



Universidad de las Ciencias Informáticas

Facultad 1

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

**Visor de paquetes de contenido educativo pa-
ra dispositivos móviles.**

Autora:

Beatriz Ontivero González

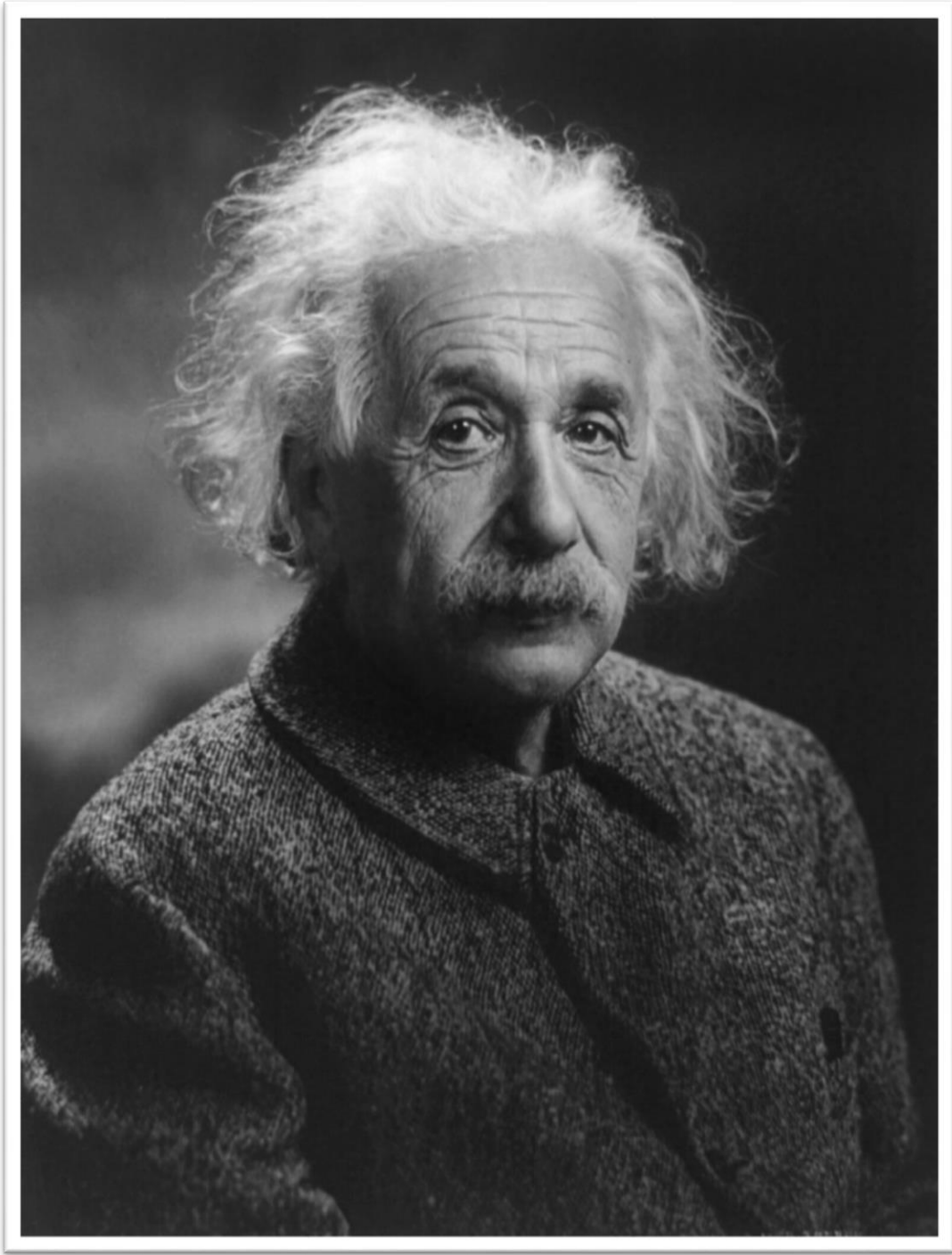
Tutores:

Msc. Iván Pérez Mallea

Ing. Leonardo Quesada Cruz

La Habana, junio de 2017

“Año 59 de la Revolución”



“Todos somos ignorantes. Lo que ocurre es que no todos ignoramos las mismas cosas.”

Einstein

|

DECLARACIÓN JURADA DE AUTORÍA

Declaro ser la autora del trabajo de diploma titulado “Visor de paquetes de contenido educativo para dispositivos móviles” y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2018.

Autora

Beatriz Ontivero González

Tutor

Ing. Leonardo Quesada Cruz

Tutor

Msc. Iván Pérez Mallea

DEDICATORIA

Dedicada a mi familia, en especial a mi mamá y a mi abuelo, por su paciencia, apoyo y dedicación.

Espero que estén orgullosos.

AGRADECIMIENTOS

A mi familia, principalmente a mi mamá y mi abuelo por haberme educado como la persona que soy en la actualidad; muchos de mis logros se los debo a ustedes.

A Leo, que siempre me ayudó en todo lo que necesitaba, incluso cuando no le tocaba, sin su ayuda yo no estuviese aquí este día.

A David, por aguantarme tanto y ayudarme en cada cosa que me hizo falta, aun estando tan lejos.

A todos los trabajadores de Droidlab, que me ayudaron durante todo el proceso de tesis, específicamente a: Ivelisse, Elvis, Abel, y una vez más, a Leo; sin ustedes todo hubiese sido más difícil.

A mis tutores, por el apoyo que me brindaron durante todo el curso.

A mi novio, por ayudarme, apoyarme y sobre todo, aguantarme, en todo este período de estrés que sé que debió ser muy difícil para él. Gracias mi niño.

A las muchachas del apartamento por todos estos años de convivencia. A mis compañeros de aula con los que me divertí muchísimo y sé que cuando nos graduemos, los voy a extrañar.

A mis amigos, que me han ayudado a lo largo de la carrera: Adannis, Carlos Yordan, Ariel García, Raúl Sánchez, Oscar, Rodas, Carlitos, Daniela Moreno, Eileen.

A los que me han apoyado y han estado presente en todo momento, en las malas y en las buenas: Juan, George, Fito, Cervantes, José Carlos, Sheila, Henry.

Y a todas las amistades que me quedan por mencionar.

Gracias a todos.

RESUMEN

El estándar SCORM es ampliamente utilizado en el área de la educación debido a que permite una interrelación entre los objetos de contenido, modelos de datos y protocolos, posibilitando presentar material didáctico estructurado de forma adecuada. Con el objetivo de facilitar la visualización, navegación e interacción con contenidos educativos empaquetados con el estándar *SCORM*, se desarrolló un visor de paquetes de contenido educativo para dispositivos móviles, llamado "*LearnView*". La aplicación permite interpretar paquetes con el estándar *SCORM* para acceder al contenido educativo. La estructura se puede desglosar en dos áreas principales: Administración de Paquetes y Visualización de Contenido. La primera área se enfoca en el trabajo con los paquetes donde se pueden realizar tareas como: Insertar, Mostrar, Ordenar, Buscar y Eliminar Paquetes. *Learnview* organiza la información en un solo directorio definido por el usuario, donde serán almacenados todos los paquetes importados. La segunda área se enfoca en la visualización de los objetos de aprendizaje a través del índice de contenidos. Para regir el proceso de desarrollo de la aplicación se utilizó como metodología de desarrollo de *software* AUP-UCI, como lenguaje de modelado *UML*, como herramienta de modelado *Visual Paradigm*, como lenguaje de programación Java y como entorno de desarrollo *Android Studio*. La arquitectura y tecnologías empleadas facilitan la incorporación de nuevas funcionalidades, con lo que se obtiene una herramienta flexible, escalable y adaptable. Fueron ejecutadas pruebas a la solución, proceso que culminó con la aceptación por parte del cliente.

Palabras clave: contenidos educativos, dispositivos móviles, paquetes, SCORM.

ABSTRACT

The SCORM standard is very widely used in the area of education because it allows an interrelation among content objects, data models and protocols, making it possible to present appropriately structured teaching materials. A viewer of educational content packages for mobile devices, called "LearnView", was developed in order to facilitate visualization, navigation and interaction with educational content packaged using the SCORM standard. The application allows interpreting packages with the SCORM standard to access the educational content. The structure is divided in two main areas: Package Management and Content Visualization. The first area is focused on working with the packages. This area can to perform tasks such as Insert, Show, Sort, Search and Delete Packages. Learnview organizes the information in only one directory defined by the user, where they will be stored all los imported packages. The second part permit the visualization of learning objects through the content index. To guide the development process of the application, AUP-UCI was used as a software development methodology, UML as a modeling language, Visual Paradigm as a modeling tool, Java as a programming language and Android Studio as a development environment. The architecture and technologies used facilitate the incorporation of new functionalities, obtaining a flexible, scalable and adaptable tool. The solution was tested, a process that culminated in the acceptance by the client.

