



Escuela  
Politécnica  
Superior

# Lector SCORM para Android



Grado en Ingeniería Multimedia

## Trabajo Fin de Grado

Autor:

Ana García Domene

Tutor/es:

Pedro Pernías Peco

Septiembre 2014



Universitat d'Alacant  
Universidad de Alicante

---

# Índice general

<b>1. Introducción</b>	<b>5</b>
<b>2. Planteamiento del problema</b>	<b>7</b>
2.1. Problema educativo . . . . .	7
2.1.1. Mobile learning: Aprendizaje ubicuo . . . . .	7
2.1.2. Open Social Learning . . . . .	8
2.1.3. Open content . . . . .	8
2.1.4. OpenCourseWare (OCW) . . . . .	9
2.1.5. Recursos educativos y multimedia . . . . .	10
2.2. Problema tecnológico . . . . .	11
2.2.1. Tecnologías de reutilización . . . . .	11
2.2.2. SCORM . . . . .	11
2.3. Estado actual: Necesidad de un lector SCORM . . . . .	14
<b>3. Objetivo general</b>	<b>16</b>
<b>4. Alternativas existentes</b>	<b>17</b>
<b>5. Metodología</b>	<b>20</b>
<b>6. Desarrollo</b>	<b>21</b>
6.1. Android . . . . .	21
6.1.1. Funcionalidades . . . . .	21
6.1.2. Lenguajes de programación . . . . .	24
6.2. SCORM . . . . .	25
6.2.1. Contenidos y Empaquetado . . . . .	25
6.2.2. Run-Time Environment RTE . . . . .	29
6.2.3. Secuenciación . . . . .	29
6.3. Hardware, Software y API's . . . . .	30
6.4. Diseño de la aplicación. Interfaz . . . . .	31
6.5. Implementación . . . . .	34

6.5.1. Manifiesto de la aplicación Android . . . . .	35
6.5.2. Actividades que forman la aplicación . . . . .	35
6.6. Pruebas/testing . . . . .	41
6.7. Difusión y comercialización . . . . .	45
6.7.1. Obtención del archivo .apk firmado . . . . .	45
<b>7. Conclusiones</b>	<b>47</b>
<b>Bibliografía</b>	<b>50</b>

---

# Índice de figuras

2.1. Ejemplo de recursos educativos multimedia . . . . .	10
4.1. Captura de la aplicación i-Scire. . . . .	18
6.1. Crecimiento de Android en activaciones de dispositivos . . . . .	22
6.2. Número relativo de dispositivos que ejecutan una determinada versión de la plataforma Android. . . . .	24
6.3. Estructura descrita en un manifiesto SCORM de un paquete PIF. . . .	27
6.4. Visualización de la pantalla de inicio . . . . .	31
6.5. Visualización de un recurso SCORM según la orientación del dispositivo	32
6.6. Icono de la aplicación . . . . .	33
6.7. Fondo de la aplicación. . . . .	33
6.8. Arquitectura de la aplicación. . . . .	34
6.9. Listado de máquinas virtuales. . . . .	41
6.10. VM Android 3.0 . . . . .	41
6.11. VM Android 4.4 . . . . .	42
6.12. Pruebas de un recurso SCORM con dispositivo en apaisado . . . . .	43
6.13. Pruebas de un recurso SCORM con dispositivo en vertical . . . . .	44
6.14. Primeros pasos de exportación firmada . . . . .	45
6.15. Datos del certificado y exportación firmada . . . . .	46

---

# Capítulo 1

## Introducción

Año tras año nuestra vida se va haciendo más tecnológica y cada vez más personas acceden a las nuevas tecnologías en la llamada sociedad de la información, consolidada con la globalización resultante del fenómeno Internet.

Debido a la naturaleza curiosa y ávida de conocimientos del ser humano, las tecnologías educativas, son un gran apoyo para la educación y sus fines, con gran éxito al ofrecer personalización e interactividad para los usuarios.

Las tabletas son dispositivos ideales para la docencia virtual por sus prestaciones. Esto es así porque permiten el uso de contenido multimedia en cualquier parte, ofreciendo al usuario información enriquecida y formación interactiva, gracias a la existencia de numerosos repositorios educativos. Dichos repositorios albergan multitud de recursos educativos de cualquier tipo de campo, accesibles para el usuario al configurarse cumpliendo unas pautas reconocibles por aplicaciones especializadas en educación, como apoyo a la educación presencial o como alternativa, ya sea enseñanza a distancia o incluso autodidacta.

Dentro del contexto de las tablets Android, existen miles de aplicaciones educativas, más de 90.000 en la categoría educación de la plataforma comercial de Google [15]. Debido a esto y la importancia de las nuevas formas de educación existe Google Play for Education [28], que además de reunir libros y vídeos educativos con aplicaciones para Chrome o Android específicas para Chromebooks y tablets, ofrece un sistema más especializado para la búsqueda adecuada de contenidos e integración con otras herramientas de Google que pueden mejorar la experiencia educativa.

Sin embargo, pese a la gran cantidad de recursos educativos existentes, si nos centramos en aplicaciones para Android con la etiqueta SCORM (Sharable Content Object Reference Model) el número se reduce drásticamente hasta 12, y de éstas, como vere-

mos con más profundidad, sólo unas pocas permiten reproducir recursos educativos en formato SCORM, algunas dependiendo de un LMS y ninguna sin conexión a Internet.

La creación de ésta aplicación centrada en usuarios de tablets, ofrecerá a los usuarios de Android acceder a recursos educativos allá a donde vayan, con la ventaja de no depender de un sistema LMS (Learning Management System) específico para poder acceder a los contenidos de un curso que se encuentre bajo el estándar SCORM, además de contar con integración con redes sociales (ej. Twitter) y la posibilidad de compartir sus progresos a través de ellas.

---

# Capítulo 2

## Planteamiento del problema

### 2.1. Problema educativo

#### 2.1.1. Mobile learning: Aprendizaje ubicuo

La proliferación de plataformas móviles, en concreto, los tablets o ultra-pc, permite que el aprendizaje haya dejado de ser una tarea que sólo se realiza en el aula o en la pantalla de un ordenador en un despacho o entorno de trabajo controlado. De hecho, cualquier lugar en el que podamos disponer de un dispositivo conectado puede convertirse en un lugar de trabajo.

Además de la proliferación de los tablets o ultra-pc, existen numerosos repositorios de materiales educativos que pueden ser descargados en formato estándar (SCORM) y que permiten aprender siguiendo las especificaciones que el autor del recurso haya establecido o reutilizarlos, modificándolos o no, para la integración en nuevos recursos.

La convergencia de las nuevas tecnologías móviles y la disponibilidad de recursos educativos de e-learning, han dado lugar a una nueva forma de educación en la que el usuario recibe más protagonismo, es él quién elige los recursos según sus necesidades, en el momento que lo necesita, explorando y solicitando la información necesaria **esté donde esté**, pues el usuario es el centro de este nuevo tipo de educación. La información recibida es inmediata y en la mayoría de los casos muy visual, realista e incluso en un formato lúdico y entretenido. Estas características fomentan la investigación y el autoaprendizaje en el estudiante.

Para poder acceder desde cualquier lugar y momento, el usuario utiliza plataformas de comunicación móviles así como tablets, teléfonos móviles, o ultra-pcs lo que presenta problemas de portabilidad y de interfaz de usuario.

### 2.1.2. Open Social Learning

El OSL (Open Social Learning) se define como un nuevo tipo de educación, una nueva forma de entender la enseñanza y aprendizaje que surge gracias a las nuevas tecnologías, reuniendo un conjunto de diferentes herramientas digitales, condiciones y materiales que permiten el acceso a todo tipo de conocimiento a través de Internet y más específicamente en la llamada web 2.0

Con la aparición del e-learning, muchos centros e instituciones han querido apostar por enseñanza online, el número de autodidactas que aprenden de la gran cantidad de contenidos que pueden obtener a través de Internet ha ido aumentando, por ello OSL no se refiere a un tipo de educación tutorizada o formal e incluye los términos:

- **Abierto:** por el que los contenidos pueden ser utilizados, copiados, modificados, redistribuidos, etc. para que todos los usuarios puedan beneficiarse de ellos de forma flexible y personalizable,
- **social:** aludiendo al concepto de cooperación y participación que existe entre los usuarios de Internet incluyendo tanto la creación y actualización de los contenidos como de su uso y aprendizaje por toda la comunidad de las llamada web social
- **y aprendizaje:** que principalmente indica la importancia de centrarse en aprender y en las necesidades del alumno y los recursos y tecnologías que éste tiene a su alcance.

### 2.1.3. Open content

Open content (OER's) o Contenido Abierto se refiere a los contenidos cuya licencia de uso permite la reutilización bajo condiciones más abiertas que las habituales. Los contenidos pueden ser más abiertos o menos según las restricciones de la licencia impuesta mediante las actividades definidas por las 5R [20]:

- **Retener:** El derecho de hacer, poseer, descargar, duplicar, almacenar y administrar contenidos,
- **reutilizar:** el derecho a utilizar el contenido en una clase, en un grupo de estudio, en un sitio web, en un video o de otras formas,
- **revisar:** el derecho a actualizar, adaptar, ajustar, modificar o alterar el contenido según las necesidades,
- **remezclar:** el derecho de combinar los contenidos con otro tipo de contenido

abierto para crear algo nuevo,

- y **redistribuir**: el derecho a compartir copias del contenido original, sus revisiones, o sus combinaciones con otros usuarios.

Por lo tanto, el contenido será más abierto cuantas más actividades de las 5R tenga permitidas el usuario, también afectan los requisitos para realizar estas actividades, por ejemplo la licencia que deben tener los nuevos contenidos.

Una forma muy utilizada y conocida para expresar estos derechos es la distribución de contenidos mediante las licencias Creative Commons [18], una organización sin ánimo de lucro destinada a permitir publicaciones abiertas del trabajo de autores y creadores voluntarios mediante este tipo de licencias.

#### 2.1.4. OpenCourseWare (OCW)

OpenCourseWare o educación abierta abarca recursos, herramientas y prácticas dedicadas a para mejorar el acceso y la eficiencia de la educación de forma abierta. Esto es posible a la mentalidad humana de intercambio de conocimientos, colaboración entre usuarios y la capacidad de creación con la tecnología actual, condiciones que permiten la creación de una inmensa fuente de recursos educativos compartidos de forma abierta.

La educación está basada en el hecho de compartir ideas, información y todo tipo de conocimientos con los demás de forma abierta, así como complementar el propio conocimiento y el de los demás con la comprensión y habilidades conjuntas, este acto se intensifica con el poder de Internet, ya que el contenido es accesible en todo el mundo, difundiéndose de forma rápida y libre.

Gracias a éstas características, el movimiento de educación abierta aumenta las oportunidades de educación en todo el mundo, con modelos para promover los recursos abiertos, la tecnología, las prácticas de enseñanza en la educación y su impacto en la educación global como:

- El Consorcio de Educación Abierta (**Open Education Consortium**) [33], una comunidad mundial de la que forman parte cientos de instituciones de educación superior y organizaciones asociadas
- o la **Declaración de Educación Abierta de Ciudad del Cabo (The Cape Town Open Education Declaration)** de septiembre de 2007 [34], que al mismo tiempo es una declaración de principios, una declaración de estrategia y una declaración de compromiso firmada por cientos de estudiantes, educadores,

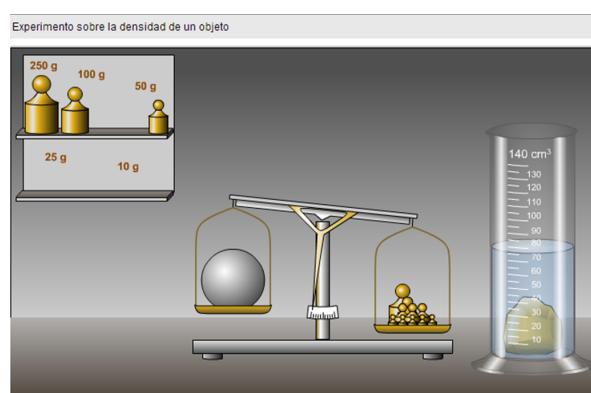
formadores, autores, escuelas, colegios, universidades, editores, sindicatos, profesionales, responsables políticos de las sociedades, gobiernos, fundaciones y otras iniciativas de educación abierta afines de todo el mundo.

### 2.1.5. Recursos educativos y multimedia

Hoy en día, la mayoría de recursos educativos contiene información multimedia, es decir, incluyen materiales como texto, audio, imágenes, videos y animación creando una composición de contenidos que conforman materiales enriquecidos.

El manejo de diferentes contenidos multimedia presenta dificultades tecnológicas por sus características como el tamaño de los recursos, la cantidad de formatos o el nivel computacional que requiere su uso y presentación. Sin embargo, este tipo de contenidos agregan valor a los materiales educativos (interactivo, audiovisual, etc.), además se ha evidenciado que son útiles aplicados al e-learning.

La presentación de recursos de aprendizaje multimedia mediante diferentes medios y tecnologías facilita la interacción y la presentación de materiales enriquecidos de forma personalizada, según las necesidades del alumno, de modos diferentes a los habituales, junto con la inclusión de elementos sociales, logra aumentar la implicación y motivación del estudiante, lo que lleva a una mayor eficacia del proceso de aprendizaje.



FERNANDO POSADA - CANALTIC



DIGITALAVMAGAZINE

- (a) Simulación de experimento sobre la densidad de un objeto. Posada [38] (b) Realidad Aumentada educativa con Geometry101 de Zientia. digitalavmagazine [22]

Figura 2.1: Ejemplo de recursos educativos multimedia

En este contexto, un curso puede estar formado por multitud de recursos multimedia estructurados siguiendo un estándar como SCORM, que permite representar diferentes tipos de recursos al reproducirlos en el navegador y comprime los recursos de forma normalizada para un mejor manejo de éstos. En ocasiones se emplean recursos multimedia

más complejos como juegos educativos, representaciones de modelos o animaciones por realidad aumentada, simuladores, etc. que amplían la coexistencia entre multimedia y educación como los de la Figura 2.1 de la página 10.

