UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

SCORMVIEW: VISOR DE PAQUETES SCORM PARA XAUCEMÓVIL

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores: **Dianelys Pérez González**
Sandy Nuñez Padrón

Tutores: **Ing. Yerandy Manso Guerra**
Ing. José González Castellanos

La Habana, 2015

"Año 57 de la Revolución"

Declaración de autoría

Declaramos que somos los únicos autores del trabajo *SCORMView: Visor de paquetes SCORM para XauceMóvil* y autorizamos a los Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2015.

_____

Firma del Autor

Dianelys Pérez González

_____

Firma del Autor

Sandy Nuñez Padrón

_____

Firma del Tutor

Ing. Yerandy Manso Guerra

_____

Firma del Tutor

Ing. José González Castellanos

Dedicatoria

A mi mamá por estar siempre incondicionalmente para mí y por ser mi motor impulsor. Te quiero mucho.

A mi abuela Amelia que, por su cariño y sus consejos, me enseñó a ser una mejor persona. Por ser como otra madre para mí y darme todo su amor.

Dianelys Pérez González

Al ser al cual no hay nada en el mundo que le pueda pagar todo lo que ha hecho por mí, todo lo que ha sufrido y las privaciones por las que ha pasado para hacerme todo un hombre.

A esa persona a la cual le tengo que agradecer todo lo que tengo y lo que soy porque sin ella no hubiera sido posible alcanzarlo.

A esa mujer que siempre me ha ayudado, apoyado y animado a alcanzar mis metas, sueños y aspiraciones

A mi mamá.

Sandy Nuñez Padrón

Agradecimientos

A mi mamá Lilian por darme todo su amor, por siempre apoyarme y haberme impulsado coger esta carrera. Por ser mi fuerza, por siempre sacrificarse por mí y estar ahí sin quejas cuando tengo algún problema.

A mi abuela linda Amelia que aunque ya no este entre nosotros fue un gran a apoyo, junto a mi mamá, para continuar este duro camino. Aunque no pudo llegar a verme graduada, sé que a ella le hubiera encantado estar aquí, yo sé que donde quiera que esté ella se siente muy feliz por mí.

A mi tía Mercedes por darme todo su cariño y ser para ella la niña linda de la casa. Por su apoyo desde que estaba en la primaria ya que cada vez que tenía un trabajo práctico ella siempre trataba de ayudarme junto con mi abuela.

A mi tío Humberto por ser como un padre para mí y siempre estar cuando lo necesitaba.

A mi tío Ubaldo que también ha sido como otro padre para mí.

A mis primos (Diana, Yunelkis y Yoandri) ya que siempre se preocuparon y me ayudaron en estos 5 años.

A mi papá Alajandro por haberme dado también un impulso para coger la UCI y por haberse portado mal-bien en estos 5 años.

A mi compañero de tesis Sandy por haberme aguantado estos 5 años y más este último. Por ser siempre un amigo para mi desde que empecé en la UCI.

A mis amigas del piquete "las 5 inseparables"(Dairelis, Evelyn, Yosle y Yelaine) que desde que llegamos en primer año hicimos una gran amistad. Por el apoyo que siempre nos dimos entre todas. Por los buenos momentos que pasamos en los campismos,

Varadero, etcétera.

A mi amigo Yoan el #1 para formar los club de estudios y el #1 para disociarlos. Por sus ocurrencias que siempre nos hicieron reír.

A mi novio Roberto Daniel, que aunque llevamos poco tiempo lo he aprendido a querer mucho a pesar de que él diga que no. Por haber estado apoyándome en estos momentos finales, ayudándome a ensayar la exposición y diciéndome cuando está mal.

A mi prima Julita que siempre me ayudó, más es este último año, aportando su gran grano de arena haciendo esta tarea un poco más facil.

A mis profesores, gracias por todo lo que me enseñaron en estos cinco años.

A mis tutores por darme su apoyo en esta tarea final.

Dianelys Pérez González

A Fidel Castro Ruz, Raúl Castro Ruz, la Revolución Cubana y todos los que lucharon por la independencia de este archipiélago por darme la oportunidad de estudiar y convertirme en un ingeniero.

A mis padres Ana Llusimí Padrón Yanes y Fernando A. Nuñez Vals por haber confiado en mí y apoyarme incondicionalmente.

A mi abuela Daysi D. Llanes Torres por tantas velas a San Lázaro y porque como dice ella solo le faltó pujarme.

A mi abuela Amada Vals Suárez y a mi abuelo Arnaldo S. Padrón Peña por estar tan orgullosos de mí.

A mi primo Ángel Martínez Padrón por ser el hermano que nunca tuve.

A mi tía Yamilé Padrón Yanes por ser mi chicharrona y siempre estar ahí para lo que necesite.

A Juan Carlos Baldanca por tantos corretajes.

Al resto de mi familia por tenerme en una posición que creo no merecer.

A mi compañera de tesis Dianelys Pérez González por ser una amiga incondicional, por preocuparse por mí y por tener que soportar mis berrinches.

A mi amiga Evelyn Pérez Rosa, esa flaca pinareña a la que tanto jodí y que se fajaba conmigo porque dejaba de comer.

A mi amiga Dairelis Lahera Hechevarría por tantos momentos vividos.

A mi amiga Yelaine Sánchez Oliú por todas sus locuras.

A Yosleidy Arteaga Gómez por su matraquilla con la organización.

A mi amigo Yoan Pérez Alfonso por todas sus ocurrencias.

A mis amigos Carlos A. Rojas Lugones y Reimnel Drake Bueno por confiar en mí y los magníficos momentos haciendo radio y televisión.

A mi amigo Yoelkys Hernández Aracil por tenerme en tan alta estima y ceer en mí.

A la FEU por tantos momentos alegres, duros, tensos y difíciles que me enseñaron mucho.

A Código y Letra, el Portal Octavitos, RCD y UCITeVe por darme las oportunidad de descubrir mi pasión por el periodismo y los medios de comunicación.

A mis compañeros de grupo y año por cruzar juntos este camino de 5 años.

A mis tutores por haberme guiado en la realización de esta investigación.

A mis profesores por abrirme las puertas al conocimiento.

A la Facultad 4 por recibirme y darme la oportunidad de ser un Octavito de corazón.

A la Universidad de las Ciencias Informáticas por darme la oportunidad de aprender una profesión que tanto me apasiona.

Sandy Nuñez Padrón

Resumen

Entre los estándares que existen en la actualidad para el área de la tecnología educativa, se destaca la especificación *Sharable Content Object Reference Model*, conjunto de estándares y especificaciones que permite crear objetos pedagógicos estructurados. En la Universidad de las Ciencias Informáticas desde hace algunos años se desarrollan objetos de aprendizaje empaquetados bajo este estándar con la herramienta de autor Creando Objetos de Aprendizaje, los cuales se almacenan en un Repositorio de Objetos de Aprendizaje RHODA y son utilizados por la Plataforma Educativa ZERA y Moodle, las cuales también utilizan este estándar para exportar sus propios contenidos. En la actualidad, debido al auge de los dispositivos móviles en la educación a través de la estrategia de aprendizaje *mobile learning*, los servicios de RHODA, ZERA y Moodle se están integrando en la plataforma XauceMóvil para que los usuarios de esta tecnología tengan acceso a ellos; razón por la cual se necesita un visor que permita la descompresión e interpretación de dichos paquetes, sin embargo, no se ha encontrado una herramienta que pueda reproducirlos en el sistema operativo Android. Por tanto, el objetivo de la presente investigación es desarrollar un visor de contenidos que permita a XauceMóvil interpretar y visualizar los contenidos empaquetados bajo el referido estándar.

Palabras clave: SCORM; visor; Android; *m-learning*; tecnología educativa; dispositivos móviles.

Índice general

Introducción

"Las Tecnologías de la Información y la Comunicación, son un conjunto de servicios, redes, *software* y dispositivos que tienen como fin la mejora de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario" [Ruiz, 2009].

Esta definición muestra claramente lo que representan las Tecnología de la Información y las Comunicaciones (TICs) en la sociedad del siglo XXI, pues su indetenible desarrollo ha impactado en varios ámbitos y niveles de vida erigiéndose como una herramienta importante para el progreso y el apoyo a las diferentes actividades que se realizan dentro de un centro de trabajo.

Hoy, el mundo se encuentra en una época de cambios donde la sociedad, dentro del amplio espectro de las TICs, trata cada vez más de independizarse de los medios computacionales tradicionales, lo cual está logrando gracias a la utilización de los dispositivos móviles que con el paso del tiempo han aumentado su nivel de utilidad y funcionalidad.

Según Anaid Guevara Soriano en su artículo "Dispositivos móviles" publicado en 2010 por la revista "Seguridad" de la Universidad Nacional Autónoma de México, expresa que los dispositivos móviles tienen características tales como [Guevara Soriano, 2010]:

- Capacidades especiales de procesamiento.
- Conexión permanente o intermitente a una red.
- Memoria limitada.

- Diseños específicos para una función principal y versatilidad para el desarrollo de otras funciones.
- Tanto su posesión como su operación se asocia al uso individual de una persona, la cual puede configurarlos a su gusto.

Se puede afirmar, que los dispositivos móviles han revolucionado el área de las comunicaciones por sus ventajosas prestaciones lo cual ha provocado que muchas personas los necesiten en el trabajo, la escuela, la casa, en fin, donde quiera que estén. Este fenómeno se ha dado producto de que con el paso del tiempo y de los avances de la ciencia, estos dispositivos han llegado a convertirse en pequeñas computadoras personales cuyo funcionamiento se ha desarrollado hasta convertirse en productos de consumo masivo y su aceptación se hace evidente a medida que transcurren los años.

Su influencia ha llegado hasta la educación ofreciendo varias ventajas, muchas de las cuales, según el artículo “Impacto de la tecnología móvil en la educación” publicado el 25 de junio de 2012 en el periódico “UniVerso” de la Universidad Veracruzana, requieren replantear metodologías y estándares en la educación; principalmente en la forma en que los profesores se comunican con sus alumnos [UniVerso, 2012]. Algunas de ellas son:

- *Permite la comunicación en tiempo real entre alumnos, profesores y directivos:* el uso de redes sociales y aplicaciones de mensajería instantánea entre alumnos y profesores agiliza el proceso de comunicación y reduce tiempos.
- *Agiliza y vuelve más eficiente la distribución de contenidos y materiales:* aplicaciones y servicios que permiten almacenar información en la nube y compartirla con otras personas.
- *Elimina la barrera geográfica en el aprendizaje:* ya no es necesario que alumnos y maestros estén todo el tiempo en un salón de clases; además, que permite que el aprendizaje llegue a lugares remotos.

- *Promueve que los estudiantes sean más activos durante el proceso de aprendizaje:* ya sea por medio de realizar investigaciones, aprender y utilizar nuevas tecnologías, crear documentos y compartirlos, etcétera.

Los dispositivos móviles pueden ser aprovechados en la docencia virtual. Permiten el uso de contenido multimedia en cualquier lugar, ofreciendo al usuario información enriquecida y formación interactiva a través de una multitud de recursos educativos accesibles para el usuario como apoyo a la educación presencial. Debido a ello, con el paso del tiempo se ha incrementado la cantidad de aplicaciones educativas para dispositivos móviles que han jugado un rol importante en la formación y elevación del nivel cultural de los usuarios, lo cual ha posibilitado el nacimiento de un nuevo concepto: " aprendizaje móvil (*m-learning*)".

Aprendizaje móvil "es una evolución del *e-learning*, que a su vez es una evolución de la formación a distancia. Significa aprendizaje electrónico móvil o educación móvil y es, como su propio nombre indica, una metodología o difusión de contenidos de enseñanza y aprendizaje a través del uso de pequeños y maniobrables dispositivos móviles..." [Rubio Gómez et al., 2011]

En Cuba, a pesar de que los dispositivos móviles por cuestiones económicas no han podido ser adquiridos por la mayor parte de la población, es cierto que cada día aumenta exponencialmente la cantidad de usuarios interesados en la adquisición de esta importante vía de comunicación. También existe una voluntad política de dominar e introducir en la práctica social las tecnologías de la información y las comunicaciones, lograr una cultura digital como una de las características imprescindibles del hombre nuevo, lo que facilitaría al pueblo cubano acercarse más hacia el objetivo de un desarrollo sostenible.

En este sentido en el lineamiento 131 de la política económica y social del Partido y la Revolución se plantea: "Sostener y desarrollar los resultados alcanzados en el campo de la biotecnología, la producción médico-farmacéutica, la industria del *software* y el proceso de informatización de la sociedad, las ciencias básicas, las ciencias natu-

rales, los estudios y el empleo de las fuentes de energía renovables, las tecnologías sociales y educativas, la transferencia tecnológica industrial, la producción de equipos de tecnología avanzada, la nanotecnología y los servicios científicos y tecnológicos de alto valor agregado"[de Cuba, 2011].

La Universidad de las Ciencias Informáticas (UCI), constituye la sede de varios centros de producción de *software*. Cuenta con diferentes líneas de investigación donde se ha comenzado a fomentar el uso de esta nueva tecnología que va incrementando su auge a nivel nacional. Específicamente en la Facultad 4 radica el Centro de Tecnologías para la Formación (FORTES), en el cual se desarrollan entre otros *software* la Plataforma Educativa ZERA, el Repositorio de Objetos de Aprendizaje RHODA y diferentes desarrollos para la plataforma Moodle.

Estas plataformas para lograr que sus contenidos sean interoperables con otros sistemas y entre ellos hacen uso de estándares y especificaciones de carácter internacional, siendo el estándar *Sharable Content Object Reference Model (SCORM)* el seleccionado por estas para intercambiar sus contenidos y cursos.

SCORM es "un conjunto de especificaciones para desarrollo, empaquetamiento y distribución de material educativo, en cualquier momento y en cualquier lugar. El estándar SCORM asegura que este material sea: reutilizable, accesible, interoperable y durable" [Rodríguez, 2012].