Keywords: *educative contents, mobile devices, packages, SCORM.*

ÍNDICE DE CONTENIDO

Introducción.....	1
Capítulo 1. Fundamentación teórica.....	9
1.1. Conceptos asociados al dominio del problema.....	9
1.1.1. Objetos de Aprendizaje.....	9
1.1.2. Sistema de Gestión de Aprendizaje.....	11
1.1.3. Sistemas de aprendizaje.....	13
1.1.4. Estándares y especificaciones de empaquetado de contenido educativo.....	14
1.1.5. Estándar <i>SCORM</i>	17
1.1.6. Android.....	21
1.2. Estudio de aplicaciones similares.....	21
1.2.1. Moodle Mobile.....	21
1.2.2. SCORM Player.....	22
1.2.3. G-OpenLMS.....	22
1.2.4. Blackboard Mobile.....	22
1.2.5. TeachByApp SCORM Player.....	23
1.2.6. SCORM Reader.....	23
1.2.7. SCORMView.....	23
1.2.8. Conclusiones del estudio de las aplicaciones similares existentes.....	24
1.3. Metodología de desarrollo.....	25
1.4. Tecnologías.....	26
1.4.1. Lenguaje de modelado.....	26
1.4.2. Herramienta de modelado.....	26
1.4.3. Lenguaje de programación.....	27
1.4.4. Herramientas y tecnologías de desarrollo.....	27
1.4.5. Herramienta para el diseño de prototipo de interfaz.....	28
1.4.6. Herramienta de pruebas.....	28
1.5. Valoraciones finales.....	29
Capítulo 2. Análisis y diseño de la propuesta de solución.....	30

2.1.	Propuesta de solución	30
2.2.	Captura de requisitos.....	31
2.2.1.	Requisitos funcionales	32
2.2.2.	Requisitos no funcionales	35
2.3.	Historias de usuario	35
2.4.	Diseño	39
2.4.1.	Descripción de la arquitectura.....	39
2.4.2.	Patrones de diseño.....	43
2.4.3.	Diagrama de clases de diseño.....	44
2.5.	Valoraciones finales.....	46
Capítulo 3. Implementación y prueba.....		47
3.1.	Implementación	47
3.1.1.	Estándares de codificación	47
3.2.	Pruebas de software	48
3.2.2.	Estrategia de pruebas.....	49
3.1.2.	Ejecución y resultados de las pruebas de software.....	53
3.3.	Validación de la investigación	57
3.4.	Valoraciones finales.....	59
Conclusiones generales.....		60
Recomendaciones.....		61
Referencias Bibliográficas		62
Anexos.....		1
	Anexo 1. Entrevista al cliente	1
	Anexo 2. Encuesta a los estudiantes del CUM	2
	Anexo 3. Acta de aceptación.....	3
	Anexo 4. Historias de Usuario	4

ÍNDICE DE FIGURAS

Figura 1: Organización y agregación de contenidos, SCO y Asset. Elaboración propia.	19
Figura 2: Organización de la estructura del imsmanifest.xml de un paquete SCORM. Elaboración propia.	20
Figura 3: Descripción de la propuesta de solución. Elaboración propia.....	31
Figura 4: Lista de paquetes.....	37
Figura 5: Listar paquetes SCORM importados. Con menú de opciones.....	37
Figura 6: Listar paquetes SCORM importados. Actualizando vista.	37
Figura 7: Índice de contenido educativo del paquete SCORM.	39
Figura 8: Comparación entre el Modelo Vista Controlador y el Modelo Vista Presentador. Elaboración propia.....	41
Figura 9: Diagrama de paquetes que ilustra la estructura de la propuesta de solución, basada en MVP. Elaboración propia.	42
Figura 10: Diagrama de clases del diseño. Elaboración propia.	46
Figura 11: Resultados de JUnit en la clase UnzipTest.	55
Figura 12: Resultados de JUnit en la clase ListTest.	55
Figura 13: No conformidades encontradas por iteración en el proceso de pruebas. Elaboración propia. ...	57
Figura 14: Importar un paquete SCORM. Elaboración propia.	6
Figura 15: Seleccionar formato en que se encuentra el paquete SCORM a importar. Elaboración propia. ..	6
Figura 16: Seleccionar paquete en formato de carpeta dentro del dispositivo. Elaboración propia.	7
Figura 17: Seleccionar paquete en formato comprimido dentro del dispositivo. Elaboración propia.....	7
Figura 18: Paquete importado. Elaboración propia.	8
Figura 19: Visor de contenido educativo. Elaboración propia.	10
Figura 20: Mensaje para seleccionar el directorio de almacenamiento. Elaboración propia.	13
Figura 21: Seleccionar directorio de almacenamiento. Elaboración propia.....	13
Figura 22: Opción Seleccionar directorio. Con el explorador de archivos abierto. Elaboración propia.....	15
Figura 23: Seleccionar directorio. Con ruta definida. Elaboración propia.	15
Figura 24: Mensaje de verificación de eliminar. Elaboración propia.	17
Figura 25: Eliminar. Con elemento marcado. Elaboración propia.....	17
Figura 26: Lista ordenada A-Z. Elaboración propia.	19
Figura 27: Lista ordenada Z-A. Elaboración propia.	19

Figura 28: Buscar. Sin elementos. Elaboración propia.	21
Figura 29 Buscar. Resultado de la búsqueda. Elaboración propia.	21
Figura 30: Importar paquete SCORM. Elaboración propia.	23
Figura 31: Importar paquete desde Url. Elaboración propia.	23

ÍNDICE DE TABLAS

Tabla 1: Comparación entre visores de contenido educativo en formato SCORM para dispositivos móviles. Elaboración propia.	24
Tabla 2: Descripción de los requisitos funcionales. Elaboración propia.....	32
Tabla 3: Historia de Usuario Listar paquetes SCORM importados. Elaboración propia.....	36
Tabla 4: Historia de usuario: Mostrar índice de contenidos del paquete SCORM. Elaboración propia.	38
Tabla 5: Caso de Prueba de Validación para la Historia de Usuario Listar paquetes SCORM importados. Elaboración propia.	50
Tabla 6: Historia de Usuario Importar paquete SCORM desde el dispositivo. Elaboración propia.....	4
Tabla 7: Historia de Usuario Mostrar paquete SCORM importado. Elaboración propia.....	7
Tabla 8: Historia de Usuario Mostrar contenidos del paquete SCORM. Elaboración propia.	8
Tabla 9: Historia de Usuario Seleccionar directorio de almacenamiento de paquetes. Elaboración propia.	10
Tabla 10: Historia de Usuario Modificar directorio de paquetes. Elaboración propia.	13
Tabla 11: Historia de Usuario Eliminar paquete SCORM de la aplicación. Elaboración propia.....	15
Tabla 12: Historia de Usuario Ordenar listado de paquetes SCORM importados. Elaboración propia.	17
Tabla 13: Historia de usuario Buscar paquete SCORM de la lista de paquetes. Elaboración propia.	19
Tabla 14: Historia de Usuario Importar paquete SCORM desde URL. Elaboración propia.....	21

Visor de paquetes de contenido educativo para dispositivos móviles

Introducción.

La Educación a Distancia (EaD), se interpreta a nivel mundial, como un modelo de educación que brinda un espacio de intercambio entre profesores y estudiantes, en el cual se ofrece una gran cantidad de información que puede ser consultada desde cualquier lugar y momento; contribuyendo así, a la integración cultural entre la universidad y la sociedad. (1)

En la educación hemos sido testigos de que los recursos didácticos mediados por tecnologías son una alternativa para desarrollar procesos de aprendizaje. La implementación de estos medios ha generado el diseño de diversos ambientes más allá de los presenciales y es así como es común ahora contar con el apoyo de estos recursos tecnológicos (con mayor o menor uso). (2)

Se habla de ambientes multimodales, ambientes combinados o mixtos (*blended learning, b-Learning, Young, 2002*), ambientes digitales en línea por Internet (*electronic learning, e-Learning, Pastore, 2002*) y recientemente, ambientes de aprendizaje móvil (*mobile learning, m-learning, Laouris, 2005*). (2) En estos ambientes educativos, se realizan diferentes formas de enseñanza, y en cada una de ellas, se utiliza un componente indispensable en el desarrollo de los cursos educativos: el Objeto de Aprendizaje (OA).

Un OA "...es una forma de transmitir el conocimiento utilizando el computador y tiene por finalidad la construcción de pequeños componentes que puedan ser reutilizados y ensamblados en diferentes contextos de aprendizaje..." (3). Los OA son almacenados en repositorios y en Sistemas de Gestión de Aprendizaje, brindando a los estudiantes la posibilidad de ingresar en el programa para acceder a cursos en línea. (3)

Actualmente, la EaD a nivel internacional, está caracterizada por un uso intensivo de las Tecnologías de la Información y las Comunicaciones (TIC), no siendo así en nuestro país. En Cuba se apreciaron manifestaciones propias de lo que luego sería la universidad de educación a distancia, dirigida a la formación profesional de alto nivel, sustentada en el trabajo independiente y la auto-preparación. Sus principales contenidos estaban representados mediante materiales didácticos escritos, medios audiovisuales adecuados, como televisión educativa, el video, la radio, el audio-casete e incluso la prensa escrita. Surgió como un modelo de desarrollo mixto en los propios centros de Educación Superior, vinculando la educación a distancia con la educación presencial. (4)

Visor de paquetes de contenido educativo para dispositivos móviles

Actualmente en nuestro país, el Ministerio de Educación Superior ha dado un gran impulso a la modalidad de EaD; razón por la cual se crea el Centro Nacional de Educación a Distancia (CENED) en el año 2014. El objetivo fundamental de CENED es contribuir al desarrollo y la excelencia de la educación a distancia en Cuba, incrementando su competitividad a través de la difusión, la mejora continua y la aplicación creadora de las TIC.

La primera tarea del CENED fue la creación de un Modelo de Educación a Distancia para la Educación Superior Cubana que contemplara un uso intensivo de las TIC. Dicho Modelo contempla, principalmente, un componente tecnológico, que plantea que un estudiante puede moverse por tres escenarios tecnológicos atendiendo a diferentes niveles de conectividad a la red:

1. **Escenario de total conectividad**, donde el usuario tiene acceso a la red, plataformas de enseñanza y todo tipo de recursos puestos en ella.
2. **Escenario de conectividad parcial o limitada**, donde el estudiante solo posee acceso desde enlaces de baja velocidad, correo electrónico o acceso telefónico.
3. **Escenarios sin conectividad**, donde el estudiante posee dispositivos tecnológicos como computadoras, *tablets*, *smartphones*, y otros; pero no posee ningún tipo de conectividad a la red.

A pesar de estos escenarios, el modelo plantea que el profesor siempre diseñará las actividades y contenidos para el escenario de total conectividad teniendo en cuenta los otros escenarios. En el escenario de total conectividad, los contenidos se montan en una plataforma que permita la gestión del aprendizaje, y los recursos pueden estar ubicados en repositorios que sigan normas de empaquetado y catalogación de recursos educativos. A partir de esta situación se debe cumplir que el estudiante:

- ✓ Pueda acceder al contenido y los recursos en cualquier situación y variedad de dispositivos.
- ✓ Los contenidos deben estar a su disposición no solo cuando está conectado a la red sino también cuando esté desconectado para garantizar sus horas de estudio.

Esto puede implicar que el profesor tenga que diseñar y producir los recursos más de una vez dependiendo de las condiciones tecnológicas presentes en los escenarios. Los recursos se almacenan en repositorios y se elaboran siguiendo estándares para la producción y empaquetado de contenidos digitales que pueden ser compartidos en la red.