## 2.2. Problema tecnológico

### 2.2.1. Tecnologías de reutilización

En muchas ocasiones, a la hora de construir un curso o cualquier forma de información, es muy útil que podamos utilizar recursos ya creados o basarnos en ellos.

Para hacer posible la reutilización de materiales es necesario que estos cumplan una serie de características:

- Que permita ser visualizado de forma apropiada por el mayor número de usuarios posible,
- deben estar en un formato normalizado y empaquetado específico que sea aceptado por la comunidad,

De esta forma se posibilita la universalización de materiales, es decir, su exportación e importación en diferentes plataformas de aprendizaje, y que sea visible para una gran cantidad de personas.

Por otra parte, es muy importante tener en cuenta la gestión de la propiedad intelectual de los contenidos, es decir, bajo qué tipo de licencia de derechos de autor se encuentran. Las licencias que permiten la redistribución de materiales, e incluso su modificación, y con ello la adaptación a cada curso, aumentan las posibilidades de reutilización de materiales.

En cuanto a e-learning, un estándar muy utilizado es el formato SCORM pues facilita la reutilización por su formato de empaquetamiento y de metadatos.

### 2.2.2. SCORM

En 1997 el Departamento de Defensa estadounidense (DOD), bajo la supervisión de la Oficina del Secretario Adjunto de Defensa, estableció la Iniciativa ADL [2] para estandarizar y modernizar la formación y la gestión de la educación con expectativas de facilitar el acceso al aprendizaje y el rendimiento de la más alta calidad, teniendo

en cuenta que puede ser adaptado a las necesidades individuales de forma rentable, en cualquier momento y en cualquier lugar.

Desde esta perspectiva, para abordar los requisitos funcionales de ADL, la accesibilidad, interoperabilidad, durabilidad y reutilización, (los “ilities”) dentro de la formación, educación y comunidades de apoyo al rendimiento del Departamento de Defensa, así como en el gobierno, la academia y industria, la Iniciativa ADL desarrolló SCORM[3].

SCORM o Sharable Content Object Reference Model (modelo de referencia de objetos de contenido compartido) es un conjunto de estándares y especificaciones de empaquetamiento muy utilizado que define la interrelación de los objetos de contenido, modelos de datos y protocolos permitiendo presentar material didáctico estructurado de forma adecuada.

Esta especificación integra una relación de estándares técnicos, especificaciones y pautas diseñadas para promover la accesibilidad, interoperabilidad, durabilidad y reutilización de los contenidos de aprendizaje, mediante un empaquetamiento característico que asegura la importación y reutilización de contenidos en diferentes Sistemas de Gestión de Aprendizaje (Learning Management System - LMS) o plataformas de e-learning que cumplen con el mismo modelo, facilitando así la interoperabilidad entre las plataformas.

Esto es así gracias a las características que implica seguir los requisitos funcionales de ADL:

- **Accesibilidad:** La capacidad para localizar y acceder a los componentes de instrucción desde varias ubicaciones y entregarlos a otros lugares,
- **interoperabilidad:** La capacidad de tomar componentes de instrucción desarrollados en un mismo sistema y utilizarlos en otro sistema,
- **durabilidad:** La capacidad de soportar la evolución tecnológica y / o cambios sin costoso rediseño, reconfiguración o recodificación,
- **y reutilización:** La flexibilidad de incorporar componentes de instrucción en múltiples aplicaciones y contextos.

## Versiones

### SCORM 1.0

La primera versión, SCORM 1.0, se remonta al año 2000. La estructura para describir el contenido se basaba en especificaciones AICC mediante un archivo XML, sin embargo, fue reemplazado en poco tiempo debido a que no presentaba un buen manifiesto o soporte para metadatos.

## SCORM 1.2

Lanzado en 2001, SCORM 1.2 es la versión previa a la integración de la secuenciación.

SCORM 1.2 fue la primera versión con una prueba real de conformidad mediante una serie de pruebas (test suite) y utiliza la especificación IMS de empaquetamiento de contenido, integrando un manifiesto que abarca todo el contenido y un soporte para metadatos que, opcionalmente, presentan un detallado etiquetado de los objetos y assets descritos en el manifiesto, de esta forma describen el curso con una precisión mucho mayor.

## SCORM 2004 (1.3)

Desde su publicación, distintas ediciones de SCORM 1.3, más conocida como SCORM 2004, comenzaron a versionarse con SCORM.

La versión más reciente y la versión actual de SCORM, es SCORM 2004 4<sup>a</sup> Edición (2009) que incluye la posibilidad de especificar secuenciación adaptativa de actividades que utilizan los objetos de contenido y los nuevos estándares para la comunicación de la API, además resuelve muchas ambigüedades.

SCORM 2004 permite compartir y utilizar la información sobre el estado de éxito de múltiples objetivos de aprendizaje o competencias a través de objetos de contenido y cursos para un alumno concreto dentro de un mismo sistema de gestión de aprendizaje.

Actualmente, ADL suministra recursos para SCORM 1.2, SCORM 2004 3<sup>a</sup> edición, y SCORM 2004 4<sup>a</sup> Edición y anima a los desarrolladores a modificar su trabajo para ajustarse a una de estas tres especificaciones aunque recomienda el uso de SCORM 2004 4<sup>a</sup> Edición sobre todas las demás, pues cuenta con más de diez años de retroalimentación, por parte de la comunidad, integrados en su diseño.[3, Overview]

## Tin Can API

Aún en desarrollo, Tin Can API [41], recientemente rebautizada como Experience API or xAPI, se presenta como la siguiente generación de SCORM. Actualmente en su versión 1.0.1 (octubre 2013), Tin Can es una nueva especificación tecnológica de aprendizaje que intenta salvar algunas de las mayores restricciones de SCORM que ahora podrán reconocerse y comunicar a la API:

- El aprendizaje móvil,
- las simulaciones,
- los mundos virtuales,
- serious games,

- las actividades del mundo real,
- el aprendizaje basado en la experiencia,
- el aprendizaje social,
- aprendizaje en línea,
- y el aprendizaje colaborativo.

A través de un vocabulario simple se lleva a cabo la captura e intercambio de actividades, permitiendo la recopilación de datos de una gran cantidad de experiencias de los usuarios tanto en línea como fuera de línea.

ADL como administrador de la especificación, recibe la ayuda de desarrollo de Rustici Software, sin embargo, xAPI está impulsado por la comunidad y su implementación es libre.

### **Common Cartridge**

Desarrollado por la especificación de empaquetado de contenido de IMS (IMS Content Packaging), Common Cartridge [29] es un formato de empaquetado de intercambio de recursos educativos ejecutable en plataformas compatibles con un LMS similar a SCORM. Soporta contenido enriquecido, evaluaciones integradas, foros, metadatos y autorizaciones de contenidos protegidos.

Este formato está en desarrollo y bajo pruebas para desarrolladores con el objetivo de completar el formato e incluir mejoras como la integración Web / empresa con las herramientas y servicios de terceros, planes de lecciones, mapas de competencias basadas en estándares educativos y compatibilidad con la accesibilidad.

## **2.3. Estado actual: Necesidad de un lector SCORM**

Ante la existencia de una inmensa cantidad de contenidos educativos, muchos de ellos abiertos, surge la necesidad de utilizar tecnologías que permitan un acceso fácil y rápido a dichos contenidos.

El hecho de que los contenidos intenten seguir estándares de creación facilita su lectura con tecnologías específicas, sobre todo si éstas también son de acceso libre. Uno de los estándares para recursos educativos más conocidos y utilizados es SCORM, que

facilita la utilización, modificación, integración y reutilización de los contenidos en cursos y procesos de aprendizaje guiados o autodidactas.

Esto, junto al gran crecimiento de las tecnologías móviles que permiten el acceso a contenidos digitales desde cualquier parte incluso sin conexión, han hecho surgir la necesidad de la creación de tecnologías para la lectura de este tipo de recursos en dispositivos como tablets, al ofrecer unas buenas características para la enseñanza. En estos momentos existe una aplicación de lectura de recursos educativos en formato SCORM para i-Pad llamada i-Scire, sin embargo, a pesar de la gran cantidad de usuarios de tablets Android, no existe ninguna de estas características para esta plataforma.

La existencia de una aplicación para Android como esta permitiría a todos los usuarios de tablets Android el acceso a la gran cantidad de recursos educativos en formato SCORM en cualquier parte según sus necesidades de aprendizaje, sin necesidad de conexión a Internet en todo momento y con la oportunidad de compartir sus progresos a través de las redes sociales.

---

# Capítulo 3

## Objetivo general

El objetivo principal del proyecto es la creación de una aplicación para tabletas Android que ejecute contenidos educativos en formato estándar, para poder acceder a lecciones de cursos estructurados en formato SCORM (Sharable Content Object Reference Model).

Para el desarrollo de un SCORM player sobre la plataforma Android se desarrollarán diferentes aspectos como subobjetivos:

- El estado del arte actual sobre el formato educativo SCORM,
- estado del arte actual sobre la plataforma y entorno de desarrollo Android,
- estado del arte actual en cuanto al desarrollo de aplicaciones educativas para ejecutar contenidos docentes en tablets,
- las soluciones y características que ofrecen al usuario dichas aplicaciones al problema educativo definido en apartado anterior,
- actualización de estas características para la inclusión funcionalidades a las ya presentes en otras aplicaciones,
- definición e implementación de una aplicación que reproduzca recursos educativos en formato SCORM,
- testeo de la aplicación,
- relación de conclusiones.

Una vez finalizado el desarrollo del software necesario, la aplicación se ubicará en la plataforma comercial de aplicaciones Android (coste "0") permitiendo su difusión.

---

# Capítulo 4

## Alternativas existentes

El formato SCORM ha tenido varias especificaciones, la última versión estable es SCORM 2004 4<sup>a</sup> edición (2009), que sigue siendo muy utilizado hasta la actualidad. Sin embargo se está desarrollando una nueva especificación, la Tin Can API (2011), recientemente rebautizado como Experience API or xAPI, que con su versión 1.0.1 (octubre 2013) [5] quiere levantarse como la siguiente generación de SCORM aunque aún está en desarrollo y sin consolidar.

Un SCORM player para dispositivos móviles no es un concepto nuevo. Dentro del contexto de las tablets Android, en la plataforma comercial de Google existen aplicaciones educativas relacionadas con el formato SCORM. Concretamente, en la búsqueda de aplicaciones por educación y SCORM nos encontramos con 12 aplicaciones [27] de las cuales sólo algunas reproducen SCORM. Para ser más exactos:

- **Rustici Software Book Scanner** sirve para añadir libros mediante escaneo de ISBN,
- **LM Viewer** es un software de creación de contenidos para e-learning,
- **eXact Mobile** y **eXact Mobile for Achmea** son una extensión de eXact LCMS suite,
- **Tabula Pro Converter** es un conversor de formatos para distintos recursos,
- **Tabula Pro Editor** sirve para crear y editar recursos educativos,
- **Mobile Aim** by Gyrus Systems depende de GyrusAim el LMS de la compañía ,
- **LearningCafe Mobile** funciona con el LMS LearningCafe,
- **BilgiOnline** funciona con Moodle y es exclusiva para alumnos de la Universidad de Bilgi, Estambul,

- **Moodle EduCloud 70:20:10** como indica su nombre también funciona sobre Moodle (Moodle EduCloud),
- **LearnTrak LMS** es un LMS en línea
- y **mTraining** necesita estar conectado a una plataforma de aprendizaje de una empresa

A partir de esto vemos que las aplicaciones que reproducen recursos SCORM precisan de conexión a Internet para su funcionamiento o dependen de un LMS específico, además no son específicas para tablets.

La aplicación desarrollada, al estar pensada para tablets, será similar a un SCORM player para iPad llamado i-Scire[21].



© DEPARTAMENTO DE LENGUAJES Y SISTEMAS INFORMÁTICOS. DLSI-UNIVERSIDAD DE ALICANTE.

Figura 4.1: Captura de la aplicación para iPad i-Scire. Departamento de Lenguajes y sistemas Informáticos. DLSI-Universidad de Alicante [21].

i-Scire es una aplicación dirigida a usuarios que quieran leer y reproducir recursos educativos en formato SCORM. Es una aplicación independiente, no está conectada a ningún LMS y permite, mediante una conexión de twitter, informar del progreso realizado.

Como deja entrever la figura 4.1 i-Scire es una herramienta complementaria para el PLE (Personal Learning Environment) del estudiante, ampliando el entorno de aprendizaje

más allá de la pantalla del ordenador, haciendo posible una verdadera situación de m-learning como complemento para un escenario de e-learning.

Desde la aplicación se puede acceder a los recursos educativos mediante una dirección URL directa o a través de una cuenta de Dropbox para navegar entre los archivos, seleccionar y descargar el archivo que desea reproducir. A su vez, i-Scire está inspirado en una aplicación existente para Windows-CE y Android desarrollado por el Club de Innovación Universitaria de la Universidad Pontificia de Salamanca [17]

---

# Capítulo 5

## Metodología

Para la organización del trabajo a realizar para la aplicación se definen una serie de pasos a seguir durante el estudio y el desarrollo:

- Estudio del problema general
- Estudio y comprensión de SCORM
- Estudio y comprensión de Android
- Implementación de acceso al manifiesto de un SCORM para obtener la organización del recurso
- Visualización del recurso en un entorno web Android con su navegación correspondiente
- Pruebas de funcionamiento básico
- Implementación de la selección de SCORMs desde la aplicación para su visualización
- Pruebas de funcionamiento en la selección
- Personalización de la interfaz de la aplicación
- Estudio y comprensión de la API de Dropbox
- Implementación de la apertura de SCORM desde Dropbox
- Pruebas de acceso a Dropbox y navegación por curso importados
- Creación de documentación para la utilización de la aplicación
- Subida de la aplicación a Google Play

---

# Capítulo 6

## Desarrollo

En los siguientes apartados se describen las características de la aplicación así como las fases de su diseño, implementación y pruebas.