Dentro del contexto de los dispositivos con sistema operativo Android, existen miles de aplicaciones educativas, más de 90.000 en la categoría educación de la plataforma comercial de Google, sin embargo, pese a la gran cantidad de recursos educativos existentes, si nos centramos en aplicaciones para Android con la etiqueta SCORM el número se reduce drásticamente y el código de las pocas existentes no permite ser reutilizado ni adaptado y no es distribuido. Además, estas necesitan de una conexión constante a internet y dependen de un *Learning Management Systems (LMS)* en específico.

Precisamente en FORTES se está desarrollando XauceMóvil, una aplicación para dispositivos móviles con sistema operativo Android. Entre las funcionalidades que tie-

ne esta aplicación se encuentra la de mostrar los contenidos de estas plataformas y con esto la posibilidad de integrar los paquetes SCORM, por lo que necesita un visor que le permita descomprimir el paquete, interpretarlo y mostrar sus contenidos.

Por lo antes planteado se presenta la siguiente **problemática a resolver:** ¿Cómo interpretar en XauceMóvil los paquetes que cumplan con el estándar SCORM y visualizar los contenidos que se encuentren dentro de este en dispositivos móviles?

Objeto de estudio: visores de paquetes que cumplen con el estándar SCORM.

Campo de acción: visores de paquetes que cumplen con el estándar SCORM en dispositivos móviles.

Objetivo general: desarrollar un visor de paquetes SCORM que le permita a XauceMóvil interpretar y visualizar los contenidos en dispositivos móviles.

Objetivos específicos:

- Elaborar el marco teórico de la investigación mediante el estudio del estado del arte acerca de las tendencias tecnológicas actuales y las estrategias utilizadas en el desarrollo de visores de archivos SCORM para el sistema operativo Android.

- Realizar el análisis, diseño e implementación de un visor de paquetes SCORM para el sistema operativo Android que contenga características de sistemas similares y se integre con la aplicación XauceMóvil.

- Realizar pruebas a la propuesta de solución desarrollada para interpretar y visualizar los contenidos que contienen los paquetes SCORM.

Hipótesis

El desarrollo de un visor de paquetes SCORM facilitará que XauceMóvil interprete y visualice contenidos empaquetados bajo este estándar en dispositivos móviles.

Posibles resultados:

Una vez finalizado el presente trabajo, se tendrá como resultado un visor que le permita a XauceMóvil interpretar y visualizar el contenidos de paquetes SCORM.

Tareas a cumplir:

- Realización de un análisis de las diferentes tecnologías para el desarrollo de aplicaciones para el sistema operativo Android.
- Realización de un análisis acerca de la estrategia de aprendizaje *mobile learning (m-learning)*.
- Realización de un análisis acerca de las metodologías de desarrollo de *software* existentes.
- Selección de la metodología y las tecnologías a usar para darle cumplimiento al objetivo propuesto.
- Identificación de los requisitos funcionales y no funcionales de la solución propuesta.
- Delimitación de las restricciones con las que tienen que cumplir los paquetes SCORM que van a ser leídos e interpretados por la aplicación a desarrollar.
- Realización del análisis y diseño de la aplicación propuesta.
- Generación de los principales artefactos de la metodología de desarrollo seleccionada.
- Implementación de la aplicación haciendo uso de las tecnologías y herramientas seleccionadas.
- Realización de pruebas a la aplicación.

Para llevar a cabo la presente investigación, se emplearon métodos científicos de investigación del nivel teórico y del empírico.

Métodos teóricos:

- *Histórico-lógico:* Se utilizó este método para realizar el estudio del estado del arte y de la evolución del estándar SCORM, así como del uso de los visores

de paquetes SCORM en la actualidad, investigar acerca de otras aplicaciones o soluciones similares y de los lenguajes y metodologías de desarrollo de software existentes.

- *Analítico-sintético:* Se empleó en el estudio de las tecnologías que se emplearán para el desarrollo de la propuesta y en el análisis de la documentación relacionada con el objeto de estudio, así como en la revisión bibliográfica.

Métodos empíricos:

- **Observación:** Permitió estudiar de cerca el objeto de la investigación, las acciones, causas y consecuencias logrando conocer la esencia del problema planteado, analizando desde varios puntos de vista la propuesta de solución y otras soluciones existentes e identificando qué está hecho y qué falta por hacer..

Estructura capitular

- *Capítulo I:* Fundamentación teórica. Se presenta un estudio del estado del arte. Se realiza un estudio sobre la visualización de paquetes SCORM en aplicaciones de escritorio, repositorios de objetos de aprendizaje y plataformas educativas. También se enuncian algunas de las características que presentan las tecnologías a emplear en el desarrollo de la investigación.

- *Capítulo II:* Propuesta de solución. Se realiza el levantamiento de requisitos funcionales y no funcionales de la solución propuesta. Se definen las historias de usuarios así como otros artefactos generados por la metodología de software seleccionada.

- *Capítulo III:* Implemantación y pruebas. Se realiza una evaluación de los parámetros expuestos en el diseño para una correcta implementación, atendiendo a la descripción de las principales funcionalidades. Además, se le realizarán pruebas a la solución propuesta, comprobando la correcta ejecución de las funcionalidades implementadas.

Capítulo 1

Fundamentación teórica

En el presente capítulo se precisan un conjunto de conceptos y fundamentos que constituyen el marco teórico relacionado con el objeto de estudio definido en la investigación. Se destacan las principales características de los elementos asociados a la temática a investigar, así como el entorno nacional e internacional donde se desarrollan. Se exponen la metodología y las principales herramientas que sirven de apoyo para la búsqueda de una solución a la problemática planteada.

1.1. Conceptos y definiciones asociados al problema de investigación

1.1.1. Dispositivos móviles

El diccionario de informática y tecnología define dispositivo móvil como el término genérico que describe computadoras tan pequeñas que entran en un bolsillo. Puede usarse como sinónimo de *handheld*, y se consideran un tipo de computadora móvil. Suelen tener una pantalla y botones pequeños, aunque algunos carecen totalmente de botones y se manejan con pantallas táctiles [Aglesa, 2012].

Mientras que Carolina Izarra expresa que “son pequeñas computadoras, tan pe-

queñas que entran en un bolsillo. Suelen tener una pantalla y botones pequeños, aunque algunos carecen totalmente de botones y se manejan con pantallas táctiles" [Izarra, 2010].

Por su parte, el ingeniero Juan Manuel Álvarez, lo define como aparatos de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales [Álvarez, 2008].

Entre sus características principales se encuentran:

- Se asocian al uso individual de una persona, tanto en posesión como en operación, el cual puede adaptarlos a su gusto.
- Pueden ser transportados en el bolsillo del propietario.
- Otros están integrados dentro de otros mayores, controlando su funcionalidad (como puede ser el ordenador integrado en una lavadora).

Tipos

Existen varios tipos de dispositivos móviles como:

Celular: Dispositivo electrónico de comunicación, normalmente de diseño reducido y basado en la tecnología de ondas de radio, que tiene la misma funcionalidad que cualquier teléfono de línea fija. Los más modernos suelen incorporar un conjunto de funciones adicionales, tales como mensajería instantánea (SMS), agenda, juegos, entre otros, que aumentan sus potencialidades. Pueden usarse para debates grupales y también son útiles para proyectos que requieren fotografías.

Asistente Personal Digital (PDA): Agenda personal electrónica (calendario, lista de contactos, bloc de notas y recordatorios), con un sistema de reconocimiento de escritura y que tienen capacidad para almacenar datos. Se puede usar como una computadora doméstica ya que permite ver películas, crear documentos, jugar, revisar el correo electrónico, navegar por Internet y escuchar música.

Smartphone: Tienen muchas funciones similares a las de una computadora y también funciona como teléfono. Puede ejecutar aplicaciones y software, grabar audio y video, enviar y recibir e-mails y mensajes de texto. Suele incorporar diversas aplicaciones ofimáticas y que requieren de una conexión a una computadora para sincronizar y actualizar correctamente los datos entre ambos dispositivos. Cuentan con cámaras, Sistema de Posicionamiento Global (GPS) y bluetooth. Le da soporte a redes Wireless Fidelity (WiFi), 3G12 y 4G12 para el acceso a Internet. Permiten el almacenamiento de hasta 32 Gb de información y sus pantallas generalmente no exceden las 5 pulgadas. Se le puede introducir tarjetas de memorias extraíbles (micro SD) e incluyen altas prestaciones técnicas como Memoria de Acceso Aleatorio (RAM) de más de 1 Gb y microprocesadores de hasta 8 núcleos; todo esto a escalas realmente reducidas.

Tablet: Tipo de computadora portátil, de mayor tamaño que un teléfono inteligente o un PDA, el cual cuenta con una pantalla táctil. Técnicamente los tablets se ven dotados de la mayoría de las características de los smartphones y los libros electrónicos. Su conectividad es mayor que la de los teléfonos inteligentes, al igual que su velocidad de procesamiento. A pesar de que estos no están provistos de líneas para efectuar llamadas telefónicas, cuentan con servicios de internet como video llamadas y chats que permiten la comunicación entre varios usuarios. Las aplicaciones descargables hacen que estos dispositivos sean casi comparables con las computadoras. Con ellos se puede jugar, mirar e incluso hacer películas y tomar fotografías.

1.1.2. Android

Android es una plataforma desarrollada por Google y el consorcio Handset Alliance 6. Presenta una serie de características que lo hacen diferente [Gironés, 2011]:

- Plataforma realmente abierta: Es una plataforma de desarrollo libre basada en Linux y de código abierto.
- Portabilidad asegurada: las aplicaciones finales son desarrolladas en Java lo que asegura que podrán ser ejecutadas en gran variedad de dispositivos.

- Arquitectura basada en componentes basada en internet.
- Filosofía de dispositivo siempre conectado a internet.
- Alto nivel de seguridad: Los programas se encuentran aislados unos de otros gracias al concepto de ejecución dentro de una caja que incorpora la máquina virtual.
- Optimización para baja potencia y poca memoria: Android utiliza la máquina virtual Dalvik, que es una implementación de la máquina virtual de Java para dispositivos móviles.

Android, al contrario que otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se despliega de forma abierta y se permite el acceso tanto al código fuente como a la lista de incidencias donde se pueden ver problemas aún no resueltos y reportar problemas nuevos. Se usa en disimiles dispositivos como son teléfonos inteligentes, ordenadores portátiles, notebooks, tablets, Google TV, relojes de pulsera, auriculares y otros.

Arquitectura de Android

La arquitectura de Android está formada de cuatro capas, las cuales están basadas en *software* libre. [Gironés, 2011]

1. *Núcleo Linux:* el núcleo de Android está formado por el sistema operativo Linux versión 2.6. Esta capa proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de *drivers* para dispositivos. Esta capa del modelo actúa como capa de abstracción entre el *hardware* y el resto de la pila. Por lo tanto, es la única que es dependiente del *hardware*.
2. *Entorno de ejecución (Runtime) de Android:* dado a las limitaciones de los dispositivos donde debe correr el sistema operativo Android (poca memoria y procesador limitado) no fue posible utilizar la máquina virtual de Java. Google tomo

la decisión de crear una nueva, la máquina virtual Dalvik, que responde mejor a estas limitaciones.

Algunas características de la máquina virtual Dalvik que facilitan la optimización de recursos son: que ejecuta ficheros Dalvik ejecutables (.dex)-formato optimizado para ahorrar memoria. Además está basada en registros. Cada aplicación corre en su propio proceso Linux con su propia instancia de la máquina virtual Dalvik. Delega al núcleo de Linux algunas funciones como el manejo de la memoria a bajo nivel.

3. *Librerías nativas:* incluye un conjunto de librerías C/C++ usadas en varios componentes de Android. Están compiladas en el código nativo del procesador. Algunas de estas librerías son:

   - *System C library:* una derivación de la librería BSD de C estándar (libc), adaptada para dispositivos embebidos basados en Linux.
   - *Media Framework:* librería basada en PacketVideo's 7 Open CORE; soporta *codecs* 8 de reproducción y grabación de multitud de formatos de audio, video e imágenes.
   - *Web Kit:* soporta el moderno navegador web utilizado en el navegador de Android y el vista webView.
   - *Librerías 3D:* implementación basada en OpenGL ES 1.0 API. Las librerías utilizan el acelerador de *hardware* 3D si está disponible, o el software altamente optimizado de proyección 3D.
   - *SQLite:* potente y ligero motor de bases de datos relacionales disponible para todas aplicaciones.
   - *SSl:* proporciona servicios de encriptación de capa de conexión segura.

   Estas librerías normalmente están hechas por el fabricante, quien también se encarga de instalarlas en el dispositivo antes de ponerlo a la venta. El objetivo de

las librerías es proporcionar funcionalidad a las aplicaciones para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez y garantizando que se llevan a cabo de la forma "más eficiente".

4. *Entorno de aplicación:* proporciona una plataforma de desarrollo libre con gran riqueza e innovaciones. Esta capa está formada por todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones. La mayoría de los componentes de esta capa son librerías Java que acceden a los recursos de las capas anteriores a través de la máquina virtual Dalvik.

5. *Aplicaciones:* este nivel está formado por el conjunto aplicaciones instaladas en una máquina Android, tanto las que tienen interfaz de usuario como las que no, las nativas (programadas en C o C++) y las administradas (programadas en Java), las que vienen pre-instaladas en el dispositivo y aquellas que el usuario ha instalado. Todas las aplicaciones han de correr en la máquina virtual Dalvik para garantizar la seguridad del sistema.

1.1.3. Tipos de aplicaciones

Cuando se va desarrollar una aplicación para el sistema operativo Android, los programadores tienen que elegir el tipo de aplicación que se debe desarrollar en función de lo que se quiere lograr teniendo dos opciones fundamentales: los sitios web móviles y las aplicaciones nativas. [González Castellanos, 2013]

Un sitio web móvil es una aplicación de Internet a la cual se accede a través de un navegador instalado en un dispositivo móvil y está basada en la Web. Los usuarios tienden a acceder a él para tener acceso inmediato a la información, como el correo electrónico, redes sociales o las compras de productos.