Visor de paquetes de contenido educativo para dispositivos móviles

La Msc. María J. Vidal Ledo, Msc. Xaily Gavilondo Mariño, Lic. Alfredo Rodríguez Díaz y el Lic. Armando Cuéllar Rojas, de la Escuela Nacional de Salud Pública, en la Habana, Cuba; publicaron en 2015, un artículo en el sitio oficial de la Educación Nacional Superior de nuestro país. En el mismo exponen, fundamentalmente, las ventajas y desventajas del uso de los dispositivos móviles en la educación y formación. Dichas ventajas son:

- ✓ **Accesibles:** Los dispositivos móviles son accesibles, debido a que, con el paso de los años, disminuye el costo.
- ✓ **Portables:** A diferencia de las computadoras, los celulares y tabletas son más ligeras y se aprovechan mejor en cuanto a lugar y momento, permitiendo que los métodos de enseñanzas se entremezclen con las actividades cotidianas.
- ✓ **Navegación:** Al igual que en los ordenadores portátiles o de escritorio, es posible la navegación por la red, *Internet*. Y dependiendo de la institución y del país, pueden tener conectividad y correo de manera inalámbrica.
- ✓ **Gestión de documentos:** Es posible la creación, edición, salva y escaneo de documentos.
- ✓ **Multimedia:** Es posible grabar conferencias o visitas médicas, ya sea por audio, imágenes o video.
- ✓ **Seguridad:** Los sistemas operativos instalados en esta tecnología, son menos propensos a ser atacados por virus; no obstante, su restauración es fácil y rápida.
- ✓ **Aplicaciones:** Se pueden instalar aplicaciones que faciliten el intercambio de información.
- ✓ **Menor costo de información:** Se disminuyen los gastos económicos y ecológicos, ya que estos dispositivos pueden almacenar una gran cantidad de literatura docente digital de menor precio, que estos mismos libros en formato duro.
- ✓ **Ahorra electricidad:** Los *tablets* y dispositivos móviles consumen 7 veces menos electricidad que una computadora de escritorio.

Sin embargo, estos dispositivos también poseen **desventajas**, tales como: que son más fáciles de extraviarse; provocan distracciones que apartan al alumno del estudio; contienen una pantalla más pequeña que los ordenadores portátiles o de escritorio; son más propensos a caerse y dañarse; con el continuo avance de las tecnologías, surge la posibilidad de que en una cantidad específica de años, la tecnología

Visor de paquetes de contenido educativo para dispositivos móviles

quede obsoleta; contienen pantalla táctil, por lo que no proporcionan un teclado cómodo a la hora de utilizarlo; sin embargo, se pueden agregar accesorios removibles o conectados por *bluetooth*. (5)

Para dar respuestas a las constantes necesidades de acercamiento a la información y de comunicación entre los propios individuos y las máquinas, surgen las tecnologías móviles. Esta nueva tecnología de acceso a las telecomunicaciones permite realizar acciones que antes solamente se encontraban disponibles en un ordenador de escritorio, con las consecuentes restricciones de movilidad. Una de las características más importantes que ofrecen los dispositivos móviles, es la capacidad de estar interconectados con la tecnología sin cable (*wireless*), y con el desarrollo de dispositivos portátiles e inteligentes, como son los *tablets* y los *smartphones*, que cada vez son más versátiles y funcionales y han provocado una revolución en la comunicación entre los usuarios. (6)

La principal ventaja que poseen los dispositivos móviles es que agrupan en un solo dispositivo, un conjunto de funcionalidades como: la comunicación por voz, mensajería instantánea (*SMS*), agenda de contactos, juegos, cámara fotográfica, calendario, acceso a Internet, reproducción de video e incluso, *GPS*¹ y reproductor *MP3*²... (7); ocasionando que para muchos individuos les sean indispensables en el trabajo, la escuela, la casa, es decir, en cualquier sitio que se encuentre.

Hoy en día, a diferencia de años anteriores, en Cuba, la mayoría de la población posee dispositivos móviles. En el Anuario estadístico 2016, se evidencia que desde el 2011 hasta el 2016, hubo un aumento de 2672 en cuanto a la cantidad de abonados del sistema celular. (8) Estos datos reflejan que la cantidad de abonados del sistema celular ha incrementado considerablemente.

Es una realidad, que la tecnología móvil, ha alcanzado importantes niveles de uso en el ámbito educativo, debido principalmente, al uso de la red inalámbrica, la cual permite que los alumnos y profesores se comuniquen en tiempo real, mediante el uso de las redes sociales y las aplicaciones de mensajería instantáneas; que agilice en cuanto a tiempo, el proceso de compartir los contenidos y materiales educativos mediante la nube, y que los alumnos puedan acceder a estos contenidos educativos sin importar su ubicación geográfica. (9)

¹ *GPS: Global Positioning System* (Sistema de Posicionamiento Global).

² *MP3: Proviene de las siglas MPEG-1 Audio Layer III*, perteneciente a la compañía que le da nombre, *Moving Picture Expert Group*. (6)

Visor de paquetes de contenido educativo para dispositivos móviles

Según el artículo Realidad aumentada, una evolución de las aplicaciones de los dispositivos móviles, publicado en la revista científica *Pixel-Bit*, Revista de Medios y Educación de la Universidad de Sevilla, España; expresa que "...los dispositivos móviles son herramientas mono-usuario que tienen cada vez más, protagonismo en las actividades independientes y no presenciales, que los alumnos usan habitualmente estos equipos, gestionando y transformando una tipología diversa de datos, donde realizan actividades que luego es convertido en conocimiento, aprovechando las ventajas y el atractivo de estos nuevos dispositivos electrónicos...". (10) De hecho, "...las tecnologías móviles han re-dibujado el panorama educativo, aportando a la educación no sólo movilidad sino también conectividad, ubicuidad y permanencia, características propias de los dispositivos móviles tan necesarias en los sistemas de educación a distancia..." (11); dando paso al aprendizaje móvil.

En la Universidad de las Ciencias Informáticas (UCI) se encuentra la sede del Centro Nacional de Educación a Distancia donde se hace uso de los escenarios planteados en el modelo de EaD para Educación Superior Cubana. Paralelamente, la UCI posee centros de desarrollo de producción de *software*, dentro de los cuales se encuentra el Centro de Tecnología para la Formación (FORTES), ubicado en la Facultad 4. La Plataforma Educativa (ZERA) y el Repositorio de Objetos de Aprendizaje (RHODA) son ejemplos concretos de software educativos, desarrollados en este Centro. (12)

RHODA tiene como objetivo gestionar objetos de aprendizajes empaquetados en formato *PIF* bajo el estándar *SCORM*³. Es un lugar de trabajo metodológico colaborativo orientado a elevar la calidad de los recursos educativos que este repositorio brinda. Además, no solo permite visualizar los cursos, sino que también posibilita exportarlos a *SCORM 1.2* y *SCORM 2004*. (13)

Los visores de los paquetes de contenido educativo existentes, la mayoría son para *PC*, y las aplicaciones que son para dispositivos móviles, poseen las siguientes limitaciones:

- ✓ No explotan correctamente o de manera total las posibilidades de interacción de estos paquetes.
- ✓ Son de pago, y las aplicaciones gratuitas existentes, es necesario efectuar el pago para descargar los objetos de aprendizaje disponibles en los *LMS*⁴.

³ *SCORM: Shareable Content Object Reference Model* (Modelo de Referencia para Objetos de Contenido Intercambiables).

⁴ *LMS: Learning Management System* (Sistema de Gestión de Aprendizaje).

Visor de paquetes de contenido educativo para dispositivos móviles

- ✓ Dependen de manera parcial o total, de una conexión con un *LMS* específico o a un repositorio educativo. (12)

En FORTES se desarrolló un visor de paquetes de contenido educativo para dispositivos móviles, llamada *SCORMView*(14), de la cual se profundizará en el Capítulo 1, epígrafe 1.2: Estudio de aplicaciones similares. No obstante, existen visores para estos paquetes de contenido, pero casi todos para PC y las aplicaciones para dispositivos móviles existentes, incluyendo esta última mencionada, no explotan correctamente o de manera total las posibilidades de interacción de estos paquetes. Por tanto, los estudiantes de educación a distancia que se encuentren en escenarios de conectividad parcial o sin conectividad, no pueden acceder a los contenidos y recursos educativos que se encuentren en sistemas de gestión de aprendizajes o repositorios educativos.

Partiendo de la **situación problemática** planteada, surge el siguiente **problema de investigación**: ¿Cómo lograr que los contenidos y recursos almacenados en repositorios estén disponibles para los estudiantes de EaD en los escenarios conectividad parcial o sin conectividad, sin que el profesor tenga que diseñar varios tipos de recursos para un mismo contenido?

Se establece como **objeto de estudio** los visores de contenidos educativos, enmarcado en el **campo de acción** los visores de contenidos educativos para dispositivos móviles.

Para dar solución al problema de investigación planteado se define como **objetivo general** desarrollar una aplicación para dispositivos móviles con Sistema Operativo Android que permita la visualización, navegación e interacción con contenidos educativos empaquetados con el estándar *SCORM*.

El objetivo general se desglosa en los siguientes **objetivos específicos**:

- 1 Definir el marco teórico de la investigación a través del estudio del estado del arte acerca de los visores de paquetes de contenido educativo para dispositivos móviles.
- 2 Implementar la aplicación visor de paquetes de contenido educativo para dispositivos móviles.
- 3 Realizar pruebas a la propuesta de solución desarrollada para interpretar e visualizar los paquetes de contenido educativo para dispositivos móviles.

Teniendo como **hipótesis**: El desarrollo de una aplicación para visualizar los paquetes de contenido educativo para dispositivos móviles garantizará que los contenidos y recursos almacenados en reposito-

Visor de paquetes de contenido educativo para dispositivos móviles

rios estén disponibles para los estudiantes de EaD en los escenarios de conectividad parcial o sin conectividad.

Obteniendo como **variable independiente**, la aplicación móvil; y como **variable dependiente**, acceso al contenido educativo.

Los métodos de investigación a utilizar para el cumplimiento del objetivo general son:

✓ **Métodos teóricos:**

- ✓ **Analítico–Sintético:** se utiliza para el estudio de las fuentes bibliográficas existentes referente a las aplicaciones móviles que visualizan paquetes de contenido educativo, identificando los elementos más importantes y necesarios para dar solución al problema planteado.
- ✓ **Histórico-Lógico:** se utiliza con el fin de realizar un estudio referente a las aplicaciones existentes que visualizan los paquetes de contenido educativo.

