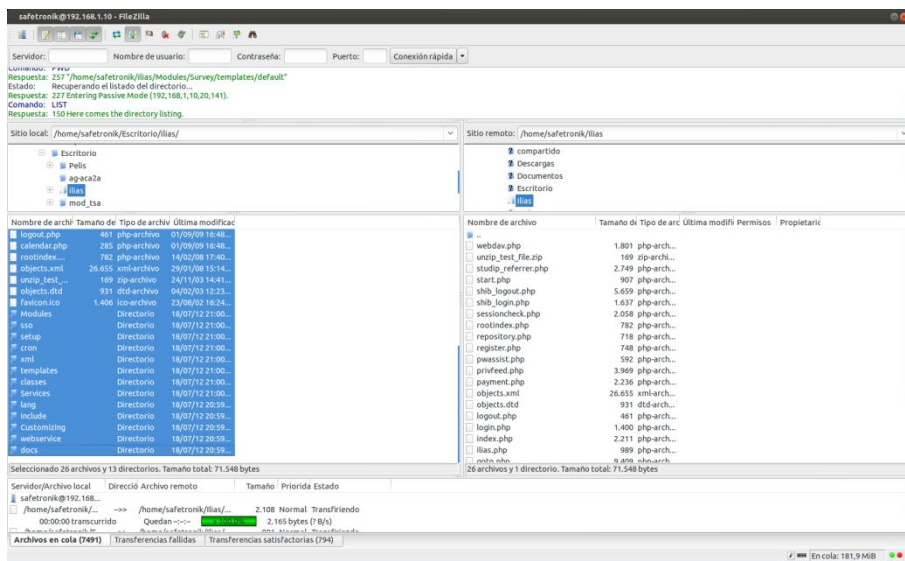


Ilustración 38 Subida de archivos al servidor

En el navegador accedemos a la dirección a la Url donde hemos copiado los archivos en nuestro caso la Url es <http://192.168.1.10/Ilias>.

Nos mostrará todos los requisitos que debe de cumplir para instalar la plataforma Ilias.



Ilustración 39 Pantalla de los requisitos para la instalación

Pulsaremos sobre el botón de instalación, que nos pedirá los parámetros para instalar la plataforma ILIAS. Introduciremos los parámetros que nos piden. Los parámetros son los siguientes Directorio de trabajo, directorio de datos fuera de la plataforma, archivos de logs, Zona de horaria, ruta convert, el directorio de trabajo de

los programas zip y unzip, la ruta de Java, estos parámetros son los más usuales. Para instalar la plataforma y que no dieran problemas tienes que las carpetas seleccionadas tengan permisos de lectura y escritura del usuario de apache en nuestro caso www-data.