Su estructura se basa en el HyperText Markup Language (HTML) y la principal diferencia con los sitios web tradicionales se encuentra fundamentalmente en el diseño de los mismos, debido a las características que tienen los dispositivos móviles (tamaño

de la pantalla e interfaz táctil).

Por otro lado, una aplicación nativa (app) es una aplicación de software que puede ser descargada desde un repositorio determinado o desde una computadora y se instala en dispositivos móviles para ayudar al usuario en una tarea determinada, ya sea de carácter profesional o de ocio y entretenimiento.

Estas aplicaciones se ejecutan nativamente desde el dispositivo aprovechando de una mejor forma los recursos del mismo (GPS, memoria, acelerómetro, notificaciones push20, cámara), así como el acceso a estos. Su objetivo es facilitar la consecución de una tarea determinada o asistir en operaciones y gestiones del día a día.

A partir de una comparación (Tabla 1) entre las características, ventajas, posibilidades y restricciones que tiene cada tipo de aplicación antes mencionada y el objetivo que se desea alcanzar con el desarrollo del visor, se considera que el tipo de aplicación que más se ajusta a las necesidades de esta investigación, es la nativa.

Sitio web móvil	Aplicación nativa
Es compatible con todos los sistemas operativos.	Es compatible con un solo sistema operativo.
Requiere una conexión a internet.	No necesita una conexión a internet.
Acceso muy limitado al dispositivo.	Acceso completo al dispositivo.
El usuario siempre dispone de la última versión.	El usuario no siempre dispone de la versión más actualizada.
No necesitan una aprobación externa para publicarse pero requiere de mayor esfuerzo en promoción y visibilidad.	Necesita tener visibilidad en APP Store.
La experiencia del usuario y el tiempo de respuesta son menores.	Mejor experiencia del usuario.

El mismo código base puede reutilizarse en otras plataformas.	El código del cliente no es reutilizable entre las plataformas existentes.
Necesitan de un navegador web.	No necesita otros programas.

Tabla 1.1: Comparación entre sitio web móvil y aplicación nativa

1.1.4. Usos en la educación

Debido a lo antes expuesto, muchos sectores de la sociedad han decidido incluir los móviles en sus procesos. Un ejemplo clave de lo anterior es la educación que se ha servido de estos para apoyar el proceso de enseñanza-aprendizaje.

Marcello Rinaldi, en su libro a cerca del *m-learning*, destaca cómo el aprendizaje a través de los dispositivos móviles marcará tendencia en poco tiempo. De ahí la necesidad de añadirlos a los proyectos de formación existentes y de futura creación, pues su uso en las actividades de aprendizaje facilita la asimilación de contenidos y ayuda a poner en práctica los contenidos abordados [Rinaldi, 2011].

Desde la décadas de los 80 diversos conceptos han descrito el fenómeno de la educación a distancia a medida que han ido produciéndose los avances tecnológicos: enseñanza apoyada por ordenador, multimedia educativa, tele-educación, enseñanza basada en entorno Web, aprendizaje electrónico (e-learning), etcétera.

Estos avances tecnológicos de los últimos años han posibilitado que se desarrollen métodos de estudio más dinámicos, completos e interactivos.

1.1.5. Aprendizaje móvil

Durante las últimas décadas han aparecido distintas alternativas para aprender fuera de un aula que aprovechan los avances tecnológicos, los cuales han introducido diversos dispositivos que manejan información digital. De esta forma surgió el *e-learning* que se vale de los medios de cómputo tradicionales para la educación a

distancia.

Si a esto le sumamos la evolución de las redes inalámbricas que han permitido la movilidad del usuario acortando distancias y tiempos de respuesta en el proceso de enseñanza-aprendizaje, es fácil entender cómo los dispositivos móviles han cobrado enorme importancia en la educación. Así, se han desarrollado múltiples aplicaciones y tecnologías que apoyan las actividades educativas provocando el surgimiento de *m-learning* que cada día está tomando más fuerza ya que estos dispositivos con el paso del tiempo se están haciendo más maniobrables y ofrecen una mayor cantidad de facilidades y/u oportunidades que pueden ser aprovechadas para impulsar y mejorar los procesos de aprendizaje.

Este novedoso término es definido por diferentes autores como Mariano G. J. quien plantea que es “un conjunto de prácticas y metodologías de enseñanza y aprendizaje mediante tecnología móvil” [Mariano, 2008].

Por su parte, Carlos Hernán establece que “se denomina aprendizaje electrónico móvil, en inglés, *m-learning*, al proceso metodológico de enseñanza y aprendizaje a través de dispositivos móviles, tales como teléfonos móviles, celulares, i-pods, entre otros dispositivos de capacidad inalámbrica, proporcionando a los estudiantes que se encuentren en constante desplazamiento, el acceso a la educación a distancia y virtual, con la finalidad de brindarles alternativas interactivas y de acceso educativo, ofreciendo así mayor flexibilidad para aprender en el momento que decida y en el lugar que lo requiera” [Mora G, 2010].

Mientras tanto los autores Luís Alejandro Flétscher y Álvaro Ignacio establecen que “conceptualmente se puede afirmar que se denomina *m-learning* a la difusión de contenidos formativos mediante dispositivos móviles” [Flétscher Bocanegra and Morales González, 2006]

A su vez Vazquez-Reina sostiene que “es una adaptación del *e-learning* a los nuevos dispositivos móviles (teléfono, PDA, MP3/MP4 o consolas portátiles) de uso común entre los jóvenes. Apuesta por incorporarlos a las aulas como un recurso tecnológico más para potenciar el aprendizaje y aprovechar las destrezas digitales de los alumnos”

[Vazquez-Reina, 2011].

Se puede concluir que el *m-learning* es una rama del *e-learning*, que utiliza los dispositivos móviles con conexión inalámbrica para llevar a cabo el proceso de enseñanza-aprendizaje ofreciendo una mayor flexibilidad al alumno para aprender en el momento que decida y en el lugar que lo requiera.

Entre sus características están [Moreno Guerrero, 2011]:

- *Utilización de juegos de apoyo en el proceso de formación:* la variedad de juegos generados para móviles, impulsa la creatividad y la colaboración.
- *Independencia tecnológica de los contenidos:* una lección no está hecha para un dispositivo concreto.
- *Disponibilidad:* Todas las actividades online del espacio de formación están disponibles para dispositivos móviles.
- Navegación sencilla y adaptación de contenidos teniendo en cuenta la navegabilidad, procesador y velocidad de conexión de estos dispositivos.
- *Acceso inmediato a datos y avisos:* los usuarios pueden acceder en forma rápida a mensajes, correos, recordatorios y noticias generados en tiempo real.
- Adaptabilidad a los ritmos de aprendizaje del estudiante.
- El aprendizaje parte del interés personal de los usuarios, los cuales requieren servicios adaptados a sus propias necesidades.
- El aprendizaje se da en los diferentes contextos sociales en los que participa el aprendiz, por ello un servicio móvil debe generar servicios para el estudiante y la institución laboral o escolar.
- El dispositivo móvil será el factor posibilitador o limitante de los procesos cognitivos del estudiante.

En cuanto a las ventajas del *m-learning* tenemos [Moreno Guerrero, 2011]:

- Aprendizaje significativo, a través del diseño de ambientes instruccionales que propicien experiencias de acuerdo a la realidad del alumno.
- La telefonía móvil está al alcance de casi todos, puesto que es más barata que las computadoras convencionales.
- Cuentan con una interfaz amigable y herramientas atractivas.
- El dispositivo móvil puede ser usado en cualquier parte y en cualquier momento, incrementando el tiempo para desarrollar actividades de enseñanza-aprendizaje.
- Portabilidad (dada por el tamaño y peso del dispositivo).
- Es más apto para los procesos de aprendizaje autónomo (dada por la duración de la batería).
- Comunicación entre dispositivos, mediante la tecnología *BlueTooth*.
- Acceso a Internet desde cualquier dispositivo vía *WiFi*.
- Permiten capturar pensamientos e ideas en el momento que se presentan.
- Posibilitan la interacción espontánea entre alumno-profesor, facilitando de una forma “anónima” y automática la retroalimentación por parte del profesor, la correcta comprensión de determinadas lecciones, temas. . .
- Aprendizaje colaborativo. La tecnología móvil favorece que los alumnos puedan compartir el desarrollo de determinadas actividades con distintos compañeros, creando grupos, compartiendo respuestas, etcétera.
- Los dispositivos móviles facilitan el aprendizaje exploratorio, el aprender sobre el terreno, explorando, experimentando y aplicando a la vez que se aprende la lección.

- Ayuda a los estudiantes a mejorar sus capacidades para leer, escribir y calcular, y a reconocer sus capacidades existentes.
- Puede ser utilizado para incentivar experiencias de aprendizaje independientes o grupales.
- Ayuda a los estudiantes a identificar las áreas donde necesitan ayuda y respaldo.
- Permite a los docentes que envíen recordatorios a sus estudiantes sobre plazos de actividades o tareas a los alumnos así como mensajes de apoyo y estímulo.
- Enriquece, anima y brinda variedad a las lecciones o cursos convencionales.
- Proporciona a menudo actividades intercurriculares.

La utilización del *m-learning* también está asociada a algunos inconvenientes como son [Moreno Guerrero, 2011]:

- Carencia de periféricos para el dispositivo móvil que limitan al usuario a modelos de aprendizaje básico.
- Con excepción de los tablet PC y los portátiles, presentan problemas asociados a la usabilidad, debido a las pequeñas pantallas. Esto conlleva dificultades en la lectura de textos medianos, la cantidad de información visible que es limitada y el desplazamiento continuo por la pantalla para leer toda la información.
- La navegación suele ser limitada.
- Los costos de acceso a la red son altos.
- El trabajo colaborativo es menor.
- La necesidad de conexión inalámbrica que se puede ver afectada por la ubicación física del usuario.
- Dificultades o imposibilidad de instalar y usar determinado software.

1.1.6. *Sharable Content Object Reference Model*

SCORM es un estándar de *e-learning* que permite la creación y empaquetado de contenidos. Está integrado por estándares y especificaciones de empaquetamiento y tiene como objetivo presentar un material didáctico estructurado.

Los materiales educativos bajo SCORM y las plataformas encargadas del uso de estos tienen que cumplir con una serie de normas o políticas para que ambos puedan comunicarse, interactuar y funcionar en conjunto.

Estos paquetes tienen varias características entre las cuales están:

- Diseñado para ser exhibido en un navegador.
- Descrito por metadatos.
- Organizado como un conjunto estructurado de objetos más pequeños.
- Puede ser importado por cualquier plataforma SCORM compatible.
- Creado para ser portable, por lo que es compatible por cualquier sistema operativo.

El contenido de un paquete SCORM está compuesto por diferentes elementos, y la estructura en que se organizan los mismos está definida en el manifiesto, el cual contiene la información necesaria para describir la estructura del paquete, dividido en cuatro secciones según expresa Ramón Verdecia Espinosa en su Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas [Verdecia Espinoza, 2012]:

- *Metadatos:* Describen los objetos educativos, la versión del estándar que se utiliza, los recursos y los objetivos didácticos.
- *Organizaciones:* Forma en que están organizados los contenidos y su desglose en actividades. Las actividades se encuentran conectadas con los recursos a través de su identificador. Dentro de las organizaciones se encuentran las instrucciones de secuenciación y navegación.

- *Recursos:* Describen los recursos externos y locales que utiliza el paquete. Los recursos locales se encontrarán en el fichero comprimido. Para que un recurso pueda comunicarse con una plataforma debe ser un *Objeto de Contenido Intercambiable (SCO)*.

- *SubManifiestos:* Está compuesto por otros manifiestos anidados. En el modelo SCORM un SCO, es la unidad mínima de contenido, es un objeto de aprendizaje que contiene código JavaScript, que le permite comunicarse con un LMS al ser ejecutado.

SCORM también ofrece la posibilidad de crear cursos de formación a partir de recursos educativos. Estos cursos se definen, según describen Marc Vincent, Graziella Duverny, Mehdi Ghezal y Georges Caldeira en su investigación SCORM 2004. Content Aggregation Model, por [Vincent et al., 2006]:

- *Asset:* Recurso básico de la formación visualizados en navegadores Web que pueden ser textos, imágenes, sonidos, páginas HTML, objetos de evaluación, entre otros. 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:* Objeto de Aprendizaje (OA) compuesto por uno o más recursos (Asset). Es la unidad más chica de la que se puede componer un curso SCORM. Por lo general el curso está integrado por un conjunto de SCO organizados de forma jerárquica. El SCO incluye el código de programación necesario para establecer la comunicación con el LMS en tiempo real. Esta comunicación se realiza a través de la Aplication Programming Interface de Android (API), y el SCO debe contener los mecanismos necesarios para encontrarle y comenzar y culminar la sesión de comunicación.

- *Actividad:* Elemento educativo empleado por el estudiante para solicitar a los LMS la puesta en marcha de un Asset o de un SCO. Para el LMS, son los elementos que constituirán el menú de la unidad educativa. Una agrupación de actividades se llama cluster o agregación mientras que el conjunto de las actividades se llama Árbol de Actividades. SCORM no limita la profundidad de esta estructura.

- *La agregación de contenido:* Montaje de una organización y recursos (SCO, Asset) de los componentes de las actividades de la organización. Esta agregación de contenido se va a utilizar para permitir la transferencia de un contenido de un LMS a otro.

- *Organización de los contenidos:* Se compone de uno o más Árboles de Actividades y puede estar asociada con reglas de secuenciación que describen las actividades de la programación educativa. También es posible factorizar reglas para una serie de actividades a la organización de los contenidos.

1.2. XauceMóvil

XauceMóvil brinda la posibilidad de conectarse a diferentes tipos de plataformas educativas ejemplo LMS, repositorios u otras. Le da soporte a ZERA, Moodle, y Repositorio de Objetos de Aprendizaje (RHODA). La aplicación es multiusuario, provee la posibilidad de agregar varios usuarios pertenecientes a una o diferentes plataformas, permitiendo el trabajo *offline* en su dispositivo móvil.