✓ **Métodos empíricos:**

- ✓ **Entrevista:** Las entrevistas con el cliente se realizan durante todo el proceso de desarrollo de la aplicación. Se pone de manifiesto, desde la primera reunión cliente-desarrollador, donde se especifican los requisitos funcionales y no funcionales de la aplicación; hasta la última entrega del producto terminado al cliente.
- ✓ **Observación:** Se utiliza para analizar el comportamiento de las aplicaciones similares estudiadas, que visualicen los paquetes de contenido educativo en dispositivos móviles.

El presente trabajo de diploma consta de tres capítulos estructurados de la siguiente forma:

- ✓ **Capítulo 1. Fundamentación teórica:** se realiza el estudio del estado del arte de los visores de paquetes de contenido educativo para dispositivos móviles, se abordan elementos teóricos vinculados a la investigación, se analiza la metodología de desarrollo a emplear, y se investigan las aplicaciones existentes que den solución a problemas similares, así como las tecnologías y herramientas a utilizar en el desarrollo de la aplicación.
- ✓ **Capítulo 2. Análisis y diseño de la propuesta de solución:** se describe la solución propuesta y se ofrecen detalles de los principales aspectos relacionados con su diseño. Se realiza el levantamiento de los requisitos funcionales y no funcionales de la solución propuesta. Se explica el fun-

Visor de paquetes de contenido educativo para dispositivos móviles

cionamiento de la aplicación a través de historias de usuario, que incluyen las descripciones y prototipos de interfaces correspondientes de cada uno de ellos para lograr una mejor comprensión. Se define la arquitectura de software con la que estarán representados los componentes de la aplicación, así como los patrones de diseño a utilizar.

- ✓ **Capítulo 3. Implementación y prueba:** en este capítulo se describen los elementos relacionados con la implementación, reflejando las prácticas de programación y los estándares de codificación utilizados para el cumplimiento de los requisitos funcionales especificados previamente. Además, se describen las estrategias trazadas para el desarrollo de las pruebas realizadas a la aplicación, con el objetivo de validar el correcto funcionamiento de las funcionalidades implementadas, así como los resultados obtenidos luego de aplicadas dichas pruebas.

Visor de paquetes de contenido educativo para dispositivos móviles

Capítulo 1. Fundamentación teórica.

Con el fin de desarrollar una solución para el problema de investigación planteado, es necesario definir el marco teórico de la investigación. Dentro del mismo, se abordan los conceptos relacionados al dominio del problema; se analizan las aplicaciones existentes que solucionen problemáticas similares; se define la metodología de desarrollo por la que se regirá la investigación; y se puntualizan las herramientas y tecnologías a utilizar.

1.1. Conceptos asociados al dominio del problema.

Para lograr una mejor comprensión sobre los visores de paquetes de contenido educativo estandarizados por *SCORM* para dispositivos móviles, es necesario analizar los criterios generales que aportan aquellas investigaciones referentes al mismo. A continuación, se presentan conceptos principales.

1.1.1. Objetos de Aprendizaje.

El término **objeto de aprendizaje** fue definido por primera vez en 1967, en el libro *Ciencia aplicada y el progreso de la tecnología*, escrito por *Gerard R.W.*, como "...un conjunto de contenidos, ejercicios y elementos de evaluación que se combinan en relación a un único objetivo de aprendizaje..." (15); popularizado por *Wayne Hodgins* en 1994 con su grupo de trabajo: *Arquitecturas de aprendizaje, API y objetos de aprendizaje*. (16)

El Instituto de Ingeniería Eléctrica y Electrónica, mejor conocida por sus siglas en inglés *IEEE (Institute of Electrical and Electronic Engineers)*, expone que "...un Objeto de Aprendizaje se define como cualquier entidad, digital o no digital, que se puede usar para el aprendizaje, la educación o la capacitación...". (17)

David A. Wiley II, profesor adjunto de psicología de la instrucción y tecnología, de la Universidad *Brigham Young*, expresó que: "...los objetos de aprendizaje son elementos de un nuevo tipo de instrucción basada en computadora basada en el paradigma orientado a la ciencia de la informática...". (18)

Mientras que, *Adrian Villegas Dianta*, escribió un artículo titulado: *Los Objetos Digitales de Aprendizaje*, en la revista científica *e-Historia*, en el cual precisa que un Objeto de Aprendizaje es "...una colección de objetos de información ensamblada usando metadatos para corresponder a las necesidades y personalidad de un aprendiz en particular. Múltiples objetos de aprendizaje pueden ser agrupados en conjuntos más grandes y anidados entre sí para formar una infinita variedad y tamaños...". (19) *Francisco José Gar-*

Visor de paquetes de contenido educativo para dispositivos móviles

cía Peñalvo expresó que: "...que un OA es un conjunto de recursos, diseñado con un propósito educativo y re-utilizable...". (20)

De los conceptos analizados, se utilizará en la investigación el expresado por Adrian Villegas Dianta, debido a que la autora considera que es el concepto más completo.

Un OA está compuesto por diferentes tipos de medias independientes como: mapas conceptuales, audios, imágenes, diapositivas, libros, videos o recursos más complejos como multimedias elaboradas u otros OA. Cada recurso provee además, metadatos, que son conjuntos de atributos que describen a dicho recurso, es decir datos sobre datos (21); Estos metadatos facilitan la búsqueda en repositorios digitales.

A través de los metadatos, el empaquetamiento de los contenidos y de su secuenciación; posibilita la interoperabilidad, re-usabilidad, adaptabilidad, accesibilidad y durabilidad de los OA; asegurando la calidad de los mismos.

- ✓ **Interoperabilidad:** Capacidad de operar entre diferentes plataformas tecnológicas, es decir, entre *hardware* y *software*.
- ✓ **Re-usabilidad:** Esta característica se trata de la posibilidad de que el objeto se pueda utilizar nuevamente en otros escenarios, permitiendo la incorporación y adaptación de los contenidos educativos, facilitando y disminuyendo el tiempo al desarrollo de este proceso y haciéndolos más portables.
- ✓ **Adaptabilidad:** Capacidad de que los objetos de aprendizaje se puedan adaptar a los futuros cambios que se produzcan en los entornos educativos.
- ✓ **Accesibilidad:** Capacidad de acceder dinámicamente a los recursos de aprendizaje disponibles en una base de datos o repositorios.
- ✓ **Durabilidad:** Esta propiedad se refiere a que las actualizaciones, tanto de hardware, como de software, no afecta al objeto de aprendizaje, en otras palabras, permanecen intactos, ante cambios externos al mismo.
- ✓ **Portable:** Cualidad que permite que el OA pueda moverse e instalarse en diferentes plataformas sin que cambie su estructura o contenido.

Visor de paquetes de contenido educativo para dispositivos móviles

Debido a estas características, se pueden catalogar para ser ubicados en repositorios, con el fin de que posteriormente, puedan ser reutilizados, modificados o re-elaborados. Por ello, Lorenzo García Aretio, Catedrático Emérito de la Facultad de Educación UNED de España, concreta que "...de ahí que el objeto y el repositorio sean dos entes complementarios. Un objeto que no guarde las características necesarias para poder integrarse en un repositorio, pierde todas sus virtualidades y a la vez, un repositorio que no cuente con una buena base de objetos, deja de ser interesante y operativo...". (22)

O sea, los OA almacenados en repositorios por si solos tienen un propósito educativo definido, pero realmente cobran importancia en la EaD, al ser insertados en los contenidos que administra un *LMS*.

1.1.2. Sistema de Gestión de Aprendizaje.

Andreas Janson, Matthias Söllner y Jan Marco Leimeister, del Centro de Investigación para el Diseño de Sistemas de Información, de la Universidad de *Kassel*, en Alemania; afirman que "...los Sistemas de Gestión de Aprendizaje se enfocan en entregar, evaluar y administrar la educación y la capacitación. Además, ofrecen un proceso de aprendizaje individualizado para apoyar a los usuarios con una retroalimentación efectiva a fin de involucrar el éxito del aprendizaje en las fases de aprendizaje autorregulado...". (23)

Aabha Chaubey, becario de investigación y el Dr. *Bani Bhattacharya*, profesor asociado, ambos del Departamento del Centro de Tecnología Educativa en la India, expresan que "...un Sistema de Gestión de Aprendizaje es un programa de *software* basado en la web o en la nube que ayuda a enseñar el proceso de aprendizaje y ayuda en la entrega efectiva de instrucción, entrenamiento y programa de desarrollo...". (24)

Mientras que, *Sunnie Lee Watson y William R. Watson*, profesores asistentes de diseño y tecnología del aprendizaje, de la Universidad de *Purdue*, expresan que "...un sistema de gestión de aprendizaje es el marco que maneja todos los aspectos del proceso de aprendizaje. Un *LMS* es la infraestructura que ofrece y gestiona el contenido instructivo, identifica y evalúa los objetivos de aprendizaje o capacitación individuales y organizacionales, rastrea el progreso hacia el cumplimiento de esos objetivos y recopila y presenta datos para supervisar el proceso de aprendizaje de la organización como un todo...". (25)

Se puede deducir que, un Sistema de Gestión de Aprendizaje, mejor conocido por sus siglas en inglés, *LMS*, es un software instalado en un servidor *web*, empleado para administrar, distribuir y controlar las actividades de aprendizaje no presencial, o de aprendizaje a distancia. Los *LMS* facilitan el seguimien-

Visor de paquetes de contenido educativo para dispositivos móviles

to del proceso de aprendizaje de los estudiantes, generan informes, realizan evaluaciones y gestionan servicios de comunicación como son los foros de debate, e incluso, videoconferencias. Teniendo como funciones principales:

- ✓ **Gestión de usuarios y registro de información:** Cada individuo que interactúa con el *LMS*; tanto estudiante, como profesor; tiene su propio perfil dentro del sistema; facilitando procesos de gestión, como son las matrículas, la selección de las asignaturas optativas, entre otros.
 - ✓ **Creación de contenidos educativos:** Algunos de los Sistemas de Gestión de Aprendizaje, tienen la opción de crear nuevos contenidos educativos, posibilitando que estos contenidos sean adaptados a las necesidades de los estudiantes y profesores.
 - ✓ **Discusión de contenido:** Los *LMS* nos permiten publicar los contenidos educativos en la red y editar los derechos de acceso para los individuos autorizados, de una manera rápida y eficaz.
 - ✓ **Planificación y organización de la formación:** Estos sistemas ofrecen un conjunto de herramientas que posibilitan la planificación y organización de los contenidos de los cursos académicos, como son las evaluaciones, conferencias y demás recursos de aprendizaje que contenga.
 - ✓ **Tutoría:** Un *LMS* facilita al profesor la tutoría de estudiantes, debido a que permite agilizar el acceso al historial académico de cada uno de ellos con el fin de realizar seguimientos personalizados a su trayectoria durante el curso.
 - ✓ **Fomento de la comunidad virtual de estudio:** Debido a que poseen herramientas de comunicación entre estudiantes y profesores; facilita la creación de un grupo de individuos que comparten los mismos objetivos e inquietudes, posibilitando la unión entre la comunidad virtual.
 - ✓ **Evaluación:** Permite evaluar los ejercicios que realizan los estudiantes mediante fórmulas de forma rápida y precisa, de tal manera que los estudiantes podrán saber su calificación de inmediato.
- (26)

Los *LMS* pueden ser plataformas bajo licencia o con recursos educativos abiertos. Entre los sistemas propietarios, se destaca *Blackboard*, por su gran flexibilidad y filosofía de trabajo, el cual hace que el aprendizaje sea atractivo, accesible y valioso, logrando la evolución de los métodos actuales de enseñanza. En cuanto a las plataformas de recurso abierto, sobresalen:

Visor de paquetes de contenido educativo para dispositivos móviles

- ✓ *Dokeos*, por ser uno de los primeros softwares que utilizan técnicas y metodologías en línea, en el ámbito de la educación universitaria en lengua francesa, adaptándose al aprendizaje informal, las redes sociales y la reubicación del personal.
- ✓ *Sakai*, ya que es una comunidad internacional que crea tecnología con el fin de mejorar el aprendizaje, la enseñanza y la investigación.
- ✓ *Moodle*, debido a que proporciona un sistema integrado único, robusto y seguro para crear ambientes de aprendizaje personalizados, financiada por una red social de cerca de 60 compañías, teniendo más de 65 millones de usuarios registrados, convirtiéndola en la plataforma más usada del mundo. (27)

Este tipo de *software* nos ofrece diversas ventajas al proceso de aprendizaje. La más importante de todas es que soportan un gran número de estudiantes matriculados y de profesores que interactúan con el sistema, integrando nuevas tecnologías que hace posible la educación a distancia con la calidad requerida. Otras, pero no menos importante, son: la reducción de tiempo y costo de procesos educativos y de gestión, como es la calificación de evaluaciones, la publicación de los recursos del curso en la red y la matrícula de los estudiantes en dichos cursos; facilita la colaboración entre los usuarios, mediante foros y videoconferencias; permite que los usuarios puedan tomarla capacitación a su ritmo y sin importar el lugar donde se encuentre siempre cuando tenga acceso a Internet. (24)

La combinación OA – LMS está presente en todos los ámbitos de enseñanza (*b-Learning*, *e-Learning*, *m-Learning*), siendo este último donde alcanzan mayor importancia, debido a que en muchas oportunidades los medios tienen un papel preponderante.

1.1.3. Sistemas de aprendizaje.

Con el desarrollo de las TIC, la educación se ha desplegado en diversos ambientes de enseñanza, tales como, ambientes combinados o mixtos (*b-Learning*), ambientes digitales en línea por Internet (*e-Learning*) y ambientes de aprendizaje móvil (*m-Learning*).

- ✓ **B-Learning:** Modelo de enseñanza y aprendizaje que combina la enseñanza presencial con la enseñanza no presencial. Según *Beatriz Fainholc*, profesora e investigadora del Centro de Diseño, Producción y Evaluación de Recursos de Aprendizaje, de la Facultad de Humanidades y Ciencias de la Educación, de la Universidad de la Plata, en Buenos Aires; considera que la modalidad *b-*

Visor de paquetes de contenido educativo para dispositivos móviles

Learning es "...una estrategia de formación más autogestionaria y autónoma por parte del estudiante como también más cercana y pertinente respecto de una aproximación individual y grupal en la construcción de conocimiento...". (28)

- ✓ **E-Learning:** Forma de educación a distancia, la cual es dependiente del uso de las tecnologías de la información y otros contenidos didácticos para la capacitación semi-presencial o no presencial de estudiantes en línea, a través de medios electrónicos; donde los estudiantes pueden consultar recursos, actividades y contenidos, sin compartir el mismo espacio físico; destacándose el uso de *Internet*. (20) Manuel Área, profesor de la Universidad de La Laguna, y Jordi Adell, profesor de la Universidad Jaume de Castellón, expresan que *e-Learning* "...es una modalidad de enseñanza-aprendizaje que consiste en el diseño, puesta en práctica y evaluación de un curso o plan formativo desarrollado a través de redes de ordenadores y puede definirse como una educación o formación ofrecida a individuos que están geográficamente dispersos o separados o que interactúan en tiempos diferidos del docente empleando los recursos informáticos y de telecomunicaciones...". (29)
- ✓ **M-Learning:** Es una adaptación del sistema *e-Learning*, que, a diferencia de este, utiliza los dispositivos móviles que posean la capacidad inalámbrica, ya sea teléfonos móviles, *tablets*, *i-pod*, entre otros; proporcionando más flexibilidad, en cuanto a momento y lugar, al proceso de enseñanza-aprendizaje del alumno. (2) Mientras que, *Francisco David Peña Esteban* y *María Concepción Burgos García*, expresan que el *Mobile Learning* es un tipo de modelo de aprendizaje que permite a los estudiantes obtener materiales de aprendizaje en cualquier sitio y en cualquier momento usando tecnologías móviles e *Internet*. (30)

Para poder utilizar estos OA en cualquier ámbito o *LMS* se necesita un sistema único que describa el objeto. Para esto existen diferentes estándares de empaquetado, tales como: *LOM*, *AICC*, *SCORM*; que facilitan esta tarea.

1.1.4. Estándares y especificaciones de empaquetado de contenido educativo.

Según la Real Academia de la Lengua Española, estándar es un tipo, modelo, patrón, nivel.

Christian Foix y Sonia Zavando expresan que el "...estándar *e-Learning* se refiere a un conjunto de reglas comunes por las que se rigen las compañías dedicadas a la tecnología *e-Learning*. Estas reglas es-

Visor de paquetes de contenido educativo para dispositivos móviles

pecifican a los desarrolladores como deben construir los cursos, así como, las plataformas sobre las cuales son impartidos estos cursos, de tal manera que puedan interactuar unas con otras...”. (31)

Para facilitar el uso de los objetos de aprendizaje, deben ser empaquetados rigiéndose por estándares, para luego ser soportados por las plataformas de aprendizaje. Consecuentemente, en un artículo de la revista científica RUSC (Revista de Universidad y Sociedad del Conocimiento) publicado en el 2007 por Josep M. Boneu, exponen los estándares más reconocidos en el ámbito educacional. En los que se destacan: “...el estándar *LOM (Learning Object Metadata)* de la *IEEE LTSC (Institute for Electrical and Electronic Engineers Learning Technology - Standards Committee)*, ampliamente aceptado debido a que permite describir un objeto de aprendizaje a través de metadatos; la *AICC (Aviation Industry CBT⁵ Committee)*, fue el primer organismo creado para desarrollar un conjunto de normas que permitieran el intercambio de cursos *CBT*; la descripción de itinerarios formativos es posible mediante las estructuras de descripción de recursos *RDF (Resources Description Framework)* o bien a través de *IMS LD (IMS⁶ learning design, basado en EML⁷)*, y el estándar *SCORM* de *ADL⁸*, ampliamente aceptado y utilizado, permite la organización de contenidos soportando la descripción de itinerarios formativos, secuenciación de contenidos, el empaquetamiento de los contenidos para su cómoda distribución, y el seguimiento del proceso de aprendizaje...”. (32)

IEEE LOM es empleado como referencia para la definición de metadatos. Sin embargo, posee una gran cantidad de elementos que para muchos usuarios son incomprensibles o carecen de sentido, situación que complica su uso debido por una parte al tiempo que se debe emplear y por otra a la definición no consensuada de la información que debe ser introducida en cada uno de ellos. La definición no consensuada de la información contenida en los metadatos puede ser un importante inconveniente a la hora de reutilizar OAs provenientes de diversas fuentes. (32)

AICC, en el 2005, impulsó la primera iniciativa para definir especificaciones con respecto al intercambio de cursos *CBT* entre plataformas de formación. Donde se resolvieron problemas como la carga de

⁵ *CBT: Computer Based Training* (Aprendizaje Basado en Computadora).

⁶ *IMS: Instruction management System* (Sistema de Gestión de Instrucciones).

⁷ *EML: Educational Modelling Language* (Lenguaje de Modelado Educativo).

⁸ *ADL: Advanced Distribute Learning* (Aprendizaje de Distribución Avanzada).

Visor de paquetes de contenido educativo para dispositivos móviles

cualquier curso en un *LMS* y la comunicación entre el *LMS* y el usuario de forma de obtener información sobre sus interacciones y el resultado de sus evaluaciones. (32)

IMS LD permite describir y empaquetar materiales de aprendizaje; agregar toda la información necesaria para asegurar la portabilidad de los OA a través de la *web* y su posterior reestructuración; crear materiales instruccionales que puedan interoperar a través de herramientas de autor, *LMS* y entornos para su ejecución que hayan sido desarrollados independientemente por varios desarrolladores. (33)

SCORM se desarrolló sobre la base de iniciativas anteriores, como el sistema de descripción de cursos en *XML* de la *IMS* y el mecanismo de intercambio de información de la *AICC*, para luego formar el estándar *SCORM*. Es un conjunto de especificaciones y estándares elaborados por distintos organismos que se postula como el modelo común para los OA. Su propósito es unificar esfuerzos entre grupos con intereses similares, creando un modelo de referencia que permita coordinar las tecnologías emergentes y las implementaciones comerciales de las mismas. Establece un empaquetamiento que permite establecer un orden jerárquico entre esos elementos. Posee una detalladamente documentación, resumida en 4 libros. Según desarrolladores de software en la Cátedra *Concytec UNAS*, "...*SCORM* es el estándar más usado en lo que se refiere a *e-learning*...". (34)

Tin Can API permite crear actividades, grabarlas y registrarlas por medio de los *LRS*⁹. Un *LRS* permite identificar quién, cómo y qué ha realizado una actividad. De este modo se consigue independencia de los *LMS*, puesto que pueden producirse fuera de cualquier plataforma. El aprendizaje puede ocurrir en cualquier lugar, no solamente en cursos basados en el estándar *SCORM*. (33)

Lo que hace tan útiles este conjunto de estándares es que, además de los metadatos, los métodos para empaquetar los contenidos se realizan en formato *ZIP*, que además incluyen los protocolos de comunicación entre el OA creado y el sistema de gestión de aprendizaje donde va a ser publicado. (34)

En resumen, ante la existencia de una inmensa cantidad de contenidos educativos, muchos de ellos abiertos, surge la necesidad de utilizar tecnologías que permitan un acceso fácil y rápido a dichos contenidos. El hecho de que los contenidos intenten seguir estándares de creación facilita su lectura con tecnologías específicas, sobre todo si estas también son de acceso libre. Siendo *SCORM*, uno de los estándares

⁹ *LRS: Learning Record Store* (Tienda de Reproducción de Aprendizaje).