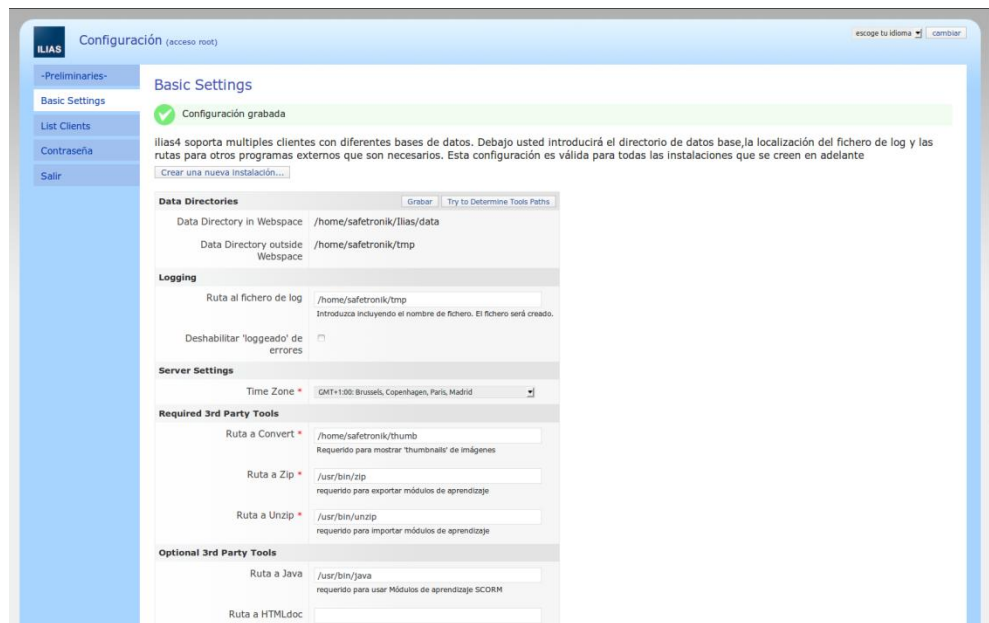


Ilustración 40 Configuración de la plataforma en la instalación

Nos pedirá información sobre la bases de datos, tanto la dirección donde está el nombre de la bases de datos, así como su contraseña. Posteriormente marcaremos que nos cree las bases de datos ya que en nuestro caso se trata de una instalación limpia. Automáticamente nos creara las tablas asociadas a la plataforma.

En el siguiente paso nos pedirá el idioma que queremos en nuestra plataforma y posteriormente nos preguntara información relativa a la información del contacto.

Y por último nos pedirá si deseamos crear el Nic para compartir información del Ilias con otras plataformas Ilias y muestra el usuario y la contraseña, en nuestro caso el usuario es root y la contraseña homer.

Anexo III, Creación e exportación de preguntas desde la plataforma Ilias.

Una vez instalada nuestra plataforma vamos a loguearnos con nuestro usuario y contraseña, en nuestro caso sería el usuario root y contraseña homer. Se nos mostraría nuestro escritorio personal, en la pantalla principal seleccionaremos la opción de espacios y cliccaremos en Añadir nuevo elemento.

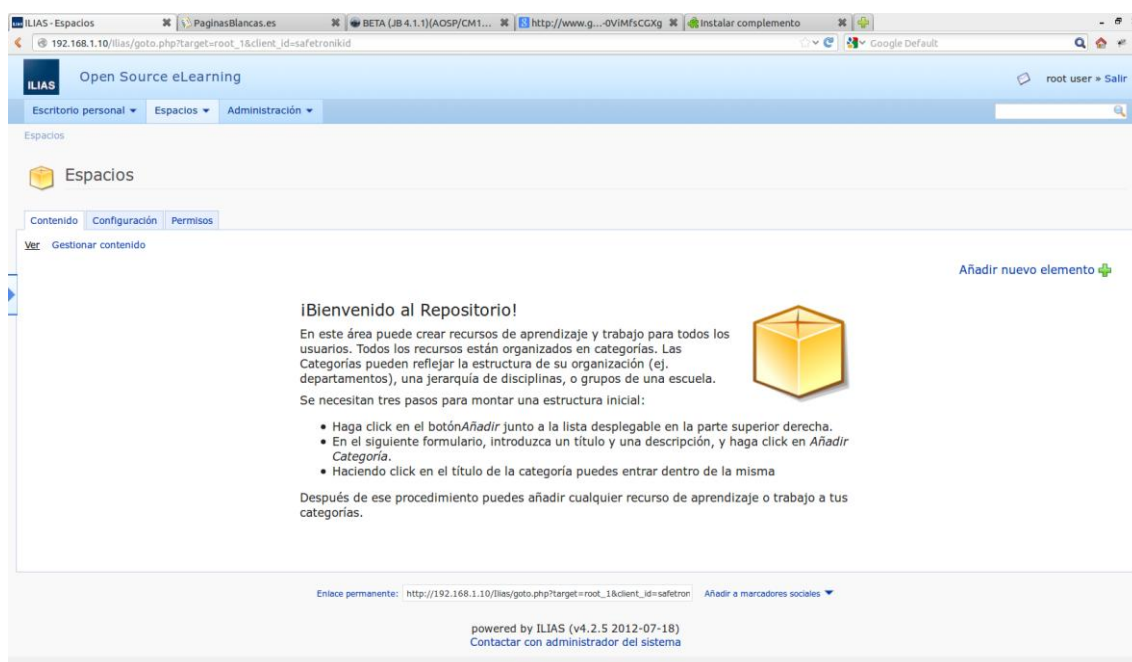


Ilustración 41 Pantalla de Espacios

Seleccionamos la opción añadir test, la plataforma te pedirá el Nombre para el cuestionario Tipo test.