En el caso de los LMS, se le permite a los usuarios acceder a los cursos en los que esté matriculado y visualizar los contenidos de estos, acceder a: las calificaciones obtenidas, las notificaciones y el calendario para ver los principales eventos que son programados por el profesor.

Para lograr todas estas características, hace uso de servicios web a través de REST. La aplicación está diseñada para facilitar la reutilización y mantenibilidad del

código por parte de los desarrolladores ya que hace uso de las buenas prácticas y tecnologías utilizadas en este tipo de aplicación.

1.3. Repositorio de Objetos de Aprendizaje RHODA

RHODA tiene como objetivo gestionar los OA en formato .zip creados en la herramienta de autor Creando Objetos de Aprendizaje (CRODA) bajo el estándar SCORM y siendo arbitrados por Revisores. Es un lugar para el trabajo metodológico colaborativo orientado a elevar la calidad de estos recursos didácticos. [RHODA,]

Para acceder al contenido de estos, el usuario deberá seleccionar la opción que permite visualizarlo y este se mostrará en una nueva pestaña del navegador en la que se podrá ver el árbol de contenido y las opciones que facilitan la navegación.

Este repositorio también brinda la posibilidad de exportar los OA a SCORM 1.2 y SCORM 2004.

1.4. Plataforma Educativa ZERA

La Plataforma Educativa ZERA se basa en la concepción pedagógica cubana llamada “Hiperentornos de aprendizaje”, los cuales constituyen medios de enseñanza que “le permiten al estudiante apropiarse del contenido de manera reflexiva y consciente” [Marrero Zamora, 2011].

ZERA integra “los principales conceptos de los hiperentornos, las mejores prácticas y elementos arquitectónicos de soluciones similares y las principales especificaciones y estándares educativos desarrollados y utilizados a nivel mundial en plataformas de aprendizaje” [Verdecia Espinoza, 2012].

Entre sus funcionalidades se encuentra la exportación e importación de recursos educativos que cumplan con lo que establece el estándar SCORM 2004 3ra Edición.

1.5. Aplicaciones similares

Icodeon:

Icodeon es un visor robusto desarrollado para la plataforma en línea Sakai cuya configuración es flexible y de sencilla adaptación.

Una de sus funcionalidades fundamentales es que registra eventos en las tablas de Sakai y entre sus ventajas se encuentran:

- Está basado en el estándar de la Advanced Distributed Learning (ADL).
- Reproduce todas las versiones de SCORM en castellano.
- Tiene un completo entorno de ejecución diseñado para ser integrado fácilmente a sistemas de gestión de aprendizaje.
- Soporta la secuencia y navegación de los contenidos.
- Excelente soporte técnico.
- Distribuible, pero hay que pagar por él.

Flex SCORM Player 2.0

Flex Scorm Player es un visor para Flex SCORM Builder que otorga la apariencia de un libro virtual e integra varias herramientas que hacen del estudio una experiencia amigable y cómoda. Es muy sencillo, rápido y ágil de usar y cumple con el estándar SCORM 2004.

Content Player Building Block

Es un visor desarrollado para Blackboard que cuenta con certificado ADL. Entre sus características principales están que un instructor puede agregar contenidos que se ajusten a SCORM 1.2 y 2004, Information Management System (IMS), o las normas National Learning Network (NLN) a un curso y que puede ver los IMS, SCORM y NLN dependiendo de los permisos otorgados por el administrador.

Reload Scorm Player

SCORM RELOAD es un visor de escritorio libre, gratuito y multiplataforma muy útil para probar contenidos SCORM sin la necesidad de subirlo a plataformas que lean este estándar. Entre sus principales funcionalides se destacan [RELOAD,]:

- Verifica que el archivo imsmanifest.xml esté correcto.
- Simula la comunicación con una plataforma educativa.
- Almacena los resultados hasta que se reinicien las pruebas.

Vi-Sco

Vi-SCO es un visor de escritorio desarrollado para visualizar los OA de Agrega o que fueron creados en herramientas como eXeLearning bajo el estándar SCORM 1.2.

Para emplear esta aplicación solo hay que ejecutarla y cargar el archivo SCORM; luego de lo cual se genera un menú con toda la secuencia de contenidos que se emplea para la navegación.

UPLAYER_SCORM2004

El visor UPLAYER_SCORM2004 es una aplicación de escritorio desarrollada para interpretar los OA almacenado en RHODA, los cuales están empaquetados bajo el estándar SCORM2004. Esta aplicación tiene como objetivo que los usuarios finales cuenten con una herramienta que les permita visualizar los contenidos de cualquier OA empaquetado bajo éste estándar. La aplicación brinda al usuario la posibilidad de abrir el fichero comprimido (Zip) que constituye el OA, el cual es visualizado estructuralmente por el sistema en forma de árbol, al seleccionar un elemento del mismo y ejecutar clic derecho Mostar Recurso, el visor muestra el navegador web externo por defecto del sistema operativo con el contenido del recurso, además al ejecutar clic en los botones Anterior y Siguiente de la barra de herramientas permite la secuenciación lineal entre los mismos.

SCORM Player

SCORM Player es un visor desarrollado para visualizar los paquetes SCORM importados a ZERA.

Esta aplicación garantiza el establecimiento de la comunicación entre el curso y la Plataforma mediante la API descrita por el estándar y almacena las trazas generadas si el curso lo contiene implementado, guardando el avance del usuario en el contenido.

A partir de este estudio de las aplicaciones similires, se puede concluir que un visor de paquetes SCORM debe descomprimir el archivo .zip y parsear el archivo imsmanifest.xml para reproducir la estructura del paquete que está compuesta por recursos, el contenido, el cual está descrito en un archivo .HTML, y el índice de este último.

Para poder lograr esto, las herramientas de desarrollo de aplicaciones para el Sistema Operativo Android ofrecen una serie componentes, librerías y funcionalidades.

Ejemplo de esto lo constituye el método de lectura y tratamiento de ficheros Extensible Markup Language (XML) llamado Document Object Model (DOM) y el componente webView, el cual permite la inyección de objetos y código JavaScript y HTML.

SCORM reader:

SCORM reader es un lector de cursos y OA SCORM para tablets que utilizan el sistema Operativo Android. Esta aplicación localiza un recurso SCORM, desde la memoria del dispositivo o desde Dropbox, maneja los datos y los muestra en una vista web o webView. 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.

Una de sus funcionalidades fundamentales es que registra eventos en las tablas de Sakai y entre sus ventajas se encuentran:

- Está basado en el estándar de la ADL.
- Reproduce todas las versiones de SCORM en castellano.
- Tiene un completo entorno de ejecución diseñado para ser integrado fácilmente a sistemas de gestión de aprendizaje.

- Soporta la secuencia y navegación de los contenidos.
- Excelente soporte técnico.
- Distribuible, pero hay que pagar por él.

1.6. Tecnologías en el desarrollo de aplicaciones para el sistema operativo Android

Android Software Development Kit

El Android Softaware Development Kit (SDK) ofrece un conjunto de librerías y herramientas necesarias para construir, probar y depurar aplicaciones para Android. También incluye depurador, emulador, documentación, demos y el código de algunas aplicaciones de ejemplo proveyendo al desarrollador con todo lo necesario para desarrollar aplicaciones para dicho sistema operativo.

Este paquete de desarrollo necesita ser importado en algún Entorno de desarrollo integrado (IDE) para que funcione.

Entorno de desarrollo integrado (IDE)

Los IDE, son programas informáticos compuestos por un conjunto de herramientas de programación. Pueden dedicarse a un lenguaje de programación específico o a varios. En cualquiera de los dos casos proveen un marco de trabajo amigable. Están constituidos por un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

Para el sistema operativo Android existen varios IDEs que brindan soporte permitiendo la inclusión del Android SDK dentro de su entorno de desarrollo.

Algunos de ellas son:

- Eclipse: Entorno de desarrollo de código abierto para el desarrollo de aplicaciones multiplataforma que facilita la asociación de varios editores para cada tipo de ficheros y posibilita la edición de estos con programas asociados por el sistema operativo. Además, compila el código en tiempo real optimizando el tiempo de desarrollo y permite la integración de plugins o agregados que enriquecen la plataforma. Un ejemplo de ellos es el Android Development Tools (ADT) Plugin, Plugin de Herramientas de Desarrollo para Android, que tiene como objetivo integrar a este IDE el Android SDK para utilizar sus herramientas.

- NetBeans: Entorno de desarrollo de código abierto, modular y de base estándar (normalizado) escrito en el lenguaje de programación Java y que puede ser usado como una estructura de soporte general (framework) para compilar cualquier tipo de aplicación. En este IDE se pueden desarrollar aplicaciones para Android a través del NBAndroid plugin el cual está disponible para las versiones 7.2 y 7.3 del NetBeans desde su sitio en Internet. Entre sus principales funcionalidades se encuentran:

  - Brinda soporte al núcleo del Android SDK.
  - Permite el uso de emuladores y dispositivos reales para ejecutar los proyectos.
  - Integra el visor LogCat, imprescindible para saber que está pasando con el proyecto que se encuentra ejecutándose.
  - Editores sofisticados para los archivos de tipo XML de Android.

- Motodev Studio for Android: Es un entorno de desarrollo que fue creado por Motorola y está destinado al desarrollo específico de aplicaciones para Android. Está basado en Eclipse y ofrece todas las herramientas necesarias para comenzar a desarrollar aplicaciones a través de un solo instalador constituyendo una opción más sencilla para los usuarios sin experiencia. Sus principales características son [Gómez Peñaranda, 2012]:

- Permite detectar y reparar condiciones inadecuadas, tales como los permisos que faltan y las configuraciones que están en conflicto con las especificaciones del dispositivo.
- Agrega código para resolver tareas frecuentes.
- Permite ver y editar las bases de datos.
- Emulador integrado en el entorno de desarrollo.
- Posee asistentes para la creación de aplicaciones.
- Utiliza herramientas de diagnóstico para detectar problemas de memoria y probar las aplicaciones antes de su distribución.
- Escribe aplicaciones nativas.

Herramienta de diseño

Balsamiq Mockup es una herramienta que permite realizar *wireframes* fácilmente. Un *wireframe* es una representación esquemática de la solución que se llevará adelante, sin entrar en etapas posteriores como el diseño gráfico o la programación. Permite acordar con el cliente aspectos claves de la solución a desarrollar, como la distribución general de los elementos, sus jerarquías y la navegación de los mismos.

Balsamiq Mockups provee representaciones de todos los elementos utilizados para la construcción de aplicaciones, como pantallas de navegadores, títulos, menús, imágenes, videos, etcétera. Haciendo uso de ellos, basta organizarlos en un documento y obtener una primera aproximación de la solución a desarrollar.

Lenguajes de desarrollo

Java

Java es uno de los lenguajes de Progamación Orientada a Objetos (POO) más usados en todo el mundo. En muchos aspectos es similar a C y C++ agregando ciertas

características y eliminando otras que ocasionaban muchos problemas propiciándose así la gran expansión y popularidad que ha obtenido en los últimos años.

Puede utilizarse para desarrollar aplicaciones cliente, cliente/servidor y web. También es el más utilizado en el desarrollo de aplicaciones para Android.

XML

XML es un Lenguaje de Etiquetado Extensible muy simple y flexible pero estricto que se deriva de Lenguaje de Marcado de Anotaciones Generales (SGML) y juega un papel fundamental en el intercambio de datos ya que su función principal es describir esos datos estructurando, almacenando e intercambiando información con el objetivo de facilitar su posterior lectura y uso en diferentes aplicaciones.

Específicamente en Android, además de tratar todo tipo de datos, este lenguaje de programación es el encargado de contener la información, los elementos de cada actividad de la aplicación y el estilo de presentación asignándoles los datos a mostrar a través de claves de identificación.

The ProLearn Query Language (PLQL)

PLQL es un lenguaje de consulta para los repositorios de OA que es utilizado por aplicaciones de *e-learning*. Tiene como objetivo combinar la búsqueda exacta, que se utiliza para la selección de los OA utilizando sus metadatos, y la búsqueda aproximada, que se utiliza para la extracción de los OA mediante de descripciones aproximadas de su contenido. [Campi et al., 2007]

Por lo tanto, una consulta en PLQL contiene tanto cláusulas exactas como cláusulas aproximadas, donde cada cláusula es sintácticamente bien definida. Normalmente, las cláusulas exactas consultan los metadatos de una estructura conocida, mientras que las cláusulas aproximadas realizan la recuperación clasificando mediante el uso de indexación de documentos en base a su contenido.

El resultado de una consulta PLQL es normalmente un conjunto de *Uniform Resource Identifier (URI)* que apunta a los documentos, posiblemente aumentado con metadatos. La recuperación real de cada documento es una tarea que debe ser realizada por el PLQL-cliente. Si la consulta incluye cláusulas aproximadas, entonces el

resultado es un conjunto ordenado. Este ordenamiento es determinado por el servidor PLQL.

1.7. Metodologías de desarrollo de software

Desarrollar software no es una tarea fácil por lo que con el paso del tiempo han surgido varias metodologías con el fin de hacer este proceso más predecible y eficiente haciendo énfasis en la planificación.

De ahí que el éxito de un producto informático dependa en gran medida de la metodología escogida, ya sea tradicional o ágil, ya que unas se adaptan mejor que otras al contexto del proyecto brindando mejores ventajas siempre y cuando los miembros del equipo exploten todo su potencial y aumenten la calidad del producto con los recursos y tiempos establecidos.

Una metodología de software según definen Avison y Fitzgerald es “una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo”. [Avison and Fitzgerald, 2006]

Metodologías tradicionales

Las metodologías tradicionales han demostrado que son efectivas y necesarias, fundamentalmente, en proyectos de gran tamaño respecto a tiempo y recursos pero no son una buena solución en entornos volátiles y/o cuando los requisitos no se conocen con exactitud.