### 6.1. Android

Presente en cientos de millones de dispositivos móviles, en más de 190 países de todo el mundo, Android es un sistema operativo para dispositivos móviles, basado en las contribuciones de la comunidad Linux, de fácil uso altamente personalizable. “Es la mayor base instalada de cualquier plataforma móvil y de rápido crecimiento, cada día otro millón de usuarios enciende sus dispositivos Android por primera vez y empieza a buscar aplicaciones, juegos y otros contenidos digitales” Android Open Source Project [6]. Este crecimiento lo podemos observar en la Figura 6.1 en la página 22

Gracias a la posibilidad de conectar el sistema a nuestra cuenta de Google, podemos acceder a los servicios que nos ofrece, como la plataforma de comercialización de Google a través de la que tenemos disponibles aplicaciones, juegos, vídeos, contenidos, etc. ampliando las posibilidades de uso de nuestros dispositivos.

Android ofrece a los desarrolladores herramientas para crear aplicaciones que se adaptan a las capacidades disponibles en cada dispositivo.

#### 6.1.1. Funcionalidades

En estos momentos la última versión de Android es la 4.4 o KitKat[6, KitKat] que incluye una serie de nuevas características:

- **Nuevas capacidades NFC mediante Emulación Host Card (HCE):** Gracias a dispositivos que soporten la tecnología NFC, se podrán realizar transacciones seguras emulando una tarjeta inteligente.

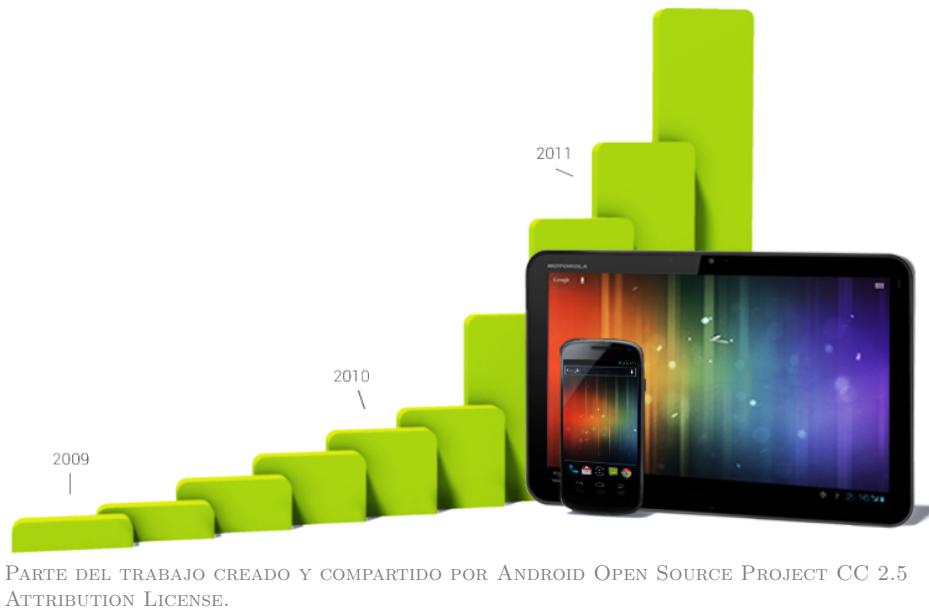


Figura 6.1: Crecimiento de Android en activaciones de dispositivos. Android Open Source Project [6]

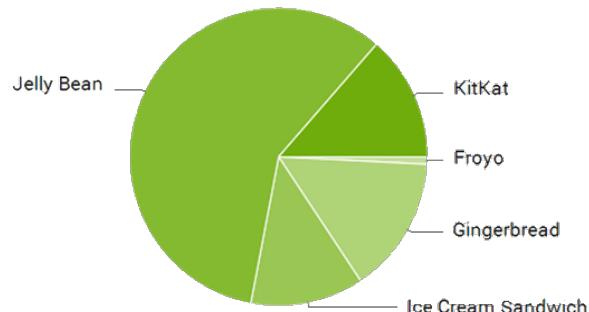
- **Framework de impresión:** Es posible imprimir cualquier tipo de contenido sobre servicios alojados en la nube, como Google Cloud Print, o mediante Wi-Fi controlando factores como la impresora, el tamaño de papel o el rango de páginas a imprimir a través de la plataforma nativa de impresión, accesible desde cualquier app que tenga la impresión habilitada.
- **Framework de acceso de almacenamiento:** El acceso a los archivos independientemente del proveedor de almacenamiento en el que se encuentren se simplifica a través de la nueva interfaz de acceso al almacenamiento local o en la nube.
- **Sensores de baja potencia:** La plataforma de procesamiento por lotes de sensores trabaja con el hardware del dispositivo para comunicar eventos en lote a medida que se detectan, incluso cuando la pantalla está apagada y el sistema está dormido. ésta es una optimización más eficiente, que puede reducir drásticamente el consumo de energía. Esta versión también incluye soporte para dos sensores nuevos, detector de pasos y contador de pasos, implementados para un consumo de energía mínimo. Aunque estos sensores actualmente sólo están disponibles en Nexus 5, se está trabajando para incluirlos lo antes posible en nuevos dispositivos.

- **Proveedor de SMS:** El nuevo proveedor de SMS y APIs definen un modelo de interacción estandarizada para todas las aplicaciones que se encargan de mensajes SMS o MMS, ayudando a mejorar la experiencia del usuario cuando hay instaladas varias aplicaciones de mensajería, y a construir nuevas funciones de mensajería con soporte completo.
- **Modo inmersivo de pantalla completa:** Permite ocultar todos los controles del sistema para mostrar las aplicaciones a pantalla completa, muy útil para mostrar contenido enriquecido, pues permite utilizar cada pixel de la pantalla para mostrar el contenido y capturar eventos táctiles.
- **Framework Transiciones:** Facilita la implementación de animaciones de gran calidad, tanto con escenas mediante base predefinida o basada en propiedades, como animando los cambio sobre la marcha.
- **Vista web basada en Chromium:** Proporciona un amplio soporte para HTML5, CSS3 y JavaScript, siendo compatible con la mayoría de características de HTML5 de Chrome para Android 3.0 y un rendimiento mejorado de JavaScript con una versión actualizada del motor de JavaScript (V8).
- **Grabación de pantalla:** Soporta la grabación de vídeos de alta calidad en MP4 de la pantalla del dispositivo, pudiendo iniciar o detener la grabación en un dispositivo conectado al SDK de Android a través de un USB.
- **Bluetooth HOGP y MAP:** Estos dos nuevos perfiles de Bluetooth permiten el soporte de un mayor rango de interacciones de baja potencia. Bluetooth HID over GATT (HOGP) permite la interacción con periféricos como joysticks y teclados mientras que Bluetooth MAP posibilita la comunicación con dispositivos cercanos.
- **Emisores de infrarrojos (IR):** Permite a las aplicaciones controlar dispositivos electrónicos con receptores de infrarrojos como televisores o sintonizadores, comprobando anteriormente que nuestro dispositivo Android dispone de un emisor para comprobar frecuencias y enviar señales.
- **Mejoras en la seguridad:** Añade dos algoritmos criptográficos más, como el algoritmo de firma digital de curva elíptica (ECDSA), proveedor de almacén de claves de la mejora de la seguridad de la firma digital, aplicable a los escenarios, como la firma de una solicitud o de una conexión de datos. Para proteger las claves criptográficas utilizadas para el cifrado de disco completo implementa la función de derivación de claves Scrypt.
- **Herramientas para analizar el uso de memoria:** Es posible analizar el uso

de memoria de aplicaciones y servicios en ejecución con procstats, una nueva herramienta que ayuda a analizar los recursos utilizados, realizando un seguimiento de cómo las aplicaciones se están ejecutando a través del tiempo, proporcionando datos sobre sus duraciones de ejecución y el uso de memoria para ayudar a determinar la eficiencia con que se están realizando, incluso si se ejecutan en segundo plano. También es posible realizar estas tareas, tanto si es un dispositivo físico o virtual, con process stats, una herramienta para desarrolladores que se apoya en procstats pero nos muestra más información y comparar datos entre aplicaciones.

Sin embargo, aunque KitKat es la última versión de Android y ofrece diversas mejoras, en la Figura 6.2 hay versiones anteriores que siguen siendo muy utilizadas, por lo que tanto en la implementación como en el testeo se tiene en cuenta que la aplicación tiene que ser compatible con las últimas versiones de Android para poder ofrecer servicio a un mayor número de dispositivos.

Version	Codename	API	Distribution
2.2	Froyo	8	0.8%
2.3.3 - 2.3.7	Gingerbread	10	14.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	12.3%
4.1.x	Jelly Bean	16	29.0%
4.2.x		17	19.1%
4.3		18	10.3%
4.4	KitKat	19	13.6%



PARTE DEL TRABAJO CREADO Y COMPARTIDO POR ANDROID OPEN SOURCE PROJECT CC 2.5 ATTRIBUTION LICENSE.

Figura 6.2: Número relativo de dispositivos que ejecutan una determinada versión de la plataforma Android. Datos recogidos durante un período de 7 días que terminó el 4 de junio de 2014. No se muestra cualquiera de las versiones con distribución de menos de 0,1 %. Android Open Source Project [6, Dashboard].

### 6.1.2. Lenguajes de programación

#### Java

Java es un lenguaje orientado a objetos (P.O.O) cuya primera versión (JDK 1.0) se lanzó en 1996 y en la actualidad es uno de los lenguajes de programación más usados en todo el mundo, “con más de 9 millones de desarrolladores en todo el mundo” Oracle Corporation [35].

En muchos aspectos es similar a C y C++, pues son sus predecesores, sin embargo agrega ciertas características y elimina algunos elementos que provocaban muchos pro-

blemas debido al uso inadecuado de desarrolladores de aplicaciones, lo que ha ayudado a su gran expansión y popularidad. Así, aparte de ser usado para desarrollar aplicaciones cliente, cliente/servidor o web, también es un lenguaje muy popular para la programación de la lógica de aplicaciones Android, aunque no es el único es el más utilizado.

## XML

Extensible Markup Language (XML) es un Lenguaje de Etiquetado Extensible muy simple, muy flexible derivado de SGML (ISO 8879), pero estricto que juega es fundamental en el intercambio de datos muy variados, ya que su función principal es describir datos, estructurando, almacenando e intercambiando información, para facilitar su posterior lectura y uso en diferentes aplicaciones.

En Android, además de para tratar todo tipo de datos, es el formato designado para contener la información los elementos de cada actividad la aplicación así como su estilo de presentación a los que asigna los datos a mostrar a través de claves de identificación.

## 6.2. SCORM

SCORM (Sharable Content Object Reference Model) es un conjunto de estándares y especificaciones de empaquetamiento muy utilizado que permite presentar material didáctico estructurado de forma adecuada mientras que asegura la importación de contenidos en diferentes sistemas de gestión de aprendizaje es posible si éstos cumplen el estándar.

Esto permite empaquetar y reutilizar contenidos de aprendizaje, de modo que un mismo paquete de contenidos se puede importar desde diferentes LMS o plataformas de e-learning, facilitando así la interoperabilidad entre las plataformas.

### 6.2.1. Contenidos y Empaquetado

Un paquete con estructura SCORM es un archivo en formato PKZip v2.04g (.zip), que en el entorno SCORM es llamado PIF (Package Interchange File) que utiliza de forma estricta la especificación IMS Content Packaging Specification. Gracias a esto es posible el intercambio entre diferentes LMS.

El PIF es un Paquete de Contenidos formado por el manifiesto, un documento en

XML que describe la estructura de contenido y recursos, y los ficheros que conforman el contenido real.

## Archivos SCO y assets

El modelo de contenido SCORM muestra cómo a partir de recursos educativos, se puede construir un curso de formación.

Los elementos que constituyen esta entidad deformación se definen por:

- **Asset** Un asset es un recurso básico de la formación visualizable en navegadores Web que representan textos, imágenes, sonidos, páginas HTML, objetos de evaluación, y otros tipos de datos.

Para el LMS, es un recurso para cargar en el navegador sin la API de SCORM pues el asset no se comunica con el LMS. Un asset puede ser una simple colección de archivos que presentan dependencias con recursos en un manifiesto SCORM o una parte de la formación similar a un SCO que puede ser utilizado con las reglas de secuencia o asociado a unos metadatos.

- **SCO (Sharable Content Object)** Un SCO es un objeto de aprendizaje formado por el conjunto de uno o más recursos (asset) y de código JavaScript que le permite comunicarse con un LMS a través de un entorno de ejecución en tiempo real RTE. Un SCO es la unidad de información más pequeña que puede mostrarse a un alumno a través de un LMS y puede estar asociado a unos metadatos.

Estos fragmentos de formación que se ensamblan en organizaciones dentro de un manifiesto son reutilizables, dentro y a través de los cursos, ya que el contenido se define como un conjunto de SCOs.

Para el LMS, se trata de un recurso a cargar pero con la API de SCORM. Un SCO debe buscar obligatoriamente la API (API-1484-11) colocada en el DOM por el LMS, para inicializar la comunicación llame al método Initialize() sin olvidar llamar al método Terminate() al salir del recurso.