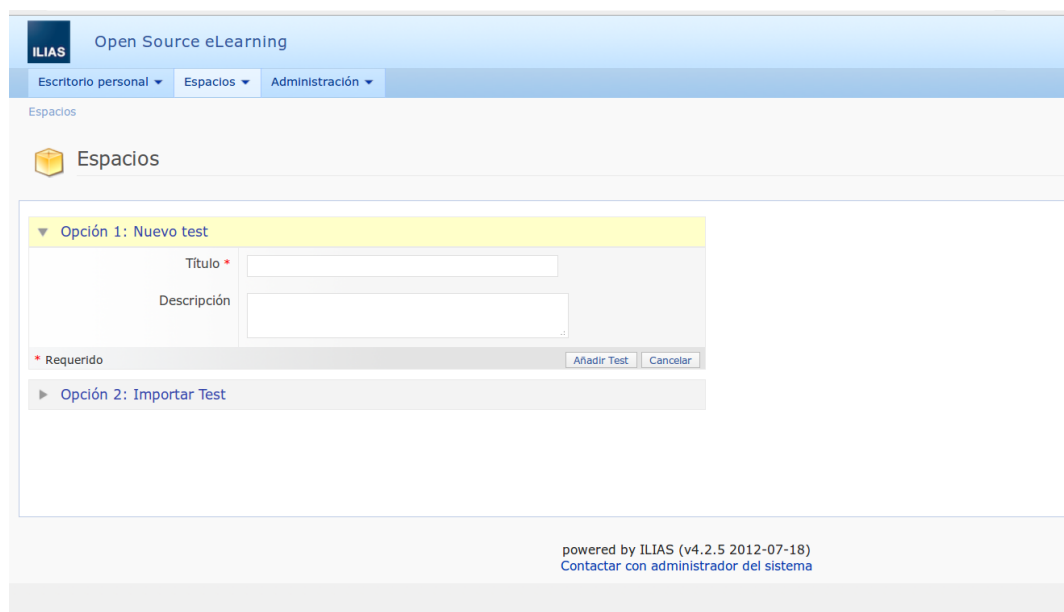


Ilustración 42 Pantalla para dar de alta nuevo Test

Introduciremos el título y la descripción de nuestro cuestionario.

El sistema nos mostrará el contenido del tipo test, que está vacío porque se ha creado desde cero. Creamos una pregunta de opción múltiple y respuesta única. Rellenamos los campos necesario que sería el título de la pregunta, el usuario quien ha escrito la pregunta y la pregunta. A la hora de puntuar las preguntas correctas seleccionaríamos puntuación mayor igual que 1 y respuesta falsas con cero. Pulsaremos guardar y volver a la pantalla anterior.

Estado: Desactivado

◀ Banco de preguntas (Test) | Editar contenido | Previsualizar | Editar propiedades | Feedback | Solución recomendada | Estadísticas

Pregunta de opción múltiple (Respuesta única)

Título *

Autor *

Descripción

Pregunta * ¿Cuántas capas tiene el modelo OSI?

Tiempo de trabajo Horas: 0 Minutos: 1 Segundos: 0

Barajar respuestas

Tipos de respuesta Respuestas de una línea

Tamaño de la miniatura
 Valor mínimo: 20
 Si introduce un tamaño de miniatura, una miniatura de este tamaño se generará para las imágenes cargadas. Sin un tamaño de miniatura, el tamaño de la imagen original es el usado.

respuestas *	Texto respuesta	Imagen respuesta	Puntos	Acciones
<input type="text" value="5 capas"/>	<input type="text"/>	<input type="text"/>	1	<input type="button" value="Examinar..."/> <input type="button" value="Enviar"/>
<input type="text" value="4 capas"/>	<input type="text"/>	<input type="text"/>	0	<input type="button" value="Examinar..."/> <input type="button" value="Enviar"/>

Por favor tenga en cuenta el tamaño límite de archivos de 20M. Tipos de archivos permitidos: .jpg, .jpeg, .png, .gif

* Requerido

Ilustración 43 Pantalla para añadir pregunta

Redes

Preguntas | Info | Configuración | Vista de Impresión | Meta Datos | Exportar | Permisos

Pregunta de opción múltiple (Respuesta única) create

(1 - 1 de 1) Mostrar filtro Columns

Título	Descripción	Tipo de pregunta	Puntos (1)	Estadísticas	Autor	Creado	Ultima actualización
Redes		Pregunta de opción múltiple (Respuesta única)	1		root user	Hoy	Hoy

(1 - 1 de 1)

Ilustración 44 Lista de preguntas que conforman el tipo test

Para exportar todas las preguntas seleccionaríamos las preguntas a exportar y seleccionamos en el menú desplegable seleccionamos exportar y después ejecutar el navegador nos dirá donde guardar nuestro fichero .zip.