Al aplicar metodologías tradicionales se obliga al cliente a que tome la mayoría de las decisiones al principio, lo cual trae como consecuencia que el cambio de una

decisión tomada puede tener un costo muy alto en términos de ingeniería de software.

Metodologías ágiles

Las metodologías ágiles surgieron como una alternativa a las tradicionales y se basan en "un desarrollo iterativo que centra más en capturar mejor los requisitos cambiantes y la gestión de los riesgos, rompiendo el proyecto en iteraciones de diferente longitud, cada una de ellas generando un producto completo y entregable; e incremental donde un producto se construye bloque a bloque durante todo el ciclo de vida de desarrollo del producto, las iteraciones individuales deben producir alguna característica completamente funcional o mejorada". [Amaya Balaguera, 2013]

Su principal objetivo es reducir el tiempo de desarrollo y tiene como características principales según Blanco y otros autores [Blanco and de autores, 2009]:

- Productos y equipo pequeños.
- La arquitectura es diseñada para los requerimientos actuales.
- Los requerimientos son emergentes y con rápidos cambios.
- El cliente es representativo y se le entrega poder.
- Se enfoca en las personas y los resultados.

A todo esto se puede añadir que la demanda de software para dispositivos móviles impone requisitos muy volátiles y en constante cambio, lo cual requiere tiempos de desarrollo más cortos.

Crystal

Crystal es un conjunto de metodologías para el desarrollo de software cuya característica principal es que se centra en las personas que componen el equipo ya que de ellas depende, en gran medida, el éxito del proyecto.

Es por ello que en Crystal, el equipo de desarrollo juega un papel clave; razón por la cual es necesario que se inviertan esfuerzos en mejorar sus habilidades y destrezas, así como que se tracen políticas de trabajo en equipo.

Dichas políticas deberán estar en función de la cantidad de integrantes del equipo de acuerdo con la clasificación por colores que establece Crystal: Crystal Clear para equipos de 1 a 8 miembros, Crystal Orange para los conformados entre 25 a 50 personas, etcétera.

Scrum

Scrum es una metodología de desarrollo muy simple que se distingue del resto de las ágiles por basarse en la continua adaptación a las circunstancias en las cuales evoluciona el proyecto y no se apega a un plan establecido desde el inicio del proceso de desarrollo.

Esta metodología establece que lo primero que se necesita es tener una visión general del producto, para lo cual, se deben especificar las funcionalidades más críticas a las que hay que darle una mayor prioridad en el proceso de desarrollo y que pueden implementarse en un período de tiempo breve.

Scrum gestiona la evolución de las iteraciones[1] a través de reuniones breves diarias en las cuales se lleva a cabo, por parte del equipo completo, una revisión de todo el trabajo que se realizó el día anterior y el que se ha previsto para el siguiente.

Extreme Programming (XP)

Extreme Programming (XP) es una metodología ágil que considera como un punto clave, en el desarrollo exitoso de software, las relaciones interpersonales por lo que algunas de sus características principales son la promoción del trabajo en equipo, la procupación por el aprendizaje de los desarrolladores y propiciación de un buen clima de trabajo. [Beck, 2000]

Esta metodología es adecuada para los proyectos que tienen requisitos imprecisos y que pueden variar con frecuencia así como para los que tienen un alto riesgo

[1] Cada uno de estos períodos de desarrollo, los cuales finalizan con la producción de un incremento operativo del producto.

técnico ya que se basa en la constante retroalimentación entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los involucrados, la simplicidad en las soluciones implementadas y el enfrentamiento a los cambios.

Está compuesta por 4 elementos fundamentales: historias de usuario, roles, procesos y prácticas.

1.8. Conclusiones parciales

El estudio y análisis del estado del arte permitió concluir que la existencia de visores de paquetes SCORM para el sistema operativo Android es casi nula y el código de las pocas existentes no permite ser reutilizado ni adaptado y no es distribuido. Además, están diseñados para depender de un LMS en específico y necesitan conexión a internet constantemente.

Para el desarrollo de la solución se seleccionaron varias tecnologías y herramientas: como marco de trabajo se de utilizar ADT 22.3.0 con Eclipse 4.2.1 y SDK 19, como lenguajes de programación Java, XML como lenguaje de etiquetado y PLQL como lenguaje de consultas de bases de datos para los servicios de RHODA. También se utiliza SQLite como gestor de bases de datos y como tipo de aplicación a desarrollar se escoge la nativa.

Por las características de flexibilidad y eficiencia se escoge Balsamiq Mockups para el diseño de los prototipos de interfaz de usuario.

Como metodología de desarrollo para la guía del proceso se seleccionó XP porque se adapta a las necesidades de cualquier equipo de desarrollo, estimula la programación en parejas, se dispone de poco tiempo para desarrollar la solución propuesta, los requisitos son muy volátiles y se necesita una comunicación fluida con el cliente y que este tenga mayor participación en el desarrollo del *software* y como tipo de aplicación a desarrollar la nativa.

Capítulo 2

Propuesta de solución

Para continuar el desarrollo de la presente investigación es imprescindible saber las características que debe tener el sistema a desarrollar. Con este propósito, en el presente capítulo se describe la solución propuesta identificando los usuarios que interactúan con la aplicación así como los requisitos funcionales y no funcionales. También siguiendo la metodología XP se recogen los resultados de las primeras fases, Exploración y Planificación, como son las historias de usuario, la estimación del esfuerzo por cada una de estas, el plan de iteraciones y el plan de entregas.

2.1. Descripción de la propuesta solución

Para fomentar el *m-learning* se propone la realización del Visor SCORMView para el sistema operativo Android, que tiene como objetivo proveer a XauceMóvil de un componente que permita ser reutilizado en todos los servidores a los que le da soporte, permitiendo de esta forma interpretar y visualizar los paquetes SCORM. SCORMView permite importar los paquetes bajo el estándar SCORM tanto desde el propio dispositivo móvil como desde los servidores que XauceMóvil le da soporte.

Esta aplicación ofrece también la navegación por cada uno de los contenidos del paquete SCORM a través del índice del contenido y también a través de las opciones

de navegación Anterior y Siguiente. Además, permite realizar búsqueda de OA en RHODA y descargarlos hacia el dispositivo a través del consumo de los servicios web que ofrece este repositorio.

2.1.1. Restricciones de SCORMView

Los paquetes SCORM de RHODA, ZERA y Moodle pueden contener archivos de audio, video y animaciones flash pero el componente WebView de Android, que es el que se utilizará para reproducir los contenidos de los paquetes SCORM, no le da soporte en la actualidad a este formato de archivo.

Esto ocasiona que estos tipos de archivos no puedan ser leídos por SCORMView aunque los paquetes SCORM que los posean sí serán reproducidos.

De igual manera, la aplicación no guardará el avance de los usuarios en los cursos empaquetados bajo el estándar SCORM.

2.1.2. Personal relacionado con el sistema

Usuario que interactúa directamente con la herramienta. Estos usuarios deben poseer conocimientos sobre los dispositivos móviles y Android. Serán los encargados de ejecutar la aplicación según las funcionalidades para las cuales ha sido creada.

2.1.3. Requisitos funcionales del sistema

Los requisitos funcionales constituyen las características del sistema y además describen como se debe comportar el sistema. A continuación se muestran las funcionalidades:

1. RF 1. Iniciar sesión anónima en RHODA.
2. RF 2. Configurar búsqueda en RHODA.
3. RF 3. Buscar OA en RHODA.

4. RF 4. Mostrar resultado de la búsqueda realizada en RHODA.
5. RF 5. Descargar OA desde RHODA.
6. RF 6. Importar OA descargado.
7. RF 7. Importar paquete SCORM desde el dispositivo.
8. RF 8. Listar paquetes SCORM importados.
9. RF 9. Visualizar contenidos del paquete SCORM.
10. RF 10. Mostrar árbol de contenido del paquete SCORM.
11. RF 11. Ocultar árbol de contenido del paquete SCORM.
12. RF 12. Mostrar contenido siguiente del paquete SCORM.
13. RF 13. Mostrar contenido anterior del paquete SCORM.
14. RF 14. Eliminar paquete SCORM de la aplicación.
15. RF 15. Eliminar paquete SCORM del dispositivo.

2.1.4. Requisitos no funcionales del sistema

Los requisitos no funcionales son las restricciones de los servicios ofrecidos por determinado sistema ya que pueden usarse para juzgar la operación del sistema y especificar las propiedades emergentes de este. A continuación se muestran los requisitos no funcionales.

- *Apariencia o interfaz externa:* La información aparecerá correctamente organizada de forma tal que el usuario, pueda encontrar lo que busca rápidamente: El diseño de las interfaces será sencillo, con pocas imágenes y colores. Consistente: siempre que sea posible las acciones similares deben ejecutarse de similar forma. El sistema debe notificar al usuario ante la presencia de un error.

- *Portabilidad:* Se necesita un dispositivo móvil con Android 2.2 o superior.
- *Usabilidad:* El sistema podrá ser utilizado por cualquier usuario que tenga conocimientos básicos en el uso de aplicaciones para dispositivos móviles táctiles. Las acciones a realizar en el visor serán visibles.
- *Restricciones de diseño e implementación:* El lenguaje de desarrollo deberá ser Java y XML en su versión 1.0 o superior. Como marco de trabajo se utilizará ADT 22.3.0 con Eclipse 4.2.1 y Target SDK 21. El motor de bases de datos deberá ser SQlite.

2.2. Exploración

En esta fase, los clientes describen lo que ellos quieren a través de la redacción de sencillas Historia de Usuario (HU). El equipo de desarrollo se familiariza con las herramientas y tecnologías que se utilizarán para llevar a cabo el proyecto. Se define el alcance general del proyecto y se estima su plazo total. Esta fase debe durar un par de semanas.

2.2.1. Historias de usuario (HU)

Las HU son la técnica utilizada en la metodología XP para especificar los requisitos del software. Son tarjetas de papel en las cuales el cliente, en un lenguaje no técnico, describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

Cada HU es lo suficientemente comprensible y delimitada para que pueda ser implementada entre 1 y 3 semanas asignándoles tareas de programación con un número de horas de desarrollo. Son la base para las pruebas funcionales ya que en la fase de pruebas se utilizan para verificar si la propuesta desarrollada cumple con lo que especifica la HU. A continuación se muestran las HU definidas por el cliente.

#	Historia de usuario	Prioridad para el negocio
1	Importar los paquetes SCORM	Alta
2	Conectar a RHODA para descargar los OA	Alta
3	Visualizar paquete SCORM	Alta
4	Navegar por los contenidos del paquete SCORM	Media
5	Eliminar paquete SCORM	Media

Tabla 2.1: HU identificadas por el cliente

Para definir historias de usuario se utilizan plantillas que contienen información sobre estas como: número de historia, prioridad en negocio, estimación técnica, descripción, observación, iteración asignada, programador responsable, nombre de historia, usuario y riesgo en desarrollo. A continuación se muestra un ejemplo de una plantilla de HU.

Historia de usuario	
Número:	**Usuario:**
Nombre de historia:	
Prioridad en negocio:	**Riesgo en desarrollo:**
Programador responsable:	
Descripción:	
Observaciones:	

Tabla 2.2: Plantilla de HU

Número: Es el número que identifica a la HU. Número sucesivo a partir de uno.

Usuario: El usuario del sistema que utiliza o protagoniza la historia.

Nombre de historia: Contiene el nombre de la HU.

Prioridad en el negocio: Define la relevancia e impacto de la HU para el negocio de acuerdo a las necesidades del usuario. Si es importante debe ser implementada lo antes posible, esta se clasifica en:

- Alta: Se le otorga a las historias de usuarios que resultan funcionalidades fundamentales en el desarrollo del sistema, las que el cliente define como principales para el control integral del sistema.
- Media: Se le otorga a las historias de usuarios que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.
- Baja: Se le otorga a las historias de usuarios que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

Riesgo de desarrollo: Riesgo que representa para el desarrollo la historia de usuario. En dependencia del riesgo que represente se le asigna un valor que puede ser:

- Alto: Cuando en la implementación de las historias de usuarios se considera la posible existencia de errores que lleven a la inoperatividad del código.
- Medio: Cuando pueden aparecer errores en la implementación de las historias de usuarios que puedan retrasar la entrega de la versión.
- Bajo: Cuando pueden aparecer errores que serán tratados con relativa facilidad, sin que traigan perjuicios para el desarrollo del proyecto. [Pérez Rodríguez and González Martínez, 2013]

Puntos Estimados: Atributo que contiene la estimación hecha por el equipo de desarrollo del tiempo de duración de la HU. Cuando el valor es 1 equivale a una semana ideal de trabajo. En la metodología XP está definida una semana ideal como cinco

días hábiles. Cuando el valor de dicho atributo es de 0.5 equivale a dos días y medio de trabajo, lo que se traduce en veinte horas.

Iteración asignada: Iteración en la que se implementará la HU.

Programador responsable: Programador responsable de realiza la historia de usuario.

Descripción: Explica en qué consiste la HU, teniendo en cuenta las acciones realizadas por el usuario y la respuesta brindada por el sistema.

Observaciones: Alguna especificación con la que deba cumplir la HU.

En esta fase de exploración como anteriormente se expuso se identificaron 5 historias de usuarios (Anexo A). A continuación se muestra una de ellas.

Historia de usuario	
Número: 1	**Usuario:** Usuario
Nombre de historia: Importar los paquetes SCORM	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Bajo
Programador responsable: Sandy Nuñez Padrón y Dianelys Pérez González	
Descripción: Posibilita importar un paquete SCORM, este es validado y es agregado a la base de datos como un recurso más del repositorio.	
Observaciones: El paquete SCORM debe cumplir que: Sea un archivo .zip y que exista en su raíz el fichero imsmanifest.xml, se debe comprobar que esté correctamente formado, que el schema version corresponda con 1.2 o SCORM 2004 y que contenga las únicas etiquetas obligatorias: *manifest*, *schema organizations*, al menos una etiqueta *organization* con un *item* y un *title*, y *resources*.	