- **Actividad** La actividad es un elemento educativo utilizado por el alumno para pedir a los LMS la puesta en marcha un asset o un SCO. Para un estudiante, un curso de formación se compone sólo de actividades.

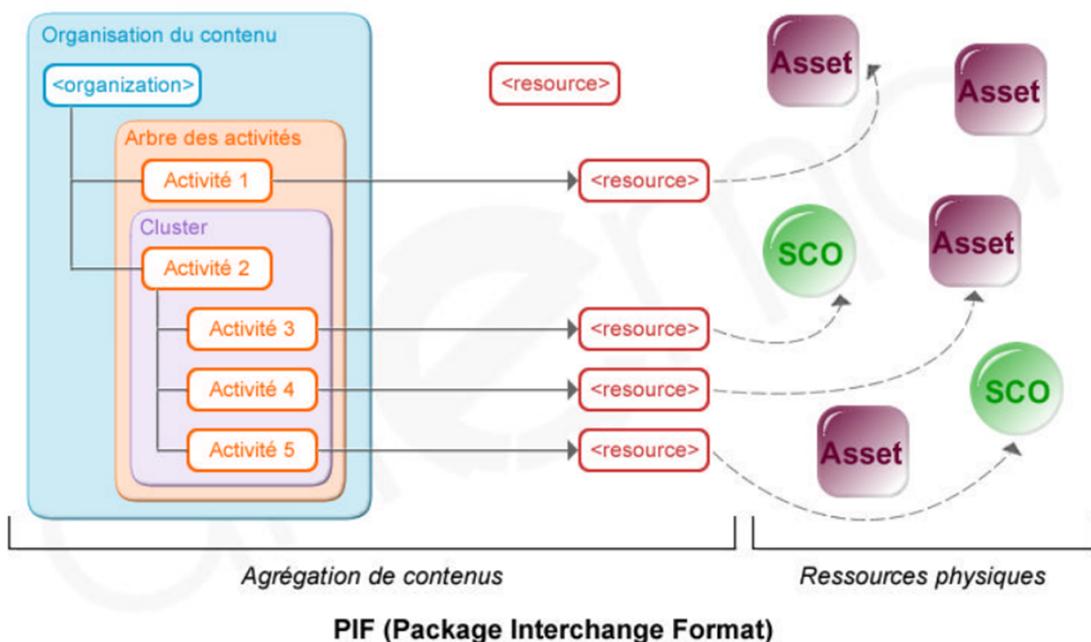
Para el LMS, son los elementos que constituirán el menú de la unidad educativa.

Una agrupación de actividades se llama clúster o agregación mientras que el conjunto de las actividades se llama el árbol de Actividades. SCORM no limita

la profundidad de esta estructura

## Manifiesto

El archivo imsmanifest.xml es un documento XML o Extensible Markup Language que debe respetar las especificaciones XML de W3C y siempre se encuentra en la raíz del paquete de contenido. El manifiesto se declara con la etiqueta *<manifest>*, contiene la información que describe el contenido que un LMS necesita para organizar y mostrar los contenidos de forma adecuada y con que reglas de secuenciación.



ANEMA SEG04 CC 2.5 BY-NC-SA.

Figura 6.3: Estructura descrita en un manifiesto SCORM de un paquete PIF. Anema [13, pág. 9]

Una vez visto el esquema de la estructura básica del formato SCORM descrita en un manifiesto (ver Figura 6.3), pasamos a analizar los elementos que conforman el manifiesto.

- **Metadatos** Los metadatos se encuentran en un formato bien definido y recomendado por ADL llamado IEEE 1484.12 Learning Object Metadata o metadatos de objetos de aprendizaje aunque es más conocido como LOM. Este estándar contiene etiquetas y atributos predefinidos para describir todos los elementos que conforman el contenido de aprendizaje, de esta forma los LMS pueden buscar en estos datos información sobre las unidades didácticas.

Los metadatos pueden incluirse directamente en el manifiesto o referenciar a un archivo del PIF que los contenga siempre indicando la conformidad con el esquema

SCORM de SDL <*schema*> y con el modelo de agregación de contenido (CAM) <*schemaversion*>.

Los metadatos se pueden aplicar a prácticamente cualquier sección del manifiesto, por ejemplo se puede aplicar para el curso como un todo, a los elementos individuales, o incluso a los recursos y los archivos individuales para mejorar su reutilización.

- **Organización** Describe la organización de los contenidos en forma de un árbol de actividades definiendo la organización por defecto en la etiqueta <*organizations*>.

Dentro de este elemento se encuentra <*organization*> que indica el elemento raíz del árbol de actividades y su nombre en la etiqueta <*title*>. Además debe encontrarse alguna etiqueta <*item*> que define una actividad del árbol.

Cada actividad está definida por un identificador, además, si es una actividad final debe presentar la referencia al identificador del recurso que necesite y si es necesario parámetros para el recurso cuando sea ejecutado como una query URL. Como hijo puede tener otro <*item*> que debe contener un título, opcionalmente puede contener metadatos, límite temporal <*adlcp : timeLimitAction*>, inicialización de datos para los SCO <*adlcp : dataFromLMS*>, un umbral para SCO <*adlcp : completionThreshold*>.

También puede incorporar las instrucciones de secuenciación de los contenidos y la presentación de los mismos en la navegación, incluidas a la definición de las actividades (item) <*imsss : sequencing*> <*adlnav : presentation*> o en forma de una colección de reglas y estrategias en el manifiesto

- **Recursos** En el elemento <*resources*> se describen los recursos físicos utilizados por el paquete, los externos a través de URL, y locales que están comprimidos en el paquete PIF. En nuestro caso queremos que se pueda acceder a todos los contenidos de forma offline, por lo que no será posible visualizar todos los recursos externos si no estamos conectados a la red.

Todos los objetos de contenido que constituyen el PIF deben estar declarados como <*resource*> definidos con su identificador(identifier) y su tipo (type y adlcp:scormType) junto a las dependencias entre ficheros <*dependency*> y el fichero <*file*> al que identifica, definiendo así el objeto de aprendizaje y como debe ser tratado por el entorno de ejecución RTE.

Si el recurso no necesita comunicarse con la plataforma es un asset, si se trata de un SCO debe contener código de ejecución para poder comunicarse con la plataforma, en este caso un fichero con código JavaScript llamado apiWrapper.js

que va en el PIF con el resto de documentos. Todo recurso que pueda ser lanzado por el alumno contiene un puntero a la página a la que el LMS debe redireccionar al alumno con el fin de poner en marcha el recurso.

### 6.2.2. Run-Time Environment RTE

La especificación de SCORM en tiempo de ejecución RTE controla como el LMS ejecuta el contenido y como éste se comunica con el LMS dentro del contexto de un único SCO.

El LMS pondrá en marcha un SCO a la vez, según lo seleccionado por el usuario, o según lo determinado por reglas de secuenciación definidas en el manifiesto, asimismo, debe contener algún tipo de tabla de contenidos navegable, así como los controles de navegación anterior y siguiente para controlar la navegación entre los SCOs. Si es necesario, el SCO debe proporcionar sus propios elementos de navegación interna.

Todo el contenido SCORM debe ser entregable en la web y toda la comunicación de SCORM se produce en el contexto de una sesión de navegador de web. Comúnmente, si un curso sólo contiene un SCO (no requiere elementos de navegación), el SCO se lanzará en una ventana emergente; si contiene muchos SCOs, el LMS lanzará la los SCOs en un conjunto de frames, rodeado de elementos de navegación.

Toda comunicación entre un SCO y el LMS debe ser iniciada por un SCO y pasa a través de una API ECMAScript (JavaScript) facilitada por el LMS, se limita a responder a las llamadas realizadas por el contenido, es la única forma, no se puede comunicar a través de servicios web, formularios, bases de datos o cualquier otro mecanismo.

### 6.2.3. Secuenciación

La secuenciación es la parte más compleja de SCORM. Es el proceso típicamente iniciado por una solicitud de navegación en el que los objetos de contenido son seleccionados por el LMS para su entrega al alumno. Durante el proceso de secuenciación, las reglas se evalúan y una actividad (SCO o agregación) se identifica para la entrega especificando cómo el alumno puede navegar entre las partes del curso (SCO). Se define por un conjunto de reglas y atributos escritos en XML.

La información de secuenciación, al formar parte del manifiesto del PIF y al estar relacionada con los elementos del árbol de contenidos, pueden incluirse los comportamientos previstos en el paquete de forma que pueda utilizarse en cualquier plataforma de e-learning conforme al estándar SCORM ofreciendo la misma experiencia de apren-

dizaje ya que cada elemento representa una actividad que el alumno puede realizar bajo el control de la información de secuenciación.

Cambiar de un SCO al siguiente, reintentar un clúster si el anterior ha sido fallido, salir de un clúster a mitad u omitir una actividad que ya se ha intentado son algunos ejemplos de secuenciación.

### 6.3. Hardware, Software y API's

- **ADT Bundle [9]** Incluye todo lo necesario para empezar a desarrollar aplicaciones, incluyendo APIs , librerías y herramientas de desarrollo para construir, probar y depurar aplicaciones: Eclipse + plugin ADT(Android Developer Tools) Android SDK Tools Android Platform-tools La plataforma de Android más reciente La apariencia de sistema Android más actual en el emulador

Esto incluye la creación de máquinas virtuales para emular diversos dispositivos Android. Para esta aplicación nos centraremos en la tablet Nexus 7 con sistema Android 4.4 KitKat aunque incluyendo herramientas de soporte para versiones anteriores.

- **Dropbox Platform [23]** La plataforma Dropbox nos proporciona API's y servicios para integrar las características de Dropbox en nuestras aplicaciones, ya sean móviles o web, permitiendo a los desarrolladores que el acceso a los archivos sea tanto descargando ficheros desde nuestra cuenta web como creando carpetas sincronizadas de forma similar a la aplicación oficial. Esto nos servirá para permitir la visualización de SCORMs desde una cuenta de Dropbox en la aplicación.

Desde junio de 2014 está disponible la versión 3.0 de Sync API y Datastore API.

- **eXe Learning [14]** Una herramienta de código abierto que permite la creación de objetos de aprendizaje en formato estándar (SCORM) que posibilita realizar pruebas con diferentes archivos para la corrección de bugs y comprobar la compatibilidad entre plataformas y versiones.

El proyecto eXe Learning fue financiado por el Gobierno de Nueva Zelanda y coordinado por la University of Auckland, The Auckland University of Technology y Tairawhiti Polytechnic.

## 6.4. Diseño de la aplicación. Interfaz

La interfaz se diseña teniendo en cuenta el acelerómetro que incluyen los dispositivos móviles y así presentar los contenidos de forma adecuada cambiando la disposición de los elementos según el tamaño de la pantalla y de su orientación respecto al usuario.

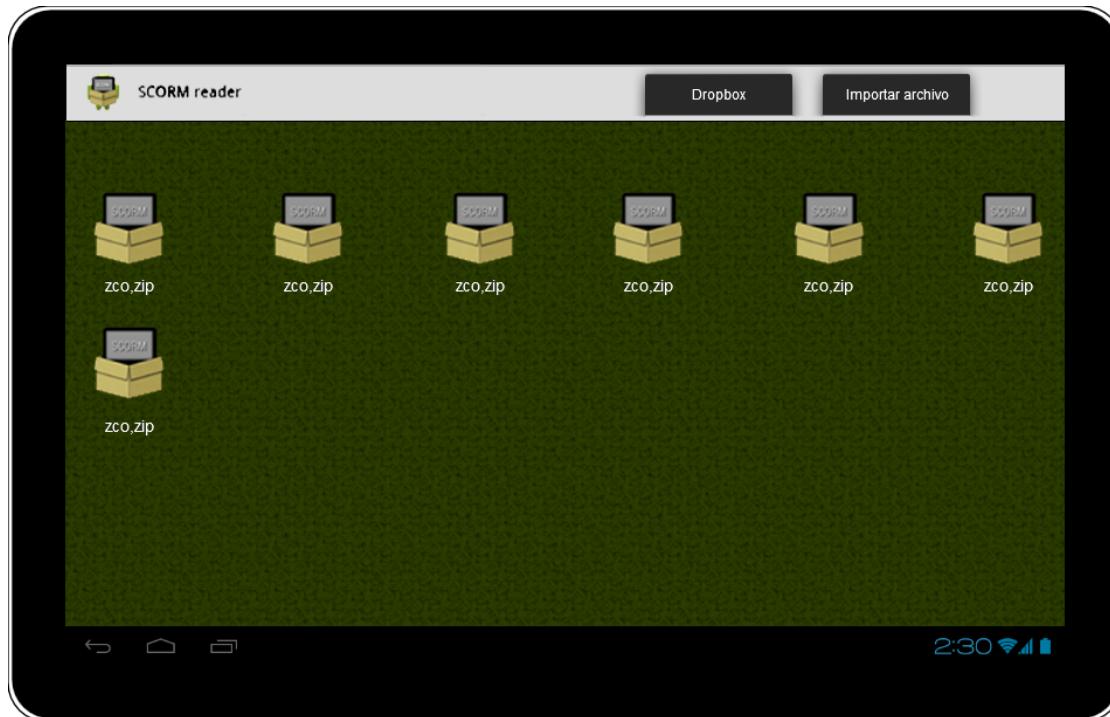


IMAGEN DE ELABORACIÓN PROPIA

Figura 6.4: Visualización de la pantalla de inicio

La aplicación se centra en dispositivos tablet por poseer pantallas más adecuadas a la hora de mostrar e interaccionar con los contenidos, por ello el recurso SCORM se mostrará de una forma u otra conforme se detalla a continuación.

Se muestra un escritorio, figura 6.4, en el que vemos los iconos de los recursos en formato SCORM que hayamos importado mediante el botón importar, además de un botón que permite seleccionar un archivo SCORM desde nuestra cuenta de Dropbox.