ILIAS Open Source eLearning root user » Salir

Escritorio personal Espacios Administración

Espacios » Redes

Redes

Preguntas Info Configuración Vista de Impresión Meta Datos Exportar Permisos

Pregunta de opción múltiple (Respuesta única) crear

(1 - 1 de 1) Mostrar filtro Columnas

Título	Descripción	Tipo de pregunta	Puntos (1)	Estadísticas	Autor	Creado	Ultima actualización
Redes	Previsualizar	Pregunta de opción múltiple (Respuesta única)	1		root user	01. Oct 2012	01. Oct 2012

(1 - 1 de 1)

powered by ILIAS (v4.2.5 2012-07-18)
[Contactar con administrador del sistema](#)

Ilustración 45 Pantalla con preguntas introducidas

Bibliografía

1. **Khan, Badrul Huda.** *Managing E-Leaqrning:Design,Delivery, Implementation and Evañuation.* 1997.
2. **Fundesco.** *Teleformacion: Un paso mas en el camino de la Formacion continua.* Madrid : s.n., 1998.
3. Fundacion Dialnet. [En línea] http://campus.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_garcia_penalvo.htm.
4. Wikipedia. [En línea] 2012. http://es.wikipedia.org/wiki/Aprendizaje_electr%C3%B3nico_m%C3%B3vil.
5. **M.V.Zelkovitz.** *Principles of Software Engineering and Design.* s.l. : Prentice Halls, 1979.
6. *Software Engineering.* **Bohem, B.W.** 1976, IEEE Trans. Computers, págs. 1226-1241.
7. **Rogers, RICK.** *Learning Android game programming.* INIDIA, ESTADOS UNIDOS : ADISSON WESLEY, 2011.
8. Ciclo de vida de una actividad. Diario de un proyecto android. *Diario de un proyecto android.* [En línea] <http://telekita.wordpress.com/2012/02/03/ciclo-de-vida-de-una-activity/>.
9. Android Data Abstraction Framework. Api. *Android Data Abstraction Framework.* [En línea] <http://api.adaframework.com/>.
10. **IEEE.** *Draft Standard for Learning Object Metadata.* Julio de 2002.

11. **LAUREN DARCEY, SHANE CONDER.** *Android 4* . s.l. : Anaya Multimedia, 2012. 978-84-415-3194-9.
12. Foro de desarrollo de aplicaciones para Android. *Foro de desarrollo de aplicaciones para Android.* [En línea] <http://stackoverflow.com/questions/tagged/android>.
13. **GOOGLE.** Documentacion oficial de la plataforma Android. *Documentacion oficial de la plataforma Android.* [En línea] <http://developer.android.com/legal.html>.
14. **Andengine.** Andengine. *Documentacion sobre andengine.* [En línea] <http://www.andengine.org>.
15. **F. Feito Higuera, J. Ruiz de Miras, A. Molina Aguilar.** *Analisis y gestión de datos.* F. Feito Higuera, J. Ruiz de Miras, A. Molina Aguilar : s.n., 1996.
16. **Ilias.** Ilias Open source e-Learning. *Ilias Open source e-Learning.* [En línea] 10 de 07 de 2012.
17. **Brown, Judy.** “*Mobile Learning ¿el futuro del aprendizaje?*” *Learning Review.* 2010.
18. **Salinas, J.** “*La gestión de los entornos virtuales de formación*”. *Seminario Internacional de la Calidad de la Formación en Red en el Espacio Europeo de Educación Superior, Tarragona.* 2005.
19. **Jimenez, J. Roberto.** *Ingeniería de Software I.*

20. **M. Espinilla, I. Palomares, L. Martínez.** *Desarrollo y empleo de juegos educativos on-line destinados al auto-entrenamiento y auto-evaluación.* 2010.
21. **Innoware, Delacroy.** *Estudio de mercado de las tecnologías móviles.* 2010.
22. **A. Villoria, G. Martín, J. M. de las Cuevas,.** *Juegos educativos en dispositivos móviles.* 2009.
23. **E. Freeman, K. Sierra, B. Bates.** *Head First Design Patterns.* s.l. : O'Reilly.
24. **A. Rodríguez Andara, y C. Lozano Salas.** *Evaluación del aprendizaje a través de exámenes tipo test.* s.l. : Universidad de Pais Vaso, 2006.