Tabla 2.3: HU Importar paquetes SCORM

2.2.2. Estimación del esfuerzo por historia de usuario

Existen varios grupos de técnicas para realizar la estimación del esfuerzo, entre ellos está el grupo de las técnicas especializadas el cual comprende la estimación para desarrollo ágil.

Esta técnica se recomienda para proyectos de corta disponibilidad de tiempo y gran probabilidad de cambios. En ella, se utiliza un enfoque de descomposición que utiliza los propios artefactos que define la metodología ágil con la que se trabaja, en este caso se utilizan las historias de usuario.

Luego, por cada historia de usuario se definen las tareas que permitirán su construcción y se estima para cada una de esas tareas. Al final la estimación del proyecto será el total de la estimación para la suma de cada tarea.

HU	Esfuerzo estimado
Importar paquete SCORM	1
Conectar a RHODA para descargar los OA	1
Visualizar paquete SCORM	2
Navegar por los contenidos del paquete SCORM	1
Eliminar paquete SCORM	0.5

Tabla 2.4: Estimación del esfuerzo por HU

2.2.3. Plan de iteraciones

Los equipos XP realizan iteraciones con un tiempo de duración de aproximadamente 3 semanas. Una iteración es la práctica en donde el equipo establece el rumbo a seguir en estas semanas. El cliente durante la planificación de la iteración selecciona las HU definidas que serán implementadas. Estas HU son divididas en tareas de programación entre 1 y 3 días. También por cada HU se planifican las pruebas de

aceptación que se realizan al final de cada iteración y en las iteraciones siguientes para evitar que las HU ya implementadas sean afectadas por ciclos subsiguientes.

Iteración 1: En esta iteración se realizaran la primer y la segunda HU ya que tienen alta importancia para el negocio.

1. Importar paquete SCORM.
2. Conectar a RHODA para descargar los OA.

Iteración 2: En esta iteración se realizan la tercera HU ya que tienen importancia alta para el negocio.

1. Visualizar paquetes SCORM.

Iteración 3: En esta iteración se realizan la cuarta y la quinta HU ya que tienen importancia media para el negocio.

1. Navegar por los contenidos del paquete SCORM.
2. Eliminar paquete SCORM.

Plan de duración de iteraciones

Ponderaciones	Orden de las HU a implementar	Tiempo de trabajo
Iteración 1	Importar paquete SCORM Conectar a RHODA para descargar los OA	2
Iteración 2	Visualizar los paquetes SCORM	2
Iteración 3	Navegar por los contenidos del paquete SCORM Eliminar paquete SCORM	1.5

Tabla 2.5: Plan de duración de las iteraciones

2.2.4. Plan de entregas

Un Plan de entregas establece las HU que se realizarán para cada versión del proyecto y las fechas en que estas serán entregadas. Para realizar este plan se tiene en cuenta varios aspectos como: los objetivos que se deben cumplir y su prioridad para el cliente, las estimaciones de tiempo y costo que realizan los desarrolladores y el número de personas que trabajarán en la implementación de estos objetivos.

HU	Final de la primera iteración	Final de la segunda iteración	Final de la tercera iteración
Importar paquete SCORM	1.0	Finalizado	Finalizado
Conectarse a RHODA para descargar los OA	1.0	Finalizado	Finalizado
Visualizar los paquetes SCORM	-	1.1	Finalizado
Navegar por los contenidos del paquete SCORM	-	-	2.0
Eliminar paquete SCORM	-	-	2.0

Tabla 2.6: Plan de entregas

2.3. Diseño

En la fase de diseño la metodología XP indica que se deben realizar diseños simples y sencillos ya que tardaran menos tiempo en implementarse. Los aspectos más importantes de esta metodología con respectos al diseño son [Joskowicz, 2008]:

Simplicidad: Un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione. Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la

iteración en la que se esté trabajando.

Soluciones "spike": Cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de usuario, pueden utilizarse pequeños programas de prueba (llamados "spike" 1), para explorar diferentes soluciones. Estos programas son únicamente para probar o evaluar una solución, y suelen ser desechados luego de su evaluación.

Recodificación: La recodificación ("refactoring") consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de hacerlo más simple, conciso y/o entendible. Muchas veces, al terminar de escribir un código de programa, pensamos que, si lo comenzáramos de nuevo, lo hubiéramos hecho en forma diferente, más clara y eficientemente. Sin embargo, como ya está pronto y "funciona", rara vez es reescrito. Las metodologías de XP sugieren recodificar cada vez que sea necesario. Si bien, puede parecer una pérdida de tiempo innecesaria en el plazo inmediato, los resultados de ésta práctica tienen sus frutos en las siguientes iteraciones, cuando sea necesario ampliar o cambiar la funcionalidad. La filosofía que se persigue es, como ya se mencionó, tratar de mantener el código más simple posible que implemente la funcionalidad deseada.

2.3.1. Patrones de diseño de interfaz de usuario

En Android, se conocen como patrones de diseño las soluciones generales a un problema recurrente. [McKenzie, 2011] A continuación se presentan los patrones de diseño de Android que se seleccionaron para la presente investigación.

ActionBar

La barra de acciones es uno de los patrones de diseño más importantes de Android y que además lo diferencia. Su funcionamiento es muy similar a la de un banner de los sitios web, con el logo o el título generalmente a la izquierda y los elementos de navegación a la derecha. El diseño de la barra de acciones es flexible y permite que

se cierren menús y se amplíen las cajas de búsqueda. Generalmente es usada como una característica global en lugar de una característica contextual. [McKenzie, 2011]

Quick Action

Básicamente, es un menú contextual que no cubre la información sobre la que se está actuando. La implementación de este patrón probablemente podría hacer la aplicación más interactiva e interesante. [Phat, 2013]

2.3.2. Prototipos de interfaz de usuario

A través de un prototipo de interfaz de usuario se obtiene un modelo de la aplicación que se va desarrollar, el cual ofrece la oportunidad de entender mejor el problema, evaluar los requerimientos, así como a tener en cuentan varias posibilidades de diseño. Estas propuestas pueden cambiar en algún momento, según los criterios del cliente.

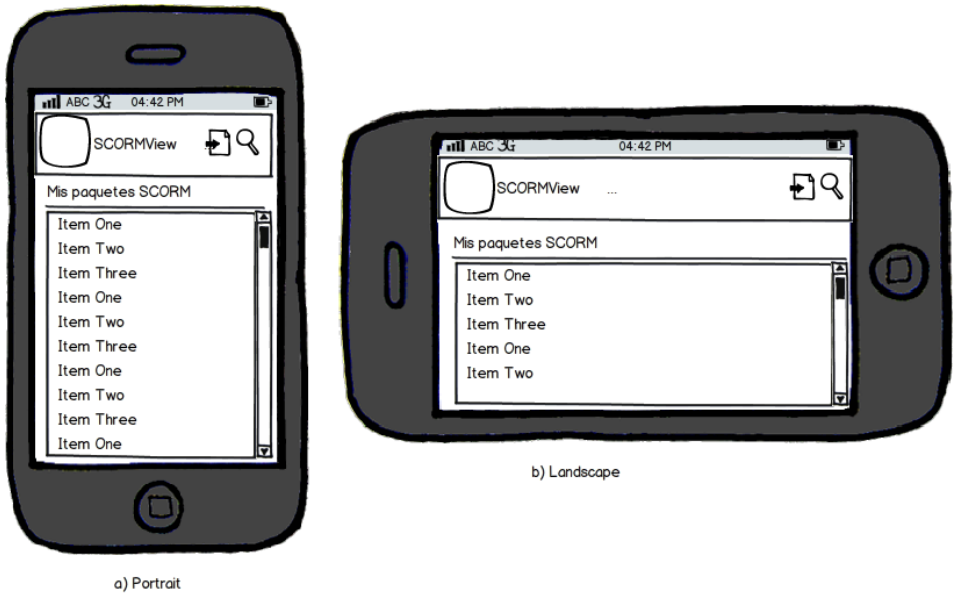

Figura 2.1: *Prototipo de interfaz de usuario de la actividad principal.*

En la fase de diseño se realizaron 5 prototipos de interfaz de usuario (Anexo B). Un ejemplo de ellos es el que se muestra en la figura 2.1.

2.3.3. Tarjetas *Class, Responsabilities and Collaboration* (CRC)

Las tarjetas *Class, Responsabilities and Collaboration (CRC)* (Anexo C) permiten la contribución de todo el equipo para el diseño del proyecto. Estas a su vez proporcionan a los desarrolladores reducir el modo de pensar de la programación procedural y apreciar el desarrollo orientado a objetos.

Las tarjetas CRC representan objetos y se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. El nombre de la clase es la clase a la cual pertenece el objeto. Las responsabilidades de una clase son las cosas que realiza o los objetivos que debe cumplir el objeto. Los colaboradores de una clase son otras clases con las que interactúa para cumplir con sus responsabilidades.

Archivo	
Responsabilidad	**Colaboraciones**
getNombre	
getDireccion	
setNombre	
setDireccion	

Tabla 2.7: Tarjeta CRC de la clase Archivo.

2.3.4. Patrón Arquitectónico

Para el desarrollo de aplicaciones para dispositivos móviles sobre la plataforma Android se propone el uso del patrón arquitectónico N-Capas, muy recomendable ya que

los entornos de ejecución de dichas aplicaciones estarán distribuidos en diferentes elementos o nodos tanto lógicos como físicos. Aquí aparecen dos conceptos importantes:

- Capas (*Layers*) que se encargan de la distribución lógica de los componentes.
- Niveles (*Tiers*) que se refieren a la colocación física de los recursos.

La característica principal de este patrón es que cada capa oculta las capas inferiores de las siguientes superiores a esta. Algunas de las ventajas de utilizar este patrón son [Peláez, 2009]:

- Mejoras en las posibilidades de mantenimiento, debido a que cada capa es independiente de la otra los cambios o actualizaciones pueden ser realizados sin afectar la aplicación como un todo.
- Escalabilidad, ya que las capas están basadas en diferentes máquinas, el escalamiento de la aplicación hacia afuera es razonablemente sencillo.
- Flexibilidad, como cada capa puede ser manejada y escalada de forma independiente, la flexibilidad se incrementa.
- Disponibilidad, las aplicaciones pueden aprovechar la arquitectura modular de los sistemas habilitados usando componentes que escalan fácilmente lo que incrementa la disponibilidad.

La descomposición en capas que se propone es [Larramendi Ferraz, 2012]:

1. **Capa de presentación:** es la única que interactúa directamente con el usuario presentándole las funcionalidades del sistema, permitiendo el intercambio de información entre ambos. Contiene una interfaz gráfica que posibilita capturar los datos insertados por los clientes, además de mostrarles los resultados de sus peticiones y los estados de la aplicación. Esta capa solo interactúa con la capa de negocio, que es su inferior inmediata.

2. **Capa de negocio:** también conocida como lógica de negocio, es la que recibe las peticiones de la capa de presentación y le envía las respuestas tras el proceso. Aquí se realiza la mayor parte del procesamiento de la información del dominio de la aplicación, se ejecutan cálculos sobre los datos de entrada o almacenados, se validan los contenidos provenientes de la capa superior y se ejecutan los algoritmos específicos del programa. Contiene un componente propio del sistema operativo Android cuya función principal es manejar el consumo de servicios web. Esta capa interactúa con la capa de presentación, para recibir sus solicitudes y presentarle los resultados; y con la capa de datos, para solicitar al gestor de base de datos almacenar u obtener datos de él.

3. **Capa de acceso a datos**: es la encargada de intercambiar con otros sistemas, sólo mediante ella se puede acceder a los recursos obtenidos de terceras aplicaciones. Contiene uno o más gestores de base de datos para lenguaje SQL. Esta capa interactúa con la capa de negocio, recibe sus solicitudes de almacenamiento o recuperación de información, y envía la respuesta a las peticiones.

A continuación se describen los elementos de cada capa:

1. **Capa de presentación:** en esta capa se encuentra el Webkit que visualiza el contenido del paquete SCORM a partir del código HTML, CSS y JavaScript de cada vista del contenido.

2. **Capa de negocio:** en esta capa se representan todos los componentes lógicos del flujo de trabajo, o sea las clases Javas que se encargan de ejecutar las tareas propias de la aplicación. También se manejan los datos introducidos por el usuario en las vistas que lo requieran gracias al APIwrapper y lanzan llamadas al adaptador API correspondiente para comunicar los estados del archivo SCO y datos relacionados con él.

3. **Capa de acceso a datos:** en esta capa, la aplicación localiza un paquete SCORM desde la memoria externa del dispositivo o desde RHODA y se interpre-

ta el archivo imsmanifest.xml que contiene los datos necesarios para reproducir el paquete.

2.3.5. Patrones de diseño

Un patrón de diseño es una descripción de un problema y su solución que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas. Muchos patrones ofrecen orientación sobre cómo asignar las responsabilidades a los objetos ante determinada categoría de problemas. [Larman, 1999] A continuación se detallan los patrones utilizados:

Adaptador (Adapter)

Adapter también conocido como *Wrapper* convierte la interfaz de una clase en otra interfaz esperada por los clientes. Permite que clases con interfaces incompatibles puedan comunicarse. Se utiliza para crear la lista de paquetes SCORM importados que se adicionan por el usuario y para visualizar el árbol de contenidos de estos paquetes en las clases SCORMView.java y ViewOA.java, respectivamente.

Memento

Memento es un patrón de comportamiento. Permite almacenar el estado interno de un objeto preservando el principio del encapsulamiento, es decir guarda parte o todo el estado interno de un objeto, para que este objeto pueda ser restaurado más tarde al estado guardado por Memento. [Caituiro Monge, 2011]

Se emplea para almacenar el contexto de la clase ViewOA.java junto con el estado de las variables en determinado momento, y que el usuario pueda volver a ese estado exacto cuando rote la pantalla del dispositivo.