Como vemos en la figura 6.5, página 32 , al abrir un curso, tendremos a la izquierda de la pantalla un menú de las lecciones del curso y los contenidos a la derecha si la tablet está apaisada como en la figura 6.5c.

Si por el contrario está vertical, primero nos muestra el menú de contenidos en el que seleccionaremos una lección tal como vemos en la figura 6.5a, la lección elegida ocupa toda la pantalla como vemos en la figura 6.5b.

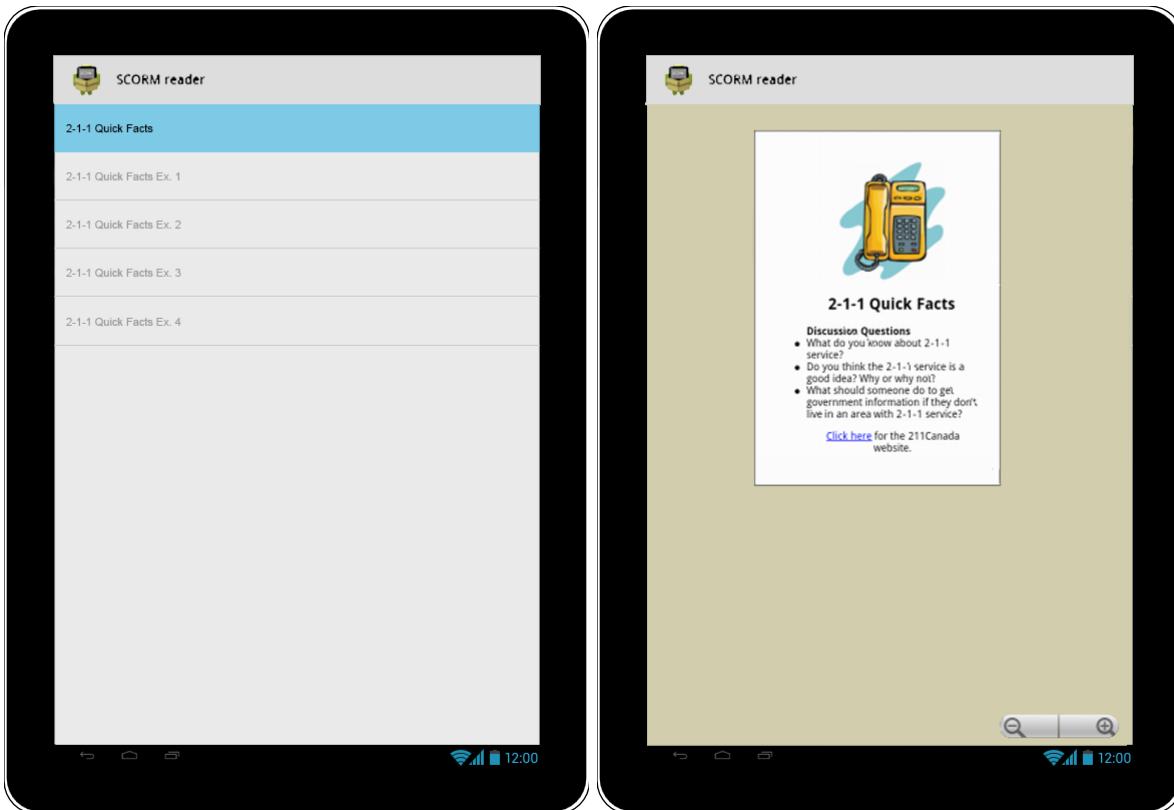


IMAGEN DE ELABORACIÓN PROPIA

IMAGEN DE ELABORACIÓN PROPIA

(a) Menú en vertical

(b) Contenidos en vertical

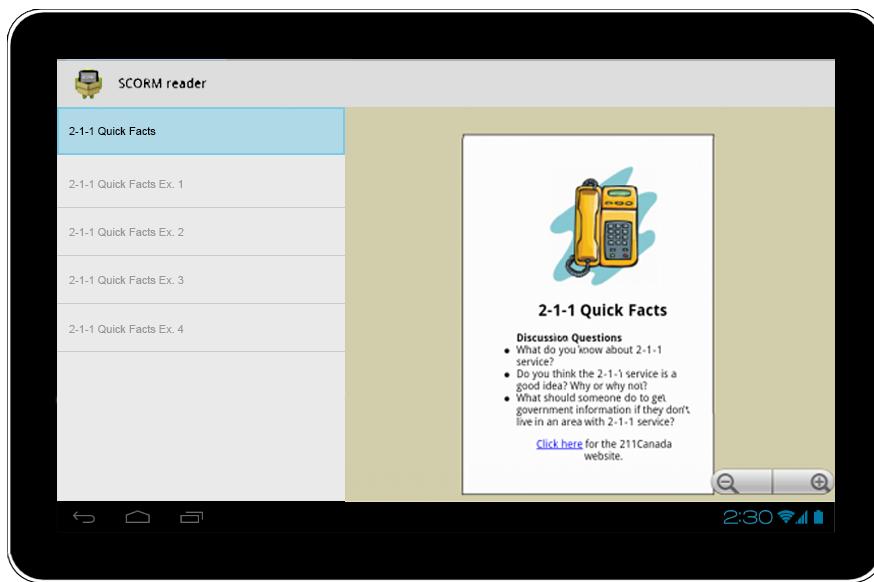


IMAGEN DE ELABORACIÓN PROPIA

(c) Ambos elementos en apaisado

Figura 6.5: Visualización de un recurso SCORM según la orientación del dispositivo

Tanto para el ícono de la aplicación como para los paquetes tipo SCORM se ha creado

un ícono que simboliza la apertura y lectura del paquete PIF y la presentación de los contenidos del recurso en la tablet.

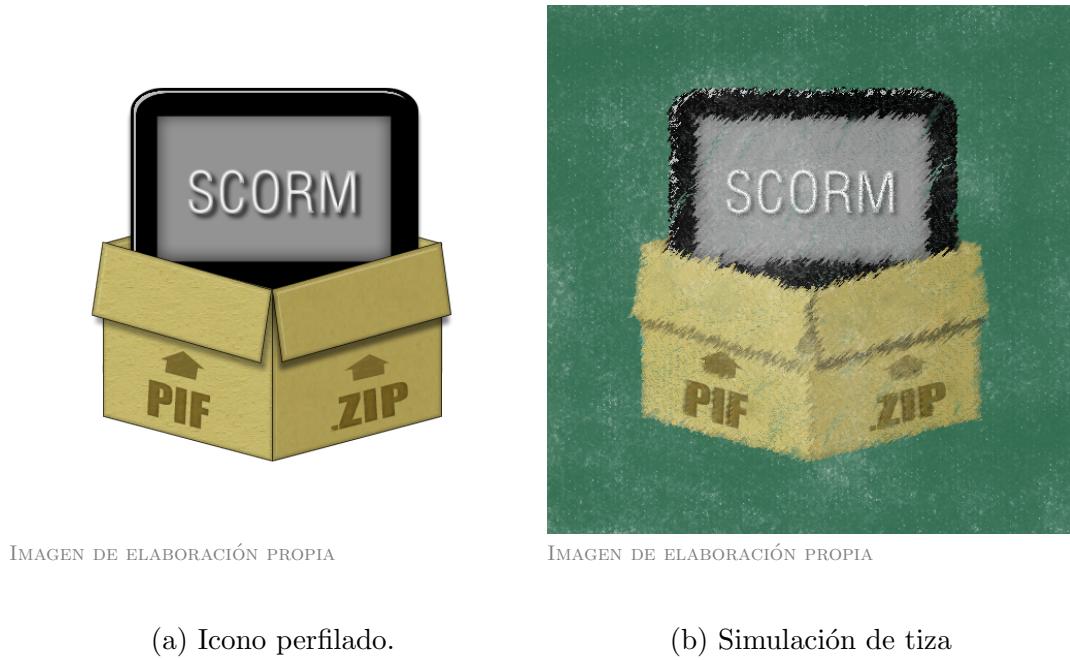


Figura 6.6: Icono de la aplicación

La iconografía de la aplicación se encuentra en la figura 6.6 vemos una tablet, con la palabra SCORM en la pantalla, que emerge de una caja donde se encuentran los contenidos. La figura 6.6a muestra unos trazados limpios y precisos, sin embargo, para simular mejor una pizarra se ha retocado la imagen hasta llegar a la figura 6.6b. La



Figura 6.7: Fondo de la aplicación.

imagen de fondo, que simula el tablero de la pizarra, figura 6.7 también se ha creado mediante diferentes filtros y pinceles, acompañados del color de los textos que son de un tono tipo blanco roto para dar más realismo.

## 6.5. Implementación

Iniciamos la fase de implementación con el estudio de la lógica y arquitectura de la aplicación resumida en la figura 6.8 de la página 34.

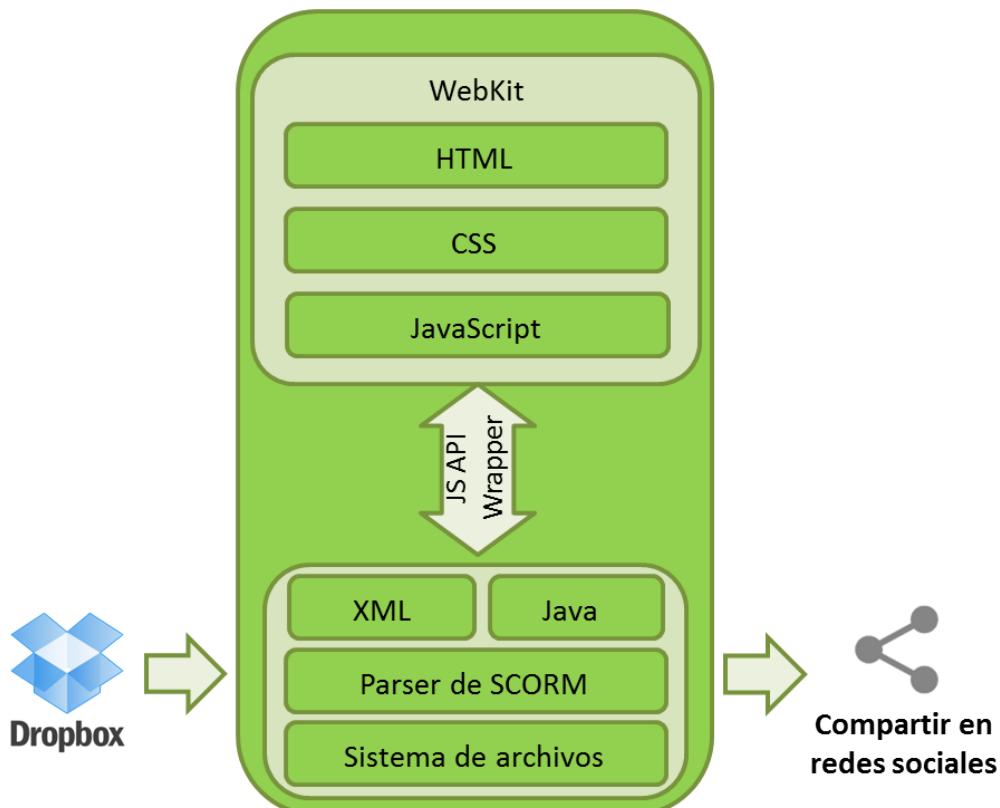


IMAGEN DE ELABORACIÓN PROPIA. LOGOTIPO PERTENECIENTE A DROPBOX.

Figura 6.8: Arquitectura de la aplicación.

Como vemos, la aplicación Android localiza un recurso SCORM, desde memoria del dispositivo o desde Dropbox, maneja los datos y los muestra en una vista web o web-View, los recursos del paquete en formato SCORM que se ejecutan en la vista manejan los datos introducidos por el usuario gracias al APIwrapper y lanzan llamadas al adaptador API correspondiente para comunicar a un LMS estados del archivo SCO y datos relacionados con él.

Esta aplicación no debe depender de ningún LMS específico, por lo que los adaptadores API implementados soportan las llamadas al LMS para evitar errores de funcionamiento del recurso. Además el lector SCORM permite su publicación en redes sociales

ofreciendo al usuario la oportunidad de justificar sus progresos.

Para consultar la API de Android y ejemplos de uso de algunas funciones se han consultado varias webs especializadas en Android [1, 7, 42]

### 6.5.1. Manifiesto de la aplicación Android

La aplicación necesita que el usuario apruebe varios permisos para que la aplicación se instale en el dispositivo, de lo contrario no podría funcionar correctamente.

- “android.permission.INTERNET”

Permite el acceso íntegro a Internet desde la aplicación, esto es necesario para la conexión a Dropbox y la visualización de materiales de los recursos que no estén adaptados a visionado offline y parte de sus recursos sean de acceso web.

- “android.permission.ACCESS\_NETWORK\_STATE”

Permite conocer el estado de la conexión a Internet en el dispositivo, esto es útil para comprobar si es o no posible lanzar tareas que requieran de conexión a Internet y avisar al usuario en su defecto.

- “sandroid.permission.WRITE\_EXTERNAL\_STORAGE”

Permite el acceso, lectura y escritura en la tarjeta SD del dispositivo si está presente. La aplicación utiliza estas características para almacenar y utilizar los archivos zip que contienen los recursos. También para acceder a los archivos descargados en el dispositivo e importarlos.

### 6.5.2. Actividades que forman la aplicación

#### MainActivity

Actividad principal de la aplicación, muestra la lista de recursos importados y permite acceder a Dropbox para descargar archivos, importar archivos descargados, abrir uno de los recursos o borrarlos.