Visor de paquetes de contenido educativo para dispositivos móviles

para recursos educativos más conocidos y utilizados internacionalmente. Además, es el estándar utilizado en el desarrollo de objetos de aprendizaje en la Universidad.

1.1.5. Estándar SCORM.

El *Shareable Content Object Reference Model*, más conocido por sus siglas, *SCORM*, "...es un conjunto de estándares y especificaciones de empaquetamiento muy utilizado que define la interrelación de los objetos de contenido, modelos de datos y protocolos permitiendo presentar material didáctico estructurado de forma adecuada. Este estándar fue desarrollado por *ADL*, el cual es un programa para el desarrollo e implementación de la formación educativa sobre la tecnología web. *SCORM* es resultado de la unión del sistema de descripción de cursos en *XML* de la *IMS*, y el mecanismo de intercambio de información mediante una *API* de la *AICC*...". (35) Por lo que se puede afirmar que, la especificación *SCORM* logra combinar los elementos de *IEEE*, *AICC* e *IMS* en un único documento consolidado de fácil implementación.

"Las características de *SCORM* se pueden resumir en tres líneas principales: una especificación basada en *XML* para representar la estructura de los cursos, logrando portabilidad de los cursos entre distintos *LMS*; un conjunto de especificaciones relacionadas al ambiente de ejecución, que incluye una *API*, un modelo de datos para la comunicación entre el *LMS* y los contenidos, y una especificación para el lanzamiento de los contenidos; y una especificación para la creación de registros que contienen metadatos del contenido". (32)

La especificación *SCORM* está dividida en libros técnicos, Estos libros se agrupan en 3 temas principales: Modelo de agregación de contenidos (*Content Aggregation Model*); Entorno de ejecución (*Run-Time Environment*); Secuenciación y navegación (*Sequencing and Navigation*). (36)

Yenny Paola Acosta Calvo y Maria E. Ráquira Numpaque, de la Facultad de Ingeniería de la Universidad Distrital Francisco José de Caldas, afirman que un contenido cumple con *SCORM* si está:

- ✓ Diseñado para ser exhibido en un navegador.
- ✓ Descrito por metadatos.
- ✓ Organizado como un conjunto estructurado de objetos más pequeños.
- ✓ Empaquetado de tal manera de que pueda ser importado por cualquier plataforma *SCORM* compatible.

Visor de paquetes de contenido educativo para dispositivos móviles

- ✓ Creado para ser portable, de forma que pueda ser distribuido por cualquier servidor web en cualquier sistema operativo. (37)

Un paquete con estructura *SCORM* es un archivo en formato *PKZip* v2.04g (.zip), que en el entorno *SCORM* es llamado *PIF* (*Package Interchange File*) que utiliza de forma estricta la especificación *IMS Content Packaging Specification*. Gracias a esto es posible el intercambio entre diferentes *LMS*. Posee 3 versiones:

- ✓ **SCORM 1.0:** La primera versión, *SCORM 1.0*, se remonta al año 2000. La estructura para describir el contenido se basaba en especificaciones *AICC* mediante un archivo *XML*, sin embargo, fue reemplazado en poco tiempo debido a que no presentaba un buen manifiesto o soporte para metadatos.
- ✓ **SCORM 1.2:** Lanzado en 2001, *SCORM 1.2* es la versión previa a la integración de la secuenciación. *SCORM 1.2*. Fue la primera versión con una prueba real de conformidad mediante una serie de pruebas y utiliza la especificación *IMS* de empaquetamiento de contenido, de esta forma describen el curso con una precisión mucho mayor.
- ✓ **SCORM 2004 (1.3):** *SCORM 2004* permite compartir y utilizar la información sobre el estado de éxito de múltiples objetivos de aprendizaje o competencias a través de objetos de contenido y cursos para un alumno concreto dentro de un mismo sistema de gestión de aprendizaje. (38)

El paquete *SCORM* contiene dos componentes principales: un documento *XML* especial con el nombre *imsmanifest.xml*, que describe la estructura del contenido y los recursos asociados al mismo, dígame también metadatos; y el contenido, que son los archivos físicos que componen el paquete de contenido. (39) Además, se rige por elementos que constituyen una entidad de formación, los cuales se definen como:

- ✓ **Asset:** Conjunto de recursos básicos de aprendizaje que se pueden observar mediante un navegador *web*. Tales como: textos, objetos de evaluación, videos, imágenes, audio, paginas *HTML*.
- ✓ **SCO¹⁰:** Es un objeto de aprendizaje que contiene uno o más *assets*, lo cual le permite comunicarse con un *LMS* en tiempo de ejecución real. Es la unidad mínima de contenido, que contiene código

¹⁰ *SCO: Sharable Content Object* (Contenido de Objeto Intercambiable).

Visor de paquetes de contenido educativo para dispositivos móviles

JavaScript, con el fin de comunicarse con los *LMS*. Estos objetos se unen en organizaciones dentro del manifiesto, definiéndose clúster como un conjunto de *SCO*.

- ✓ **Activity (Actividad):** Representa una acción realizada por el alumno, el cual desencadena un pedido al *LMS* de objetos de aprendizajes o *assets*. Para el *LMS*, las actividades son los contenidos que se encuentran en el menú de la unidad educativa que se muestra en el curso. Este conjunto de actividades se encuentra estructurada en forma de árbol evidenciándose la jerarquía que poseen las mismas.
- ✓ **Content Organization (Organización del contenido):** Es una representación jerárquica que está compuesto por un *SCO* o un clúster, descritos por metadatos que permiten el uso de los contenidos a través de las unidades estructuradas de enseñanza, es decir las actividades, mostrando como se relacionan entre sí.
- ✓ **Content Aggregation (Agregación de contenido):** Se utiliza para describir una entidad conceptual creada como parte del proceso, el contenido de un paquete, así como, la estructura de los mismos, para luego ser almacenados en un repositorio y ser transferidos entre estos sistemas. (39)

A continuación, se muestra una figura donde se evidencia la organización y agregación de contenidos, así como el contenido de objeto intercambiable y el *asset*.

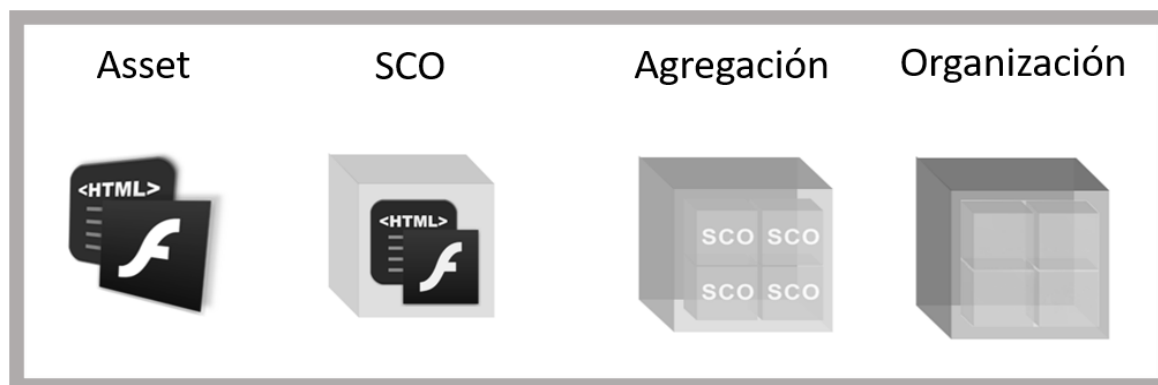


Figura 1: Organización y agregación de contenidos, *SCO* y *Asset*. Elaboración propia.

El *XML* principal se declara con la etiqueta `<manifest>`, y describe la información que un *LMS* necesita para organizar y mostrar los contenidos de forma adecuada. Teniendo la siguiente estructura:

Visor de paquetes de contenido educativo para dispositivos móviles

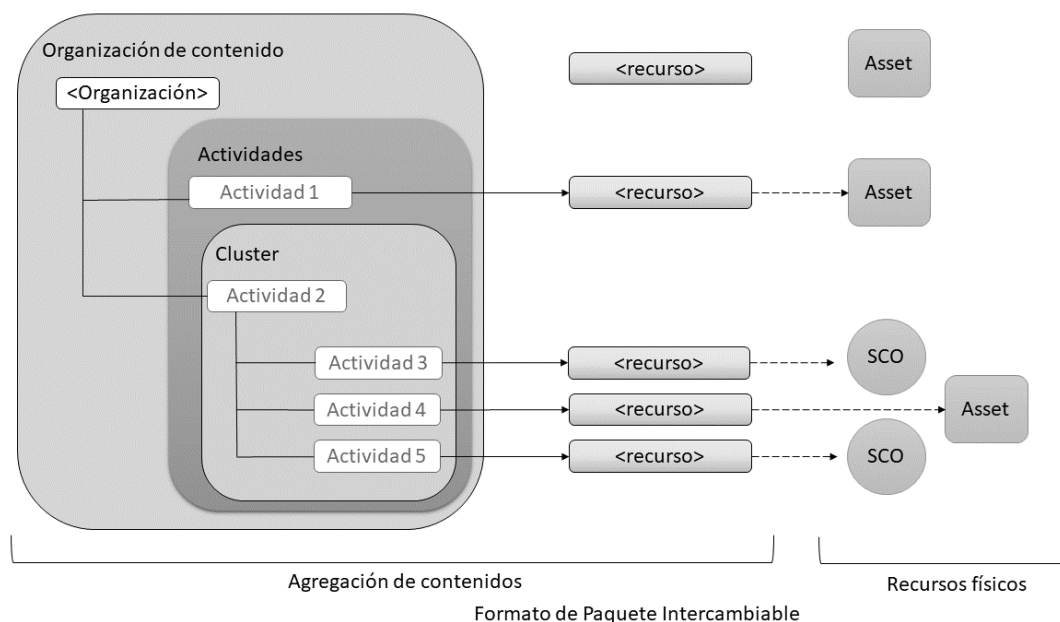


Figura 2: Organización de la estructura del *imsmanifest.xml* de un paquete SCORM. Elaboración propia.