Bajo acoplamiento

Consiste en el diseño de clases mucho más independientes, que en caso de ocurrir alguna modificación se reduzca el impacto de los cambios, y también mas reutilizables, que acrecientan la oportunidad de una mayor productividad. Su uso se evidencia en las clases PaqueteSCORM.java, Archivo.java y PaquetesScormSQLiteHelper.java porque estas no dependen de ninguna otra y en SCORMParseDOM.java porque tiene la menor dependencia de otras clases.

Alta cohesión

Existe alta cohesión funcional cuando los elementos de una clase colaboran para producir algún comportamiento bien definido, es decir cada elemento debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos, y auto-identificable. Este patrón se emplea en las clases PaqueteSCORM.java, SCORMParseDOM.java, Archivo.java y PaquetesScormSQLiteHelper.java

2.4. Conclusiones parciales

En el presente capítulo se describió la propuesta de solución para el desarrollo del intérprete y el visor de contenidos para SCORMView.

Las fases de Planificación y Diseño posibilitaron la confección de los artefactos que la metodología de desarrollo propone para estas etapas y que sirven de base para las próximas.

A partir de un análisis llevado a cabo por los miembros del equipo de desarrollo, se realizó un diseño simple y sin ambigüedades de las clases del sistema, el cual estará en constante perfeccionamiento durante la Fase de Implementación.

Capítulo 3

Implementación y pruebas

En el presente capítulo se exponen las fases de Implementación y Prueba propuesta por la metodología de desarrollo de software seleccionada, describiendo las tareas de ingeniería, las cuales facilitarán el desarrollo de la propuesta de solución. Además se realizarán las pruebas a la aplicación, comprobando la correcta ejecución de las funcionalidades implementadas, documentando los resultados arrojados.

3.1. Implementación

En esta fase la asistencia del cliente tiene gran importancia ya que al desarrollar cada HU el cliente debe precisar claramente lo que ésta hará y también tendrá que permanecer cuando se realicen las pruebas que comprueben que la historia implementada cumple la funcionalidad especificada.

XP se inclina por la programación en pareja ya que proporciona un código más eficiente y con una gran calidad. Posibilita la transferencia de conocimientos de programación entre los miembros del equipo y varias personas pueden entender las diferentes partes del sistema.

Para ayudar a la implementación de las HU estas se descomponen en tareas de programación descritas con lenguaje técnico y asignadas al responsable de su imple-

mentación. Como parte de la planificación realizada en el capítulo anterior se llevaron a cabo tres iteraciones.

3.1.1. Manifiesto de la aplicación Android

La aplicación necesita que el usuario apruebe varios permisos para poder instalarse 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 RHODA así como la búsqueda y descarga de OA desde ese repositorio.
- *android.permission.ACCESS_NETWORK_STATE:* Permite conocer el estado de la conexión a internet en el dispositivo, lo cual es necesario para comprobar si es o no posible lanzar tareas que requieran de ella y notificar al usuario en caso negativo.
- *android.permission.WRITE_EXTERNAL_STORAGE:* Permite el acceso, lectura y escritura en la memoria externa del dispositivo si está presente. La aplicación utiliza estas características para descargar, descomprimir y eliminar los archivos .zip que contienen los recursos.
- *android.permission.READ_EXTERNAL_STORAGE:* Permite el acceso y la lectura de la memoria externa del dispositivo si está presente. La aplicación utiliza estas características para importar los archivos, .zip y comprobar que los paquetes importados existan físicamente en el dispositivo.

Con el objetivo de que la aplicación sea funcional, también es necesario que el manifiesto de la aplicación contenga:

- *android:uiOptions="splitActionBarWhenNarrow":* Se utilizó en RhodaActivity y ViewOA para que la ActionBar de la aplicación aparezca dividida en las *activities* a las que se les agregue esta propiedad, ubicando el logo y el nombre de

la aplicación en la parte superior de la pantalla y los botones de acción en la parte inferior.

- *meta-data android.support.UI_OPTIONS:* Se utilizó en las activities RhodaActivity y ViewOA para lograr la compatibilidad de la ActionBar dividida con las versiones de Android inferiores a la API 14.

- *meta-data android.support.PARENT_ACTIVITY:* Se utilizó en todas las *activities* para definir la *activity* padre de la actual y ubicar un enlace a la primera en el ActionBar de la segunda.

3.1.2. Compatibilidad entre las *Aplication Programming Interface de Android* (API)

Uno de los problemas a los que habitualmente se enfrentan los programadores de aplicaciones para Android es que la aplicación se vea de la misma forma en todas las API de este sistema operativo y ofrezca los servicios con la misma calidad y nivel de funcionalidad.

Para solucionar esto, el paquete *Android Support Library* ofrece un conjunto de bibliotecas de código que proporcionan versiones compatibles con versiones anteriores de la API de marco para Android, así como características que sólo están disponibles a través de las API de la biblioteca. Cada *Support Library* es *backward*-compatible a un nivel específico de la API de Android. Este diseño significa que las aplicaciones pueden utilizar las funciones de las bibliotecas y seguir siendo compatible con los dispositivos que ejecutan viejos sabores de Android. [Xamarin, 2014]

Incluyendo las Support Library en un proyecto Android se considera una buena práctica para los desarrolladores de aplicaciones, dependiendo de la variedad de versiones de la plataforma en las cuales está focalizada y las API que utiliza. Usando las funciones de las bibliotecas se puede mejorar el aspecto de la aplicación, aumentar su rendimiento y ampliar su alcance a más usuarios.

Para lograr esto en el caso de SCORMView se hace necesario utilizar:

- *appcompat_v7:* Biblioteca de Android que ofrece a los desarrolladores la oportunidad de añadir la Barra de acciones a las aplicaciones dirigidas a los dispositivos con API 7 o superior que no sean compatibles con dicha barra. También permite una acción de intercambio estandarizado como el intercambio de texto o imágenes desde el interior de la barra de acción. [Montemagno, 2015]

- *android.support.v4.app:* Clases de apoyo android.app para ayudar en el desarrollo de aplicaciones para el nivel de API de Android 4 o posterior. Las características principales aquí son versiones compatibles con versiones anteriores de FragmentManager y LoaderManager. [Developers, 2015]

3.1.3. Iteración 1

En esta iteración se realizaran las HU que tienen alta prioridad para el negocio con sus respectivas tareas de ingeniería (Anexo D).

HU	Tiempo estimado	Tiempo real	Tareas de ingeniería
Importar paquete SCORM	1	1.2	**T1:** Comprobar que el archivo es un paquete SCORM. **T2:** Registrar en la base de datos. **T3:** Realizar pruebas unitarias.

Conectarse a RHODA para descargar los paquetes SCORM	1	1.4	**T4:** Conectarse a RHODA. **T5:** Buscar el OA en RHODA. **T6:** Descargar OA. **T7:** Realizar pruebas unitarias.

Tabla 3.1: Plan de tareas de ingeniería de la iteración 1.

A continuación se muestra una de las tareas de ingeniería realizadas para las HU de esta iteración.

Tarea de ingeniería	
Número de tarea: 1	**Número de HU:** 1
Nombre de tarea: Comprobar que el archivo es un paquete SCORM	
Tipo de tarea: configuración - desarrollo	**Puntos de estimación:** 0.6
Fecha inicio: 23/03/2015	**Fecha fin:** 26/03/2015
Programador responsable: Sandy Nuñez Padrón y Dianelys Pérez González	
Descripción: La aplicación debe comprobar que el archivo .zip que se va a importar es un paquete SCORM.	

Tabla 3.2: Tarea de ingeniería Comprobar que el archivo es un paquete SCORM

Comprobar que el archivo a importar sea un paquete SCORM

Antes de importar un archivo a la aplicación y registrarlo en la base de datos, se hace imprescindible comprobar que sea un paquete SCORM. Para ello se descomprime el archivo en un hilo secundario y luego se comprueba que exista el archivo imsmanifest.xml. A continuación se parsea para verificar que su estructura sea correcta, la

cual debe ser como se muestra en el Anexo H.

Para hacer esto último, se utiliza el DOM, uno de los métodos disponibles en Android para leer y tratar fiheros XML desde las aplicaciones. Con DOM, el documento XML se lee completamente antes de poder realizar ninguna acción en función de su contenido. Esto es posible gracias a que, como resultado de la lectura del documento, el parser DOM devuelve todo su contenido en forma de una estructura de tipo árbol, donde los distintos elementos del XML se representan en forma de nodos y su jerarquía padre-hijo se establece mediante relaciones entre dichos nodos. [Gómez Oliver, 2013]

Una vez obtenido el árbol que contiene la estructura del archivo imsmanifest.xml, se verifica que la etiqueta del nodo raíz sea *manifest* y que sus nodos hijos sean *metadata*, *organizations* y *resources*.

Luego se comprueba que dentro del nodo *metadata* se encuentren *shema* como padre de *ADL SCORM*, *shemaversion* con su hijo *1.2* o *2004 3rd Edition* e *imsmd:lom* con cada uno de sus nodos hijos. También se verifica que *organizations* sea el padre de *organization* y que este tenga como hijos a *title* y varios *item*, comprobando que este último tenga un solo nodo hijo que deberá ser *title*. En el caso de *resources* se verifica que no sea una hoja y que todos sus nodos hijos sean *resource*. Este nodo deberá ser padre solamente de nodos hoja *file*, verificándose que exista al menos una de estas hojas.

3.1.4. Iteración 2

En esta iteración se realizan las HU que tienen importancia alta para el negocio con sus respectivas tareas de ingeniería (Anexo E).

HU	Tiempo estimado	Tiempo real	Tareas de ingeniería

Visualizar paquete SCORM	2	2.2	**T8:** Descomprimir archivo .zip. **T9:** Recuperar el contenido. **T10:** Parsear el imsmanifest.xml. **T11:** Reconstruir el contenido del paquete SCORM. **T12:** Realizar pruebas unitarias.

Tabla 3.3: Plan de tareas de ingeniería de la iteración 2.

Visualizar contenido del paquete SCORM

Para visualizar el contenido de un paquete SCORM se descomprime el archivo .zip y se localiza el imsmanifest.xml, el cual se parsea para recuperar los datos necesarios utilizando el método DOM. En este caso, se recorre el árbol con la estructura de dicho archivo hasta recuperar el nombre del recurso educativo, el cual sería el valor del nodo hoja que es hijo de *imsmd:langstring* cuyo padre es *imsmd:title*.

Se visitan, además, los nodos hojas que son hijos de *title* cuyo padre es *organization* para obtener sus valores y luego reconstruir el árbol de contenidos. También se recupera el contenido del recurso educativo; esto se logra al visitar los nodos *resource* y obtener la hoja file cuyo valor sea un archivo .html y la referencia al título de contenido al que está asociado.

Luego se construye un DrawerLayout en el que se le muestra el árbol de contenidos del recurso educativo al usuario, el cual podrá ocultarlo y visualizarlo cuando desee, y se lanza en un webView el archivo .html que corresponde con el contenido inicial o con el que seleccione el usuario.

3.1.5. Iteración 3

En esta iteración se realizan las HU que tienen importancia media para el negocio con sus respectivas tareas de ingeniería (Anexo F).

HU	Tiempo estimado	Tiempo real	Tareas de ingeniería
Navegar por los contenidos	1	1.6	**T13:** Recuperar el árbol de contenidos. **T14:** Reconstruir el árbol de contenidos. **T15:** Mostrar contenido Siguiente del paquete SCORM. **T16:** Mostrar contenido Anterior del paquete SCORM. **T17:** Realizar pruebas unitarias
Eliminar paquete SCORM	1	0.4	**T18:** Eliminar físicamente. **T19:** Eliminar de la base de datos. **T20:** Realizar pruebas unitarias.

Tabla 3.4: Plan de tareas de ingeniería de la iteración 3.

3.2. Pruebas

Las pruebas son las que muestran si la aplicación se encuentra funcionando en perfectas condiciones y ofrecen la oportunidad de conocer si el sistema desarrollado hasta el momento se corresponde con lo que el usuario tenía en mente. Las prue-

bas constituyen uno de los procesos más importantes de XP y es una actividad que se realiza de forma continua a lo largo del desarrollo de el software. Con estas se disminuye el número de errores no detectados durante la fase de implementación y el tiempo entre la aparición de un error y su detección. Son las encargadas de aumentar la seguridad y de evitar efectos colaterales no deseados a la hora de realizar modificaciones en la aplicación.

La metodología XP define dos tipos de pruebas: pruebas unitarias, realizadas al código y diseñadas por los programadores, y pruebas de aceptación, las cuales están diseñadas por el cliente y garantizan que se cumplan todas las funcionalidades del sistema.

3.2.1. Pruebas unitarias

Las pruebas unitarias o también llamadas pruebas técnicas se aplican a los métodos de una clase antes de ser liberadas o publicadas. Son elaboradas por los desarrolladores antes que los métodos, proporcionándole máxima precisión sobre lo que va a programar, así como dominar cada uno de los casos de prueba que deberá pasar, lo que optimizará su trabajo y su código será de mejor calidad.

El uso de pruebas unitarias posibilita la liberación continua de versiones por consiguiente al implementar un método o funcionalidad nueva, solo es efectuar los casos de pruebas definidos y ver que funciona satisfactoriamente para integrarlo al código ya existente sin alterar el funcionamiento actual de mismo. En caso de que existan errores, estos deben ser corregidos inmediatamente y se realizan nuevas pruebas para verificar que el error haya sido resuelto.

Dichas pruebas se desarrollan utilizando la extensión Android JUnit que provee el SDK de Android. Ella permite probar componentes específicos mediante clases de casos de pruebas, las cuales proporcionan métodos auxiliares para la creación de objetos de imitación y métodos que ayudan a controlar el ciclo de vida de un componente (Figura 3.1).