- **Visualización de un archivo HTML local en un webView lanzado desde un intent [1]**

Programamos dos actividades, la principal que nos pide el nombre de un archivo visualizable en un entorno web. Una vez seleccionado se envía esta información

a la segunda actividad mediante un intent.

Creada la segunda actividad, busca el archivo especificado recuperando su nombre del intent, abriendo una vista web webView y muestra el contenido. Si no se encuentra el archivo la webView nos muestra un aviso.

- **Eliminar archivos de la aplicación**

Una buena manipulación de archivos debe permitir borrar archivos de la aplicación. Si en la pantalla principal mantenemos pulsado la aplicación nos pregunta si deseamos borrar el archivo. Si aceptamos eliminamos el archivo de la aplicación

## Actividad2

Actividad que incluye dos fragmentos para mejorar la experiencia de usuario, muestra el listado de contenidos

- **Descompresión de un archivo zip de un SCORM y lanzamiento de index.html**

Mediante la selección en la pantalla de inicio del nombre del curso a visualizar, éste se busca en la tarjeta SD del dispositivo, extrayendo el contenido del zip correspondiente en la carpeta ”unzipped” donde se encontrarán todos los archivos del curso para poder acceder a ellos.

En el caso que no quede espacio suficiente para la descompresión o se produzca un error muestra al usuario un dialogo informativo. Antes de descomprimir el recurso en la carpeta ”unzipped” se comprueba que esté vacía con el fin de optimizar espacio.

En primer lugar se accede al manifiesto, el cual tendremos que analizar para poder reconstruir el curso en formato SCORM en el entorno de nuestra aplicación para visualizarlo en un webView Android.

- **Parser de feeds XML de Android [11] y su adaptación a imsmanifest.xml**

Se ha estudiado el funcionamiento del XMLParser de Android mediante un ejemplo para un feed RSS de Android developers [7].

Para las etiquetas organizations y resources se crean varias funciones que las recorren en profundidad para obtener sus datos y los de las etiquetas hijo diferenciando y relacionando los elementos que las componen. De esta forma se repro-

duce la estructura del recurso detallada en el manifiesto asignando los recursos correspondientes mediante sus respectivos identificadores.

- **Visualización del curso mediante fragments [8]**

Esto permite visualizar de diferente forma el curso dependiendo tanto del tamaño de la pantalla como de la orientación del dispositivo. Para permitir la correcta visualización en dispositivos con versión de Android anterior a la de la aplicación incluimos en el proyecto los fragments mediante las librerías de soporte que ofrece Android.

En este caso incluiremos en la carpeta libs del directorio de la aplicación la v4 Support Library que está pensada para su utilización con la versión 1.6 de Android (API 4) y posteriores. Sin embargo los fragments se introducen en la versión 3 de Android o Honeycomb.

- **Listado archivos de la carpeta de la aplicación y mostrarlo en un gridView [10, 25]**

Una vez iniciada la aplicación, muestra la pantalla principal que nos permite seleccionar uno de los recursos educativos importados a la aplicación. Éstos están representados con un ícono, acompañado del nombre del recurso de extensión zip, dentro de una gridView que hará las funciones de rejilla para la correcta colocación de los iconos en pantalla.

En cuanto pulsemos en algún elemento de la rejilla comienza el proceso de descompresión, lectura de manifiesto para estructurar el recurso y lanzarlo en la webview.

- **Importar archivos descargados a la aplicación**

En las primeras pruebas de la aplicación se han importado los archivos en formato SCORM desde la perspectiva DDMS (Dalvik Debug Monitor Server) de ADT Bundle, sin embargo en la aplicación final debe permitir la importación de archivos desde la propia aplicación.

Para ello se incluye un botón descargas que al pulsarse abre una ventana modal con un listado filtrado que nos muestra únicamente los archivos en formato .zip, extensión que utilizan los recursos SCORM. Si no hay archivos de este tipo nos muestra un aviso, si no, al seleccionar uno de ellos el archivo se importa a la carpeta de la aplicación y aparece en la pantalla principal para poder seleccionarlo.

- **Conectar con SCORM API Wrapper[32, 39, 44]**

En el formato SCORM, todo recurso de tipo sco intenta comunicarse con un LMS a través de API Wrapper, sin embargo nuestra aplicación no va a depender de ningún LMS, así que debe comunicarse con el recurso mediante JavaScript para conseguir un funcionamiento correcto.

Es decir se implementa un adaptador LMS en JavaScript para evitar errores en el funcionamiento del recurso y la aparición de popups informativos relacionados con estos errores de comunicación.

Para aumentar la compatibilidad del adaptador JavaScript contemplamos dos casos según la versión de SCORM, el ApiWrapper busca un objeto que representa al adaptador, en el primer caso el objeto se denomina “API”, en el segundo “API\_1484\_1”. Ambos objetos deben contener las 8 funciones con las que el SCO se comunicaría con el LMS:

1. LMSInitialize(); / Initialize();
2. LMSSetValue(); / SetValue();
3. LMSGetValue(); / GetValue();
4. LMSCommit(); / Commit();
5. LMSFinish(); / Finish();
6. LMSGetString(); / GetErrorString();
7. LMSGetLastError(); / GetLastError();
8. LMSGetDiagnostic(); / GetDiagnostic();

Las funciones con el prefijo LMS pertenecen a versiones anteriores a SCORM 2004. El algoritmo de descubrimiento del adaptador API presente en ApiWrapper.js comienza a buscar el objeto en la ventana padre de forma recursiva, por lo tanto el elemento en el que se instancian las variables objeto de las librerías adaptador LMS debe estar relacionado con la ventana en la que se ejecuta el SCO, de otro modo este no será capaz de encontrarlo.

En este caso, tras estudiar las posibilidades de webView en Android, mediante inyección de objetos, código JavaScript y HTML desde el código Java a la WebView de Android se coloca cada uno de los objetos adaptador API en un frame, mientras que la página se crea en otro.

De esta forma ambos adaptadores son accesibles desde JavaScript por el ApiWrapper, así como desde la aplicación Java se pueden seguir haciendo llamadas

JavaScript a través de las funciones  *loadData()* y  *loadDataFromBsaeURL()* o mantenemos la conexión entre las clases Java “API” y “API\_1484\_11” con ApiWrapper gracias a  *addJavascriptInterface(ObjectJavaObj, StringID)*

Para marcar el progreso del usuario en el entorno de la aplicación, sería posible recoger algunos datos de progreso de dicho recurso a través del adaptador JavaScript y guardarlos de tal forma que los relacione con cada uno de los recursos importados y así poder mostrarlos en la siguiente visita del usuario. Normalmente estos datos se guardan en la base de datos del LMS. En esta aplicación el seguimiento del progreso se realiza mediante publicaciones sociales.

#### ■ Publicar progreso

Como la aplicación no depende de ningún LMS, los datos de progreso no pueden comunicarse a través de ellos. La aplicación permite esta comunicación de información gracias a su conexión con las redes sociales que tengamos instaladas en nuestro dispositivo.

A través del botón compartir que encontramos en la pantalla del curso se nos muestra una ventana con las posibles aplicaciones por las que podemos compartir como pueden ser las diferentes redes sociales (ej. Twitter) así como correo electrónico o mensajería móvil.

La función compartir está programada para consultar la información recopilada en la lectura del manifiesto y obtener los datos título del recurso y unidad en la que nos encontramos y generar un texto genérico como sugerencia para publicar el progreso.

## DropboxFiles

#### ■ Importar archivos desde Dropbox [23, 24, 43]

Se proporciona la posibilidad de importar archivos desde una cuenta de Dropbox facilitando el aprendizaje ubicuo, pues de esta forma el usuario puede acceder a más recursos allá donde los necesite.

Esta utilidad se implementa introduciendo en las librerías del proyecto Dropbox Core API para Android versión 1.6.1 y JSON simple 1.1 disponibles en la página de desarrolladores de Dropbox, dónde hay que registrar la aplicación en la consola para obtener claves de acceso que junto a al nombre y contraseña permiten autenticar la conexión y acceder a todos los archivos que el usuario tiene en Dropbox y trabajar con ellos. Para comprobar el funcionamiento de la SDK de

Dropbox para Android se estudia **DBRoulette**, la aplicación que proporciona Dropbox en su SDK como ejemplo.

Al pulsar el botón Dropbox la aplicación comprueba si la conexión a Internet está disponible para lanzar las llamadas a Dropbox, si no muestra un aviso informando de que no la conexión a Internet no está activada.

- **Autenticación**

Para acceder a los archivos de una cuenta Dropbox es necesario que el usuario autorice a la aplicación acceder a los archivos a través de la autenticación OAuth2.

Una vez se ha autorizado que el lector SCORM conecte a la cuenta de usuario Dropbox devuelve unos tokens de acceso que es necesario guardar para iniciar la conexión con ellos sin que el usuario se vea obligado a otorgar el permiso de acceso cada vez que la aplicación necesite conectar con la cuenta.

En este caso este tipo de datos se guardan en las preferencias Android de nuestro dispositivo para asegurar su persistencia. Sin embargo, estos token pueden ser revocados cada cierto tiempo, por lo que el permiso tendrá que ser renovado para seguir accediendo a nuestra cuenta.

- **Acceso y manipulación de archivos**

SCORM reader permite navegar por todas las carpetas de una cuenta de Dropbox, además de los diferentes directorios, al estar dirigida a recursos en formato SCORM, la aplicación únicamente muestra aquellos archivos con extensión .zip

Para descargar un archivo no tendremos más que seleccionarlo en la lista, este se importa automáticamente a la carpeta de recursos SCORM de la aplicación.

Debido a la configuración de Android para seguridad y buen funcionamiento del dispositivo no permite ejecutar operaciones potencialmente costosas en el hilo principal o de interfaz de usuario (UI), por ello, las tareas de acceso a carpetas de Dropbox o descarga de archivos se realizan en nuevos hilos (threads) o tareas asíncronas, de esta forma nos aseguramos de que el funcionamiento principal del dispositivo no se ralentiza y no disminuye la experiencia de usuario.

## AuthActivity

Esta actividad es propia de la SDK de Dropbox, por lo tanto no es una actividad como las anteriores, AuthActivity contiene las claves que la aplicación tiene asignadas en la consola de Dropbox para poder finalizar el proceso de autenticación en el proceso de conexión a la cuenta de Dropbox.

## 6.6. Pruebas/testing

Las pruebas se realizan en las máquinas virtuales Android con las características que observamos en la figura 6.9.

AVD Name	Target Name	Platform	API Level	CPU/ABI
And3Nexus_7	Android 3.0	3.0	11	ARM (armeabi)
And4_4_Nexus...	Android 4.4.2	4.4.2	19	ARM (armeabi-v7a)

IMAGEN DE ELABORACIÓN PROPIA

Figura 6.9: Listado de máquinas virtuales.

Para las distintas versiones de Android tenemos que tener en cuenta las compatibilidades y cada visualización con el dispositivo tanto en vertical como apaisado, los test se han realizado en las siguientes interfaces:

- La primera máquina, configurada como un tablet Nexus 7 con sistema operativo Android 4.4.2 (Kit Kat), figura 6.11, la última versión por el momento,

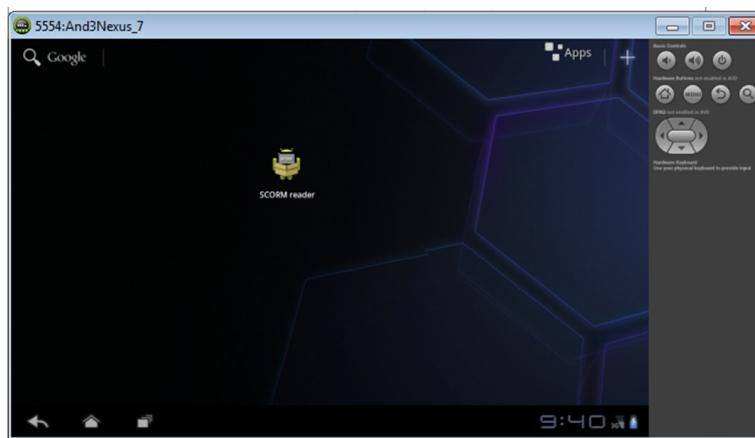


IMAGEN DE ELABORACIÓN PROPIA

Figura 6.10: VM Android 3.0

- y la segunda máquina virtual, también Nexus 7, pero con versión de Android 3.0 (Honeycomb), visible en la figura 6.10, por ser la primera versión con una actualización específica para este tipo de dispositivos [6, Honeycomb].

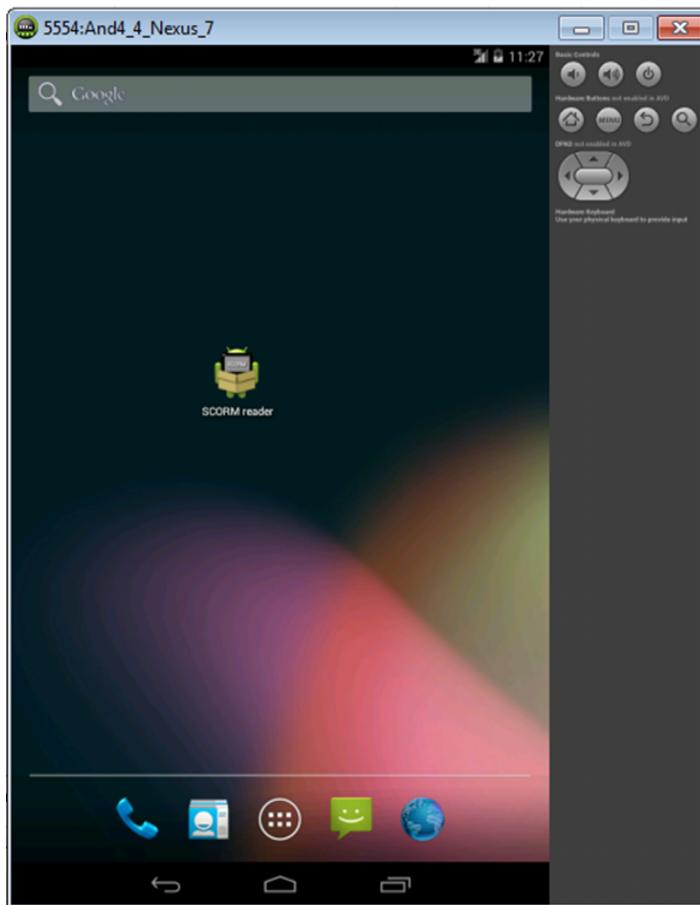


IMAGEN DE ELABORACIÓN PROPIA

Figura 6.11: VM Android 4.4

Trabajar en una máquina virtual de Android en eclipse nos permite acceder al sistema de archivos de la tablet virtual mediante DDMS, realizar un seguimiento de los eventos que ocurren en la máquina virtual mediante consola o ejecutar la aplicación en modo debug.