Compuesta por los siguientes elementos:

- ✓ **Metadatos:** Describen los objetos de aprendizaje, especifican datos como la versión del estándar que se utiliza, los objetivos y los recursos didácticos.
- ✓ **Organizaciones (*Organizations*):** Constituye la estructura de los recursos de aprendizaje y las actividades que en ellas se encuentran, constituyendo una o varias unidades de instrucciones independientes de secuenciación y navegación, enlazadas con los recursos mediante su identificador.
- ✓ **Recursos (*Resources*):** Contienen los cursos de aprendizaje que se encuentran agrupados en el paquete de contenido, que estos a su vez, deben ser un objeto de contenido intercambiable para poder relacionarse con las diferentes plataformas.
- ✓ **Sub-manifiestos:** Contienen uno o varios manifiestos, los cuales describen cualquier unidad lógica de instrucción anidada, que pueden ser tratadas como unidades independientes. (14)

Los principales requerimientos que el modelo SCORM trata de satisfacer son:

- ✓ **Accesibilidad:** capacidad de acceder a los componentes de enseñanza desde un sitio distante a través de las tecnologías web, así como distribuirlos a otros sitios.

Visor de paquetes de contenido educativo para dispositivos móviles

- ✓ **Adaptabilidad:** capacidad de personalizar la formación en función de las necesidades de las personas y organizaciones.
- ✓ **Durabilidad:** capacidad de resistir a la evolución de la tecnología sin necesitar una re-concepción, una re-configuración o una re-escritura del código.
- ✓ **Interoperabilidad:** capacidad de utilizarse en otro emplazamiento y con otro conjunto de herramientas o sobre otra plataforma de componentes de enseñanza desarrolladas dentro de un sitio, con un cierto conjunto de herramientas o sobre una cierta plataforma. Existen numerosos niveles de interoperabilidad.
- ✓ **Re-usabilidad:** flexibilidad que permite integrar componentes de enseñanza dentro de múltiples contextos y aplicaciones. (14)

1.1.6. Android.

Android es el sistema operativo para dispositivos móviles más popular del mundo; desde dispositivos móviles y relojes, hasta autos y televisores; basadas en las contribuciones de la comunidad Linux, por lo que es una plataforma de desarrollo libre y de código abierto. Este se utiliza en teléfonos inteligentes, ordenadores portátiles, *notebooks*, *tablets*, *Google TV*, relojes de pulsera, auriculares y otros. Debido a que está basada en software libre, su arquitectura está dividida en capas. *Android* ofrece a los desarrolladores herramientas para crear aplicaciones que se adaptan a las capacidades disponibles en cada dispositivo. (40)

1.2. Estudio de aplicaciones similares.

A continuación, se presenta un conjunto de aplicaciones que permiten visualizar paquetes de contenido educativo en formato *SCORM* para dispositivos móviles.

1.2.1. Moodle Mobile.

Moodle Mobile es una aplicación oficial del sistema de gestión de aprendizaje *Moodle* para dispositivos móviles de hasta 5 millones de instalaciones y la última actualización fue el 29 de septiembre de 2017 desde *Google Play*. Esta aplicación permite observar el contenido de los cursos incluso, cuando esta *offline*; recibir notificaciones instantáneas de mensajes y otros eventos, encontrar fácilmente a otras personas que también están en un curso común; subir imágenes, videos, audios y otros archivos desde su

Visor de paquetes de contenido educativo para dispositivos móviles

dispositivo móvil; ver las calificaciones del curso, entre otras. Permite visualizar varias versiones de SCORM y verifica la estructura del fichero *imsmanifest.xml*. (14)

1.2.2. SCORM Player.

Scorm Player es una aplicación que se puede usar sin depender de una constante conexión a *internet*. Solicita los materiales educativos públicos y gratuitos de *scormeditor.com*, por lo que en ese instante necesitara conexión a *internet*. Permite visualizar varias versiones de SCORM y verifica la estructura del fichero *imsmanifest.xml*. Requiere tener instalada a partir de la versión 2.2 de *Android*. Esta aplicación ha tenido alrededor de 5 mil descargas desde *Google Play*. (41)

1.2.3. G-OpenLMS.

G-openLMS es una aplicación móvil para el acceso *LMS Sakai*, el usuario puede realizar todas las funcionalidades como *Sakai* sin utilizar el navegador. Por esta razón, depende de un *LMS* y necesita conexión a *internet* para el correcto funcionamiento de la aplicación. Permite visualizar varias versiones de SCORM y verifica la estructura del fichero *imsmanifest.xml*. Posee hasta 5 mil descargas desde *Google Play*. Requiere tener instalada a partir de la versión 4.0 de *Android*. La última versión liberada fue el 19 de julio de 2017. (42)

1.2.4. Blackboard Mobile.

Blackboard es una aplicación para dispositivos móviles, la cual utiliza un servidor y una base de datos independientes para el manejo del sistema. Se conecta directamente al servidor *Blackboard Learn*. Esta aplicación modifica la visualización de la interfaz *web* de Blackboard Learning System, adaptándola a las características y tamaño del dispositivo. Permite visualizar varias versiones de SCORM y verifica la estructura del fichero *imsmanifest.xml*.

Contiene alrededor de 4 233 usuarios y hasta 5 millones de descargas desde *Google Play*. Requiere tener instalada a partir de la versión 4.2 de *Android*. Esta aplicación incluye *Blackboard Collaborate* con la experiencia Ultra, una solución de conferencias *web* síncrona para clases y reuniones virtuales de alta calidad. Permite ver actualizaciones de cursos y contenidos rápidamente; realizar actividades y pruebas; ver calificaciones para cursos, actividades; entre otras. (43)

Visor de paquetes de contenido educativo para dispositivos móviles

1.2.5. TeachByApp SCORM Player.

TeachByApp SCORM Player es una aplicación que permite reproducir lecciones *SCORM* sin necesidad de estar conectado a *internet*. Es compatible con *SCORM-2004*, por lo que verifica la estructura del *imsmanifest.xml*. Contiene un número limitado de clases y brinda una lección de bonificación por parte de *e-levelcom.com*. Por lo tanto, a pesar de que la aplicación es gratuita, es necesario efectuar el pago para adquirir las lecciones. Para adquirir esas lecciones, la aplicación tiene que establecer conexión a *internet*. Soporta la secuencia y navegación de los contenidos. Esta aplicación ha tenido alrededor de 100 descargas desde *Google Play*. Requiere tener instalada a partir de la versión 4.0.3 de *Android*. (44)

1.2.6. SCORM Reader.

SCORM Reader es una aplicación para dispositivos móviles, específicamente *tablets*, con sistema operativo *Android*, lector de cursos educativos y de Objetos de Aprendizaje que se encuentren empaquetados bajo el estándar *SCORM*. La aplicación localiza el recurso desde la memoria del dispositivo o *Dropbox*, evidenciándose la necesidad de conectarse a *internet*.

Esta aplicación no depende de ningún *LMS* específico. Permite, además, su publicación en las redes sociales ofreciendo al usuario la posibilidad de justificar sus progresos. Está basada en *ADL*. Soporta la secuencia y navegación de los contenidos. Es distribuible, pero para ello es necesario efectuar el pago. (45)

1.2.7. SCORMView.

SCORMView es una aplicación móvil para el sistema operativo *Android*, desarrollada en el centro FORTES en la UCI, con el objetivo de proveer a XauceMóvil de un componente que permita interpretar y visualizar paquetes *SCORM*. Permite importar los paquetes bajo el estándar *SCORM* tanto desde el propio dispositivo móvil como desde los servidores que XauceMóvil le da soporte. (14)

Ofrece también la navegación por cada uno de los contenidos mediante un índice. Permite realizar una búsqueda de Objetos de Aprendizaje en RHODA y descargarlos hacia el dispositivo a través del consumo de los servicios web que ofrece este repositorio. Esta aplicación no reproduce los archivos de audios ni videos. No guarda el avance de los usuarios en los cursos empaquetados. Para la realización de algunas de sus funcionalidades es necesario tener conexión a *internet*. (14)

Visor de paquetes de contenido educativo para dispositivos móviles

1.2.8. Conclusiones del estudio de las aplicaciones similares existentes.

A continuación, se muestra una tabla comparativa, resultado del estudio de los visores de contenido educativo en formato *SCORM* para dispositivos móviles, explicados anteriormente. Para ello, la autora considera que los parámetros a tener en cuenta son las características en común que puedan darle solución al problema de investigación planteado. Los parámetros están confeccionados por las ventajas y desventajas de estas aplicaciones, lo que posibilita un mejor análisis de estas debido a que existe una mayor organización de los datos.

Tabla 1: Comparación entre visores de contenido educativo en formato SCORM para dispositivos móviles. Elaboración propia.