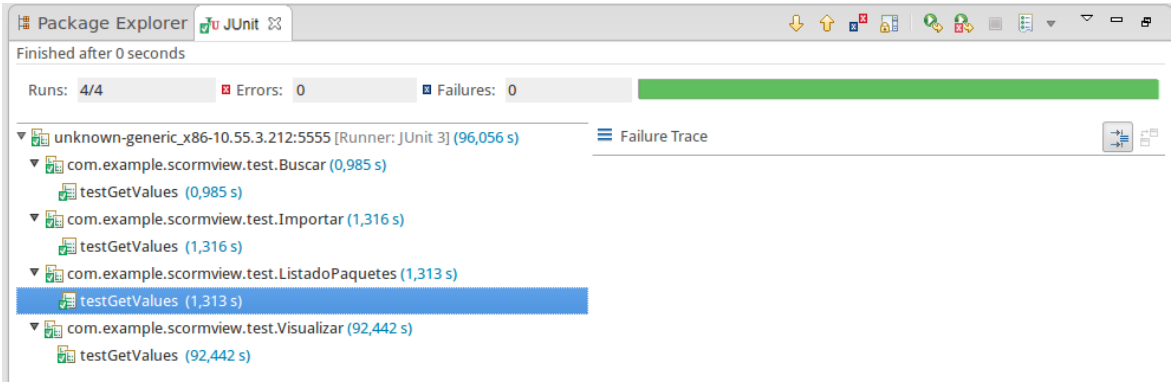

Figura 3.1: *Resultados de las pruebas unitarias.*

3.2.2. Pruebas de aceptación

Las pruebas de aceptación (Anexo G) o también llamadas pruebas funcionales son creadas por el cliente para comprobar que las HU cumplen con lo requerido. El cliente debe especificar diversos escenarios para cada HU de tal forma que puedan ser evaluadas. Estas pruebas se llevan a cabo al final de cada iteración puesto que tienen una gran importancia para el éxito de la misma marcando el camino a seguir, indicándole al equipo de desarrollo hacia donde tiene que ir y en qué puntos o funcionalidades debe poner el mayor esfuerzo y atención. En caso de que existan fallas, el cliente debe especificar un orden de prioridad con el cual se le dará solución a los errores detectados.

Caso de prueba de aceptación	
Código: HU1_P1	**Número de HU:** 1
Nombre: Importar paquete SCORM	
Descripción: Prueba para la funcionalidad que permite importar un paquete SCORM.	
Condiciones de ejecución: El archivo .zip que se desea importar debe ser un paquete SCORM.	
Pasos de ejecución: El usuario selecciona en el menú la opción “Importar desde archivo”.	

La aplicación muestra los datos que el usuario contiene en su tarjeta SD. El usuario selecciona el paquete SCORM que desea importar.
Resultado esperado: Paquete SCORM sea importado a SCORMView.
Evaluación de la prueba: Satisfactoria

Tabla 3.5: Caso de prueba de aceptación Importar paquete SCORM

Resultados de las pruebas de aceptación: En la etapa de pruebas se realizaron 7 pruebas de aceptación las cuales arrojaron un número de 10 no conformidades (Figura 3.2) entre las tres iteraciones realizadas: en la primera 4 significativas y 1 no significativa, en la segunda 2 significativas y 1 no significativa y en la tercera 1 significativa y 1 no significativa.

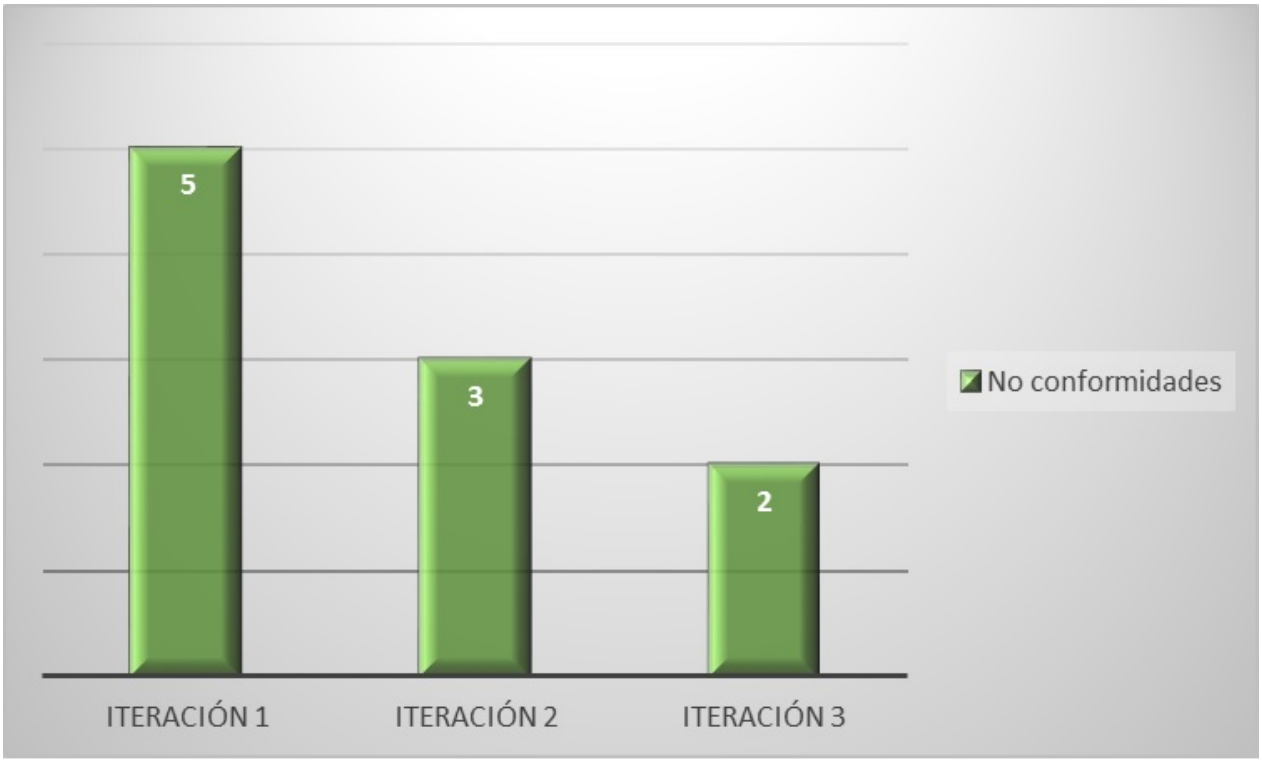

Figura 3.2: *Resultados de las pruebas de aceptación por iteración.*

Las no conformidades no significativas detectadas se centraron más en errores ortográficos: omisiones de tildes y cambios de mayúsculas por minúsculas. Las significativas se centraron en errores de validación y de diseño resultados.

3.3. Conclusiones parciales

Se desarrollaron las funcionalidades requeridas por el cliente a través de la utilización de las herramientas y tecnologías seleccionadas. Como parte de la planificación realizada se llevaron a cabo tres iteraciones, en las cuales se implementaron la tareas de ingeniería correspondientes a las HU. También se realizaron pruebas unitarias y de aceptación arrojando varias no conformidades que fueron resueltas satisfactoriamente.

Conclusiones

La investigación desarrollada y los resultados obtenidos permiten a los autores plantear las siguientes conclusiones:

- Las herramientas existentes para visualizar paquetes SCORM en el sistema operativo Android están hechas a la medida, no permiten el trabajo *off-line*, no son distribuidas y su código no pude ser adaptado ni reutilizado.
- El desarrollo del visor SCORMView para la plataforma XauceMóvil permite visualizar los contenidos empaquetados bajo el estándar SCORM, ya sea 2004 o 1.2, contribuyendo así al aprendizaje móvil.
- La realización de pruebas unitarias y de aceptación al visor SCORMView contribuye al correcto funcionamiento de la propuesta de solución y a que esta cumpla con las funcionalidades descritas por el cliente.

Recomendaciones

- Implementar una solución para que el visor reproduzca los archivos de audio y video en formato flash que puedan tener los paquetes SCORM.
- Desarrollar una nueva versión que pueda guardar el avance de los usuarios en los cursos empaquetados bajo el estándar SCORM.
- Desarrollar soluciones similares para otros sistemas operativos que son utilizados en los dispositivos móviles.

Referencias bibliográficas

[Aglesa, 2012] Aglesa, L. (2012). Definición de dispositivo móvil.

[Amaya Balaguera, 2013] Amaya Balaguera, Y. D. (2013). Metodologías ágiles en el desarrollo de aplicacionespara dispositivos móviles. estado actual.

[Avison and Fitzgerald, 2006] Avison, D. E. and Fitzgerald, G. (2006). Information system development. maidenhead: Mcgraw-hill education.

[Beck, 2000] Beck, K. (2000). *Extreme Programming Explained*. Addison-Wesley.

[Blanco and de autores, 2009] Blanco, P. and de autores, C. (2009). *Metodología de desarrollo ágil para sistemas móviles. Introducción al desarrollo con Android y el iPhone*. Madrid.

[Caituiro Monge, 2011] Caituiro Monge, H. (2011). Patrón memento.

[Campi et al., 2007] Campi, A., Ceri, S., Duval, E., Guinea, S., Massart, D., and Ternier, S. (2007). The prolearn query language.

[de Cuba, 2011] de Cuba, P. C. (2011). *Lineamientos de la política económica y social del Partido y la Revolución*.

[Developers, 2015] Developers, A. (2015). android.support.v4.app.

[Flétscher Bocanegra and Morales González, 2006] Flétscher Bocanegra, L. A. and Morales González, I. (2006). Modelo de desarrollo de servicios m-learning, una propuesta desde la concepción del servicio hacia la pedagogía.

[Gironés, 2011] Gironés, T. (2011). *El Gran Libro de Android. Barcelona: Marcombo*.

[González Castellanos, 2013] González Castellanos, J. (2013). Desarrollo de una aplicación para móviles para la plataforma educativa zera.

[Guevara Soriano, 2010] Guevara Soriano, A. (2010). Dispositivos móviles.

[Gómez Oliver, 2013] Gómez Oliver, S. (2013). Manual de programación android. versión 3.0.

[Gómez Peñaranda, 2012] Gómez Peñaranda, Y. A. (2012). Instalación de motodev studio for android.

[Izarra, 2010] Izarra, C. (2010). Movile learning.

[Joskowicz, 2008] Joskowicz, J. (2008). Reglas y prácticas en extreme programming.

[Larman, 1999] Larman, C. (1999). *UML y Patrones. Introducción al análisis y diseño orientado a objetos*.

[Larramendi Ferraz, 2012] Larramendi Ferraz, C. B. (2012). Propuesta de arquitectura para desarrollo de aplicaciones compuestas para dispositivos móviles con android.

[Mariano, 2008] Mariano (2008). Aprendizaje electrónico móvil o mobile learning: m-aprendizaje.

[Marrero Zamora, 2011] Marrero Zamora, A. (2011). Cuadernos de educación y desarrollo.

[McKenzie, 2011] McKenzie, D. (2011). Designing for android.

[Montemagno, 2015] Montemagno, J. (2015). Android tips: Hello material desing v7 appcompat.

[Mora G, 2010] Mora G, C. H. (2010). ¿qué es el móvil learning?

[Moreno Guerrero, 2011] Moreno Guerrero, A. J. (2011). Móvil learning.

[Peláez, 2009] Peláez, J. (2009). *Arquitectura basada en capas. Algunas notas sobre tecnologías Microsoft 2009*.

[Phat, 2013] Phat, H. V. (2013). Quick action pattern in android and simple implementation.

[Pérez Rodríguez and González Martínez, 2013] Pérez Rodríguez, L. O. and González Martínez, D. (2013). Sistema de control de laboratorios (clab).

[RELOAD,] RELOAD. Reload projects: Scorm player.

[RHODA,] RHODA. Bienvenido al repositorio de objetos de aprendizaje.

[Rinaldi, 2011] Rinaldi, M. (2011). Revolución mobile learning. 15 clases en 15 días.

[Rodríguez, 2012] Rodríguez, O. (2012). El estándar scorm — unia opencourseware.

[Rubio Gómez et al., 2011] Rubio Gómez, C., González Moreno, V., and Selas Felguera, J. (2011). En clave de tic.

[Ruiz, 2009] Ruiz, C. (2009). Impacto de las tics en nuestra sociedad.

[UniVerso, 2012] UniVerso (2012). Impacto de la tecnología móvil en la educación.

[Vazquez-Reina, 2011] Vazquez-Reina, M. (2011). M-learning: aprender a través del móvil: Eroski consumer.

[Verdecia Espinoza, 2012] Verdecia Espinoza, R. (2012). Análisis y diseño de un visor de paquetes scorm para la plataforma educativa zera.

[Vincent et al., 2006] Vincent, M., Duverny, G., Ghezal, M., and Caldeira, G. (2006). Scorm 2004: Content aggregation model. ganesha lms community.

[Xamarin, 2014] Xamarin (2014). Android support library v4.

[Álvarez, 2008] Álvarez, J. M. (2008). Dispositivos móviles.

Acrónimos

ADL Advanced Distributed Learning

ADT Android Development Tools

API Aplication Programming Interface de Android

CRC Class, Responsabilities and Collaboration

CRODA Creando Objetos de Aprendizaje

DOM Document Object Model

e-learning aprendizaje electrónico

FORTES Centro de Tecnologías para la Formación

GPS Sistema de Posicionamiento Global

HTML HyperText Markup Language

HU Historia de Usuario

IDE Entorno de desarrollo integrado

IMS Information Management System

LMS Learning Management Systems

m-learning mobile learning

NLN National Learning Network

OA Objeto de Aprendizaje

PDA Asistente Personal Digital

PLQL The ProLearn Query Language

POO Progamación Orientada a Objetos

RAM Memoria de Acceso Aleatorio

RHODA Repositorio de Objetos de Aprendizaje

SCO Objeto de Contenido Intercambiable

SCORM Sharable Content Object Reference Model

SDK Softaware Development Kit

SGML Lenguaje de Marcado de Anotaciones Generales

SMS mensajería instantánea

TICs Tecnología de la Información y las Comunicaciones

UCI Universidad de las Ciencias Informáticas

URI Uniform Resource Identifier

WiFi Wireless Fidelity

XML Extensible Markup Language

XP Extreme Programming