Esto nos permite realizar las pruebas necesarias a la aplicación ofreciéndole archivos que necesite, observando que no haya errores y en su caso ver que excepciones han tenido lugar e incluso en qué línea. También es posible consultar los valores de las variables paso por paso proporcionando un medio para detectar fallos de funcionamiento más rápido y con mayor precisión.

Para comprobar el funcionamiento de la aplicación y controlar la gestión de excepciones se realizan pruebas con distintas condiciones como:

- Presencia o no de carpetas y archivos a utilizar,

- archivos zip de formato incorrecto,
- correcta lectura del manifiesto,
- posibles excepciones en la apertura, lectura y manejo de archivos,
- funcionamiento con y sin conexión,
- dispositivos distintos,
- diferentes versiones de Android,
- orientación vertical y apaisada de la pantalla,
- conexión con servicios externos
- o manejo de datos de la aplicación.

Estas pruebas también se han realizado en un dispositivo físico, una tablet Android 4.0 Ice Cream Sandwich con pantalla de 8.0 pulgadas instalando la aplicación a través del archivo .apk como podemos observar en las figuras 6.12 y 6.13



IMAGEN DE ELABORACIÓN PROPIA

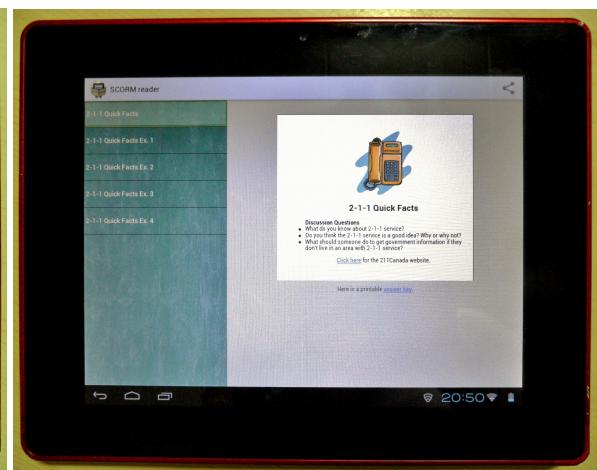


IMAGEN DE ELABORACIÓN PROPIA

(a) Inicio apaisado.

(b) Curso apaisado

Figura 6.12: Pruebas de un recurso SCORM con dispositivo en apaisado

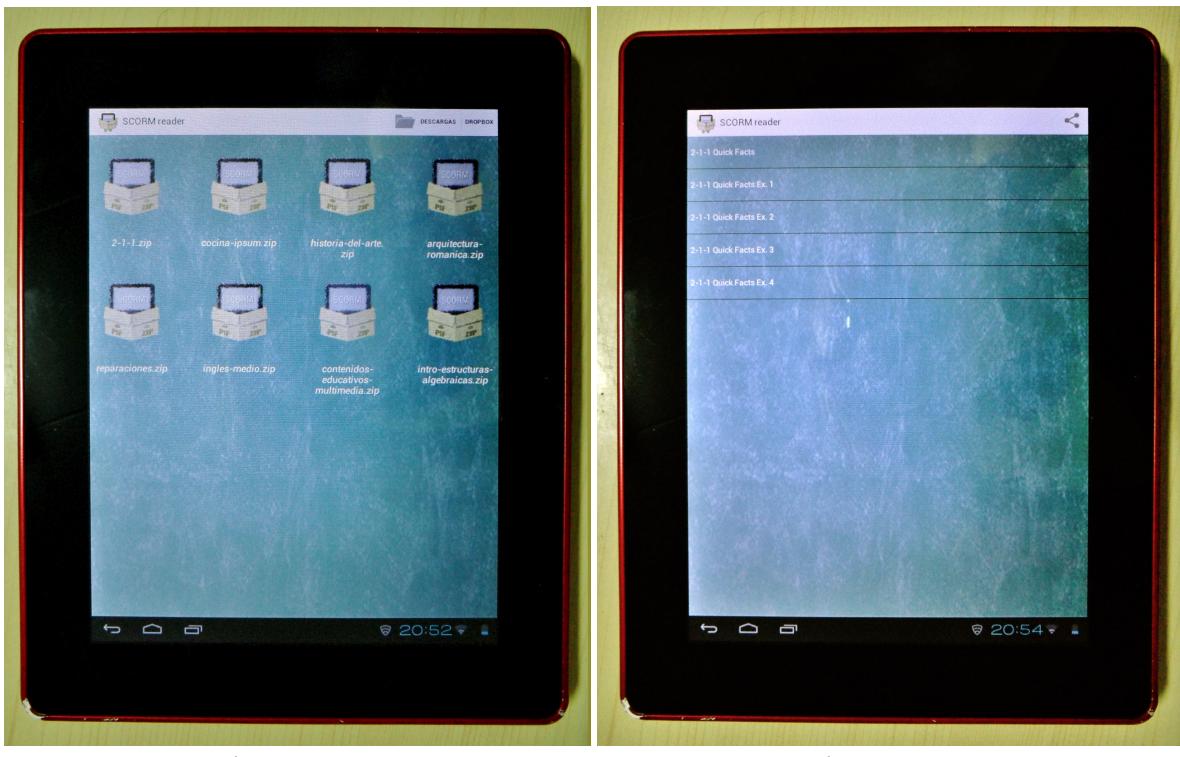


IMAGEN DE ELABORACIÓN PROPIA

IMAGEN DE ELABORACIÓN PROPIA

(a) Menú en vertical

(b) Listado de unidades

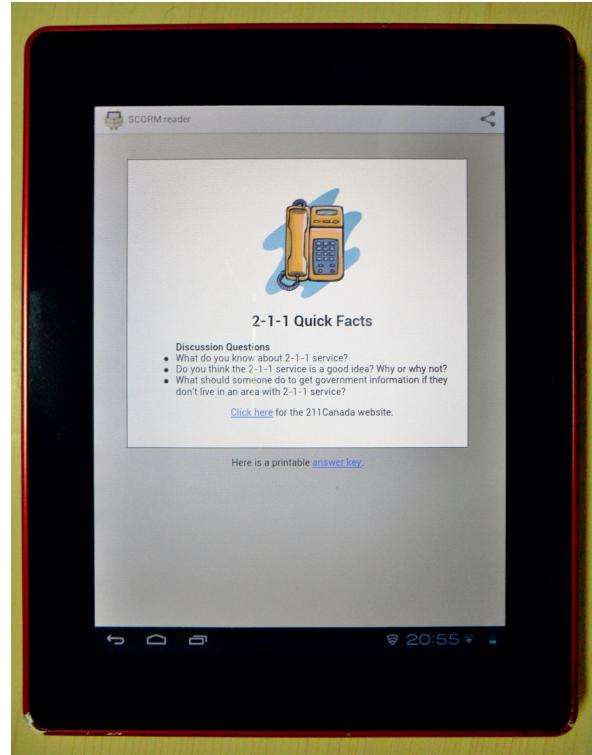


IMAGEN DE ELABORACIÓN PROPIA

(c) Contenidos en vertical

Figura 6.13: Pruebas de un recurso SCORM con dispositivo en vertical

## 6.7. Difusión y comercialización

### 6.7.1. Obtención del archivo .apk firmado

A fin de publicar la aplicación en una plataforma de comercialización como la de Google o entregarla al usuario final es necesario firmar la aplicación en modo release [12] pues Android precisa que las aplicaciones tengan una firma digital a partir de un certificado que identifica al autor de la aplicación para su distribución.

Trabajando en el entorno de desarrollo Eclipse y ADT, el primer paso es seleccionar la exportación de una aplicación Android y allí especificar el proyecto a exportar Figura 6.14a. Una vez seleccionado nos pide seleccionar un keystore existente, Android recomienda usar un único keystore para todas nuestras aplicaciones, o crear una nueva, opción que seleccionamos Figura 6.14b.

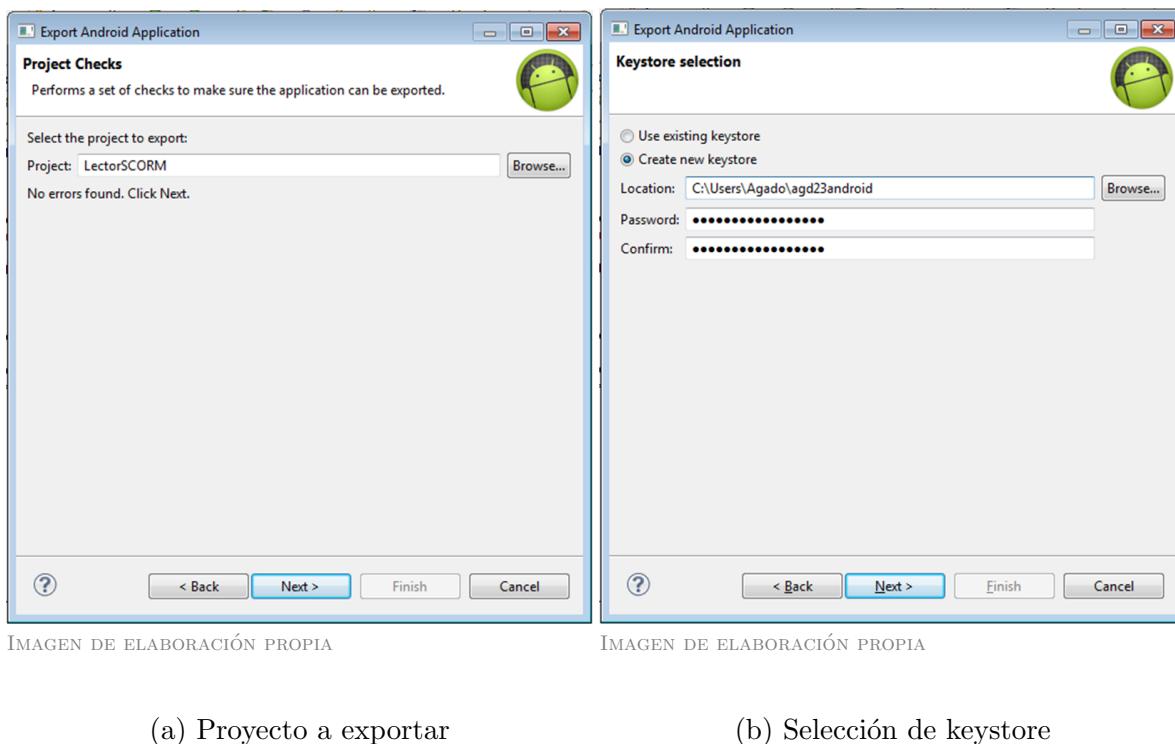


Figura 6.14: Primeros pasos de exportación firmada

A continuación introducimos datos para la identificación del autor de la aplicación y validez del certificado como vemos en la Figura 6.15a.

Cualquier nueva versión de la aplicación debe estar firmada mediante el mismo certificado o no será posible publicar actualizaciones, por esto es muy importante preservar la privacidad de las claves y conservar el archivo keystore que se generará en la localización indicada en la anterior Figura 6.14b

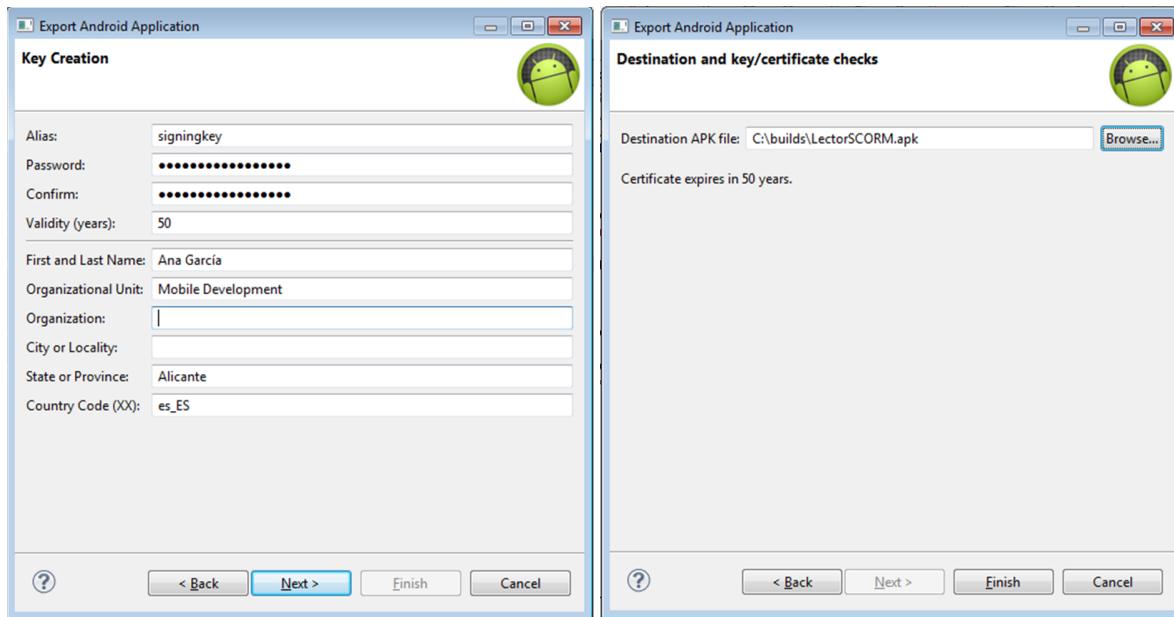


IMAGEN DE ELABORACIÓN PROPIA

IMAGEN DE ELABORACIÓN PROPIA

(a) Datos para el certificado de firma digital

(b) Localización de exportación

Figura 6.15: Datos del certificado y exportación firmada

La localización de exportación de la aplicación firmada digitalmente se especifica en el último paso, en la Figura 6.15b observamos que nos indica el tiempo en el que expira el certificado, este valor se indica en la ventana de la Figura 6.15a. El tiempo de validez es el tiempo de vida de la aplicación, los usuarios solo podrán descargar actualizaciones de la aplicación mientras el certificado esté vigente.

Google recomienda asignar un periodo de validez de mínimo 25 años, en este caso se ha asignado el doble. En estos momentos Google Play no acepta la publicación de aplicaciones cuyo certificado caduque antes del 22 de octubre 2033.

La aplicación se publicará en la plataforma de comercialización de Google.

---

# Capítulo 7

## Conclusiones

Tras el estudio de la plataforma y entorno de desarrollo Android así como el funcionamiento y estructura del estándar SCORM o las aplicaciones educativas para ejecutar contenidos docentes en tablets y las soluciones que estas ofrecen, la aplicación implementada permite ejecutar recursos en este formato en el entorno de una vista web del dispositivo sin necesidad de conexión a Internet ofreciendo de esta forma una solución al problema de enseñanza ubicua y visualización de contenidos SCORM en dispositivos móviles Android, prestando especial atención a las tablets.

Gracias a la posibilidad de importar ficheros a través de la API de Dropbox o de archivos descargados en el dispositivo el usuario puede adaptar la aplicación a sus necesidades de aprendizaje ya sea guiado o autodidacta compartiendo sus progresos a través de las redes sociales u otros medios de comunicación basadas en texto presentes en el dispositivo.

Para que el alcance sea mayor y aumentar la compatibilidad de dispositivos y formato ha sido necesario utilizar la librería de compatibilidad de versiones Android y la implementación de dos adaptadores API para soportar los dos tipos de llamadas de la API SCORM.

Con la definición e implementación de una aplicación de estas características se aborda el problema accesibilidad, interoperabilidad, durabilidad y reutilización de los contenidos de aprendizaje en cuanto a tablets Android.

Como ampliación, los datos recogidos por los adaptadores API podrían almacenarse por curso y usuario, teniendo que implementar para ello un registro de usuarios y una base de datos basada en los nombres de datos que utiliza el estándar SCORM, emulando de esta forma un LMS. Otra característica a añadir los servicios de Google Drive pensando en la posibilidad de integrar alguna de las características de Google

académico u obtener recursos de esta plataforma educativa.

En cuanto al desarrollo tecnológico actual es necesario tener en cuenta las posibilidades que puede proporcionar el uso de todas las funciones que ofrecen los dispositivos móviles. La cámara, la localización del usuario, el acelerómetro, el reconocimiento de voz o la naturalidad de movimientos con la pantalla táctil pueden aportar grandes beneficios a la enseñanza desarrollando cursos que se adapten a dispositivos móviles.

La Tin Can API o Experience API empieza a tener más en cuenta estas características, por ello es conveniente considerar la incorporación de este nuevo estándar a aplicaciones como este lector SCORM si su desarrollo e implantación siguen adelante con éxito.

Aunque todo esto puede llegar a ser un gran reto tecnológico y educativo, la inclusión de nuevas tecnologías en la educación puede suponer una gran experiencia de usuario en la que mejore el aprendizaje del alumno si los recursos se adaptan adecuadamente a la educación.



---

# Bibliografía

- [1] Actiformación. Curso online de introducción a la programación android. URL [www.actiformacion.com](http://www.actiformacion.com). Último acceso: junio de 2014.
- [2] ADL (Advanced Distributed Learning Initiative). ADL initiative. . URL <http://www.adlnet.gov/overview/>. Último acceso: junio de 2014.
- [3] ADL (Advanced Distributed Learning Initiative). Adl scorm. . URL <http://www.adlnet.gov/scorm/>. Último acceso: junio de 2014.
- [4] ADL (Advanced Distributed Learning Initiative). Scorm user guide for programmers. Inf. téc., ADL (Advanced Distributed Learning Initiative), 2011. URL [http://www.adlnet.gov/wp-content/uploads/2011/12/SCORM\\_Users\\_Guide\\_for\\_Programmers.pdf](http://www.adlnet.gov/wp-content/uploads/2011/12/SCORM_Users_Guide_for_Programmers.pdf).
- [5] ADL (Advanced Distributed Learning Initiative). xAPI development plans for 2014. 2014. URL [www.adlnet.gov/xapi-development-plans-for-2014/](http://www.adlnet.gov/xapi-development-plans-for-2014/). Último acceso: 10 julio 2014.
- [6] Android Open Source Project. About Android - Android, the world's most popular mobile platform. . URL <http://developer.android.com/about/index.html>. Último acceso: junio de 2014.
- [7] Android Open Source Project. Android package index. . URL <http://developer.android.com/reference/packages.html>. Último acceso: julio de 2014.
- [8] Android Open Source Project. Building a dynamic ui with fragments. . URL <https://developer.android.com/training/basics/fragments/index.html>. Último acceso: julio de 2014.
- [9] Android Open Source Project. Get the android SDK. . URL <http://developer.android.com/intl/es/sdk/index.html>. Último acceso: junio de 2014.

- [10] Android Open Source Project. Grid view. . URL  
<http://developer.android.com/guide/topics/ui/layout/gridview.html>. Último acceso: julio de 2014.
- [11] Android Open Source Project. Parsing xml data. . URL  
<http://developer.android.com/training/basics/network-ops/xml.html>. Último acceso: junio de 2014.
- [12] Android Open Source Project. Signing your applications. . URL  
<http://developer.android.com/tools/publishing/app-signing.html>. Último acceso: agosto de 2014.
- [13] Anema. Livre blanc SCORM 2004 CAM. Inf. téc., Anema, 2006. URL  
[http://ganesha.fr/public/images/livre-blanc/livre\\_blanc\\_ANEMA\\_-\\_SCORM2004\\_CAM\\_v1-0.pdf](http://ganesha.fr/public/images/livre-blanc/livre_blanc_ANEMA_-_SCORM2004_CAM_v1-0.pdf).
- [14] Antonio Monje. eXeLearning. URL <http://exelearning.net>. Último acceso: junio de 2014.
- [15] AppBrain. Appbrain stats - most popular google play categories. 2014. URL  
<http://www.appbrain.com/stats/android-market-app-categories>. Último acceso: 10 Julio de 2014.
- [16] aprenderaprogramar.com. ¿qué es java? concepto de programación orientada a objetos vs programación estructurada. URL [http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=68&Itemid=188](http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188). Último acceso: julio de 2014.
- [17] Club de Innovación Universitaria UPSA. Scormmobile. 2010. URL <http://www.upsa.es/clubinnovacion/proyectos/2010/scormmobile/index.html>. Último acceso: julio de 2014.
- [18] Creative Commons Corporation. Creative commons. URL  
<http://creativecommons.org/>. Último acceso: julio de 2014.
- [19] David Gañán, Santi Caballé, y Jordi Conesa . Recursos multimedia para aprendizaje on-line. 2013. URL <http://mosaic.uoc.edu/2013/11/11/recursos-multimedia-para-aprendizaje-on-line/>. Último acceso: julio de 2014.
- [20] David Wiley. . Open Content. 1999. URL  
<http://www.opencontent.org/definition/>. Último acceso: julio de 2014.

- [21] Departamento de Lenguajes y sistemas Informáticos. DLSI-Universidad de Alicante. i-scire scorm player para ipad. 2012. URL  
<http://www.i-scire.com/doku.php?id=start>. Último acceso: julio de 2014.
- [22] digitalavmagazine. La plataforma zientia facilita el aprendizaje en las aulas utilizando la realidad aumentada. URL  
<http://www.digitalavmagazine.com/2014/01/21/la-plataforma-zientia-facilita-el-aprendizaje-en-las-aulas-utilizando-la-realidad-aumentada>
- [23] Dropbox. Build the power of dropbox into your app. URL  
[www.dropbox.com/developers](http://www.dropbox.com/developers). Último acceso: julio de 2014.
- [24] Dropbox. Dropbox android sdk 1.6.1. URL <https://www.dropbox.com/static/developers/dropbox-android-sdk-1.6.1-docs/index.html>. Último acceso: agosto de 2014.
- [25] George Mathew. Listing images in gridview using simple adapter in android - knowledge by experience. URL <http://wptraffic analyzer.in/blog/listing-images-in-gridview-using-simple-adapter-in-android/>. Último acceso: julio de 2014.
- [26] Inés Gil-Jaurena y Daniel Domínguez Figaredo. Open social learning y educación superior. oportunidades y retos. Inf. téc., Facultad de educación, Universidad nacional de educación a Distancia (UNED), España, 2012. URL  
<http://www.rieoei.org/rie60a12.pdf>.
- [27] Google Inc. Google play - resultados de la búsqueda aplicaciones scorm educativas. . URL <https://play.google.com/store/search?q=scorm%20education&c=apps&hl=es>. Último acceso: julio de 2014.
- [28] Google Inc. Google play for education. . URL  
<https://www.google.com/edu/play/>. Último acceso: julio de 2014.
- [29] IMS Global Learning Consortium, Inc. Common cartridge working group. URL  
<http://www.imsglobal.org/commoncartridge.html>. Último acceso: julio de 2014.
- [30] Iolanda García (Ed.). El open social learning y su potencial de transformación de los contextos de educación superior en españa. Inf. téc., Universitat Oberta de Catalunya, Barcelona, Cátedra UNESCO de e-learning, eLearn Center, 2010. URL [http://openaccess.uoc.edu/webapps/o2/bitstream/10609/2062/6/informe\\_OSL\\_final.pdf](http://openaccess.uoc.edu/webapps/o2/bitstream/10609/2062/6/informe_OSL_final.pdf).

- [31] ISEA S.Coop. Análisis prospectivo de las potencialidades asociadas al mobile learning. Inf. téc., ISEA Con el apoyo de Ministerio de Industria, Turismo y Comercio, PLAN AVANZA, 2009. URL  
[http://www.iseamcc.net/eISEA/Vigilancia\\_tecnologica/informe\\_4.pdf](http://www.iseamcc.net/eISEA/Vigilancia_tecnologica/informe_4.pdf).
- [32] JCA Solutions Network. Scorm - shareable content object reference model. URL  
[www.scormsoft.com/scorm](http://www.scormsoft.com/scorm). Último acceso: julio de 2014.
- [33] Open Education Consortium. About the open education consortium. URL  
<http://www.oec.org/about-oec/>.
- [34] Open Society Institute (OSI) y Shuttleworth Foundation. The cape town open education declaration. URL <http://www.capetowndeclaration.org/>. Último acceso: julio de 2014.
- [35] Oracle Corporation. Java. URL  
<http://www.oracle.com/es/technologies/java/overview/index.html>.  
Último acceso: julio de 2014.
- [36] Paco Sanguino y Pedro Pernías. Las tecnologías que hacen posibles los contenidos abiertos. 2006. URL <http://tecnologia-contenidos-abiertos.blogspot.com.es/2006/05/las-tecnologias-que-hacen-posibles-los.html>.  
Último acceso: junio de 2014.
- [37] Miguel Rebollo Pedruelo. *El estándar SCORM para EaD*. Proyecto Fin de Carrera, Tesina del Máster en Enseñanza y Aprendizaje Abiertos y a Distancia Universidad Nacional de Educación a Distancia, 2004. URL  
<http://www.mrebollo.es/pubs/tesina.pdf>.
- [38] Fernando Posada. Simulaciones. canaltic.com, 2012. URL  
<http://canaltic.com/blog/?p=849>. Último acceso: julio de 2014.
- [39] Mike Rustici. Scorm api discovery algorithms. 2009. URL <http://scorm.com/scorm-explained/technical-scorm/run-time/api-discovery-algorithms/>.
- [40] Rustici software. SCORM explained. URL  
<http://scorm.com/scorm-explained/>. Último acceso: julio de 2014.
- [41] Rustici Software, LLC. Tin can api. URL <http://tincanapi.com/>. Último acceso: julio de 2014.
- [42] Stack Exchange Network inc. Stackoveflow android. URL  
<http://stackoverflow.com/questions/tagged/android>. Último acceso: junio de 2014.

- [43] Stack Exchange Network inc - Kanika. Download file from dropbox and save it into sdcard. 2012. URL <http://stackoverflow.com/questions/9566884/download-file-from-dropbox-and-save-it-into-sdcard>. Último acceso: agosto de 2014.
- [44] Stack Exchange Network inc - swapnesh. How to develop lms api for scrom in .net. URL <http://stackoverflow.com/questions/19005100/how-to-develop-lms-api-for-scrom-in-net>. Último acceso: agosto de 2014.
- [45] w3c - World Wide Web Consortium. Extensible markup language (xml). . URL <http://www.w3.org/XML/>. Último acceso: julio de 2014.
- [46] w3c - World Wide Web Consortium. Guía breve de tecnologías xml. . URL <http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>. Último acceso: julio de 2014.