Aplicaciones/ Características	Moodle Mobile	SCORM Player	G- OpenLM S	Blackboard Mobile	TeachByApp SCORM Player	SCORM Reader	SCORMView
Visualiza varias versiones de SCORM.	Sí	Sí	Sí	Sí	No, solo con SCORM-2004	Sí	Sí
Visualiza cursos y objetos de aprendizaje.	Sí	Sí	Sí	Sí	Sí	Sí	Sí, pero no reproduce archivos de audio ni videos.
Soporta secuencia y navegación de los contenidos.	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Verifica la estructura <i>imsmanifest.xml</i>.	Sí	Sí	Sí	Sí	Sí	Sí	Sí

Visor de paquetes de contenido educativo para dispositivos móviles

Efectúa el pago	No	No	No	No	Si	Sí	No
Depende de un LMS o repositorio educativo	Sí	Sí	Sí	Sí	Sí	No	Sí
Contiene un número limitado de paquetes	No	No	No	No	Sí	No	No
Necesita conexión a Internet	Sí	Sí	Sí	Sí	No, (solo para obtener la lección de bonificación)	No	Sí

A partir del análisis anterior, se puede observar que los visores de contenido educativo con estándar SCORM para dispositivos móviles con sistema operativo *Android*, precisan de conexión a Internet para su funcionamiento, dependen de un LMS específico o de un repositorio educativo. Debido a estas desventajas, no puede emplearse ninguna de estas aplicaciones como solución al problema de la investigación. Teniendo en cuenta lo expresado anteriormente, se decide desarrollar un Visor de Paquetes de Contenido Educativo con estándar SCORM para dispositivos móviles.

1.3. Metodología de desarrollo

Se define como metodología, un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo *software*. La metodología sigue el ciclo de vida del proceso de desarrollo del software, indicando la manera de obtención de los productos, ya sean parciales o finales. Tiene como principal objetivo aumentar la calidad del software que se produce.

En el presente proceso de desarrollo de software se utilizará la variación AUP-UCI, una variante de la metodología Proceso Unificado Ágil, (más conocido por sus siglas en inglés AUP); debido a que es la metodología por la que se rigen los proyectos productivos en la UCI. Esta variante une las particularidades de AUP, disminuyendo el esfuerzo en tiempo y cantidad de personas que converjan en este proyecto; adaptada al ciclo de vida de los proyectos que se realizan en la Universidad. (46)

Visor de paquetes de contenido educativo para dispositivos móviles

Además, cuenta con cuatro escenarios para modelar el sistema en los proyectos. De ellos, se utilizará el cuarto escenario debido a que los procesos del negocio del sistema no se modelan, porque no es un proyecto muy extenso y el cliente estará acompañando a la autora a tiempo parcialmente completo. Este escenario permitirá encapsular los requisitos mediante las historias de usuario (HU), definiendo la información necesaria para cada requisito. (46)

1.4. Tecnologías

Las herramientas y tecnologías son usadas en el proceso de desarrollo del software, unificadas a una metodología adecuada, con el fin de obtener un producto con la calidad requerida.

1.4.1. Lenguaje de modelado

UML v2.0

El Lenguaje de Modelado Unificado, más conocido por sus siglas en inglés *UML (Unified Modeling Language)* ayuda a especificar, visualizar y documentar modelos de sistemas de *software*, incluyendo su estructura y diseño, de una manera que cumpla con todos estos requisitos. También puede usar *UML* para modelar y modelar negocios de otros sistemas que no son de *software*. Puede modelar casi cualquier tipo de aplicación, ejecutándose en cualquier tipo y combinación de hardware, sistema operativo, lenguaje de programación y red. Su flexibilidad le permite modelar aplicaciones distribuidas que usan casi cualquier middleware en el mercado. Basado en conceptos fundamentales de orientado a objeto que incluyen clase y operación, es un complemento natural para los lenguajes y entornos orientados a objetos como C++, *Java* y el C# reciente. (47)

1.4.2. Herramienta de modelado

Visual Paradigm v8.0

El *Visual Paradigm* es una herramienta de Ingeniería de *Software* Asistida por Computación (CASE), que propicia un conjunto de ayudas para el desarrollo de *software*, desde el análisis y diseño, hasta la generación del código fuente del *software* y la documentación. Ha sido concebido para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Permite administrar la complejidad de la transformación empresarial, ayuda en la planificación empresarial, en la gestión de proyectos y en el desarrollo de software ágil, en general. Disponible en múltiples plataformas (*Windows*, *Linux*). Diseño centrado en casos de uso y enfocado al negocio que genera un

Visor de paquetes de contenido educativo para dispositivos móviles

software de mayor calidad. Posee capacidades de ingeniería directa e inversa. Con licencia gratuita y comercial. (48)

1.4.3. Lenguaje de programación

Java v.1.8

Java 8 es una versión de Java que incluye características, mejoras y correcciones de bugs, comparada con la versión anterior, con el fin de mejorar la eficacia en el desarrollo y la ejecución de programas *Java*. *Java* es un lenguaje de programación orientado a objeto, utilizada para desarrollar aplicaciones de red, además del estándar global para desarrollar y distribuir aplicaciones móviles y embebidas, juegos, contenido basado en *web* y *software* de empresa. Con más de 9 millones de desarrolladores en todo el mundo. Algunas de las mejoras que ofrece esta versión de *Java* son: la implantación de expresiones Lambda y funciones adyacentes a la plataforma, la creación de un *API* de fecha y hora, la cual permite gestionar datos de hora y fecha de una forma más natural y fácil de comprender; y la seguridad mejorada. (49)

1.4.4. Herramientas y tecnologías de desarrollo

Gradle

Gradle es un sistema de compilación que reúne en un solo las mejores prestaciones de otros sistemas de compilación. Está basado en JVM (Java Virtual Machine), lo que significa que puedes escribir tu propio script en java. Gradle es un plugin, lo que facilita su actualización y su exportación de un proyecto a otro. Esto significa que se puede tener su propio lenguaje de programación y automatizar el proceso de compilación en un solo paquete (de la misma manera que un *.jar* en caso de java) y poder distribuirlo al resto del mundo. Permite reutilizar fácilmente código, realizar de una manera más sencilla la tarea de configurar y personalizar la compilación, gestionar las dependencias de forma potente y cómoda, la compilación desde consola, lo que facilita la compilación en sistemas sin el entorno de desarrollo montado y proporciona la creación de diferentes versiones de la aplicación. (50)

Genymotion

Es un emulador para *Android* que incluye un conjunto de sensores y características con el fin de facilitar la interacción con un entorno virtual de *Android*. Permite probar sus aplicaciones de *Android* en una amplia gama de dispositivos virtuales para fines de desarrollo, prueba y demostración. Esta herramienta

Visor de paquetes de contenido educativo para dispositivos móviles

es rápida, fácil de instalar, y potente gracias a los *widgets* de sensor fáciles de usar y las funciones de interacción. Está disponible para sistemas operativos *Windows*, *MacOS* y *Linux*. (51)

Android Studio

Android Studio es el entorno de desarrollo integrado (*IDE*) oficial para el desarrollo de aplicaciones para *Android* y se basa en *IntelliJ IDEA*. Además del potente editor de códigos y las herramientas para desarrolladores de *IntelliJ*, *Android Studio* ofrece aún más funciones que aumentan tu productividad durante la compilación de aplicación para *Android*, como un sistema de compilación basado en *Gradle* flexible; un emulador rápido con varias funciones; un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos *Android*; *Instant Run* para aplicar cambios mientras la aplicación se ejecuta sin la necesidad de compilar un nuevo APK; integración de plantillas de código y *GitHub* para ayudarte a compilar funciones comunes de las aplicaciones e importar ejemplos de código; gran cantidad de herramientas y *frameworks* de prueba; herramientas *Lint* para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, entre otros; compatibilidad con C++ y *NDK*; soporte incorporado para *Google Cloud Platform*, lo que facilita la integración de *Google Cloud Messaging* y *App Engine*. (52)

1.4.5. Herramienta para el diseño de prototipo de interfaz

Balsamiq Mockups v2.1.13

“Es una herramienta que reproduce la experiencia de realizar bocetos en una pizarra, a través de un ordenador. Permite el diseño de aplicaciones de escritorio, para la web y orientadas a dispositivos móviles. Cuenta con gran cantidad de componentes visuales para la construcción de interfaces de usuario, íconos y elementos reutilizables tales como plantillas. Incluye la funcionalidad de exportar a los formatos *PDF* y *PNG*, y una interfaz sencilla que posibilita el diseño de prototipos simplemente arrastrando componentes hacia una pantalla. Cuenta con soporte para los sistemas operativos *Windows* y *MacOS*, *Linux* no lo soporta, pero el ejecutable de *Windows* (.exe) puede emularse a través de la aplicación *Wine*.” (53)

1.4.6. Herramienta de pruebas

JUnit

Es un marco de trabajo creado para realizar pruebas unitarias a aplicaciones que empleen el lenguaje de programación *Java*. Incluye formas de ver los resultados de las pruebas ya sea en forma de texto, gráfico o como tarea. Incluido a través de *plugins* en los principales *IDEs* como son *NetBeans*, *Eclipse* y

Visor de paquetes de contenido educativo para dispositivos móviles

Android Studio. De código abierto, puede integrarse con *Maven*, emplea anotaciones y condiciones de aceptación para facilitar el trabajo. (54)

1.5. Valoraciones finales

A partir del desarrollo del presente capítulo se arribó a las siguientes conclusiones:

- ✓ El uso del estándar *SCORM* para el empaquetado de los objetos de aprendizaje, brinda una estructura adecuada para ser utilizados en diversas plataformas.
- ✓ El estudio del estado del arte permitió concluir que la existencia de aplicaciones estudiadas es escasa, y las existentes no brindan una solución completa al problema a resolver. Por lo que se consideró desarrollar un visor de contenido educativo estandarizado por *SCORM*, para dispositivos móviles con sistema operativo *Android*.
- ✓ Para guiar el proceso de desarrollo de la aplicación, se utilizará la metodología *AUP-UCI*; y para el modelo del sistema, el escenario 4.
- ✓ Las herramientas y las tecnologías que se utilizaran, son las adecuadas para el desarrollo de aplicaciones para dispositivos móviles con sistema operativo *Android*.

Capítulo 2. Análisis y diseño de la propuesta de solución.

Siguiendo la metodología *AUP-UCI*, se desea obtener un producto con buena calidad, teniendo en cuenta las prácticas que dicho documento plantea. Para ello es necesario describir las características y el diseño de la propuesta de solución a desarrollar. Con el fin de identificar las necesidades del cliente, es preciso especificar los requisitos funcionales y no funcionales; detallándolos mediante historias de usuarios correspondientes a estos. Además, para ayudar a una mejor comprensión de la solución, se deben exponer diagramas de clase, patrones de diseño y arquitectura de la aplicación que la autora estime conveniente. Debido a ello, en el presente capítulo se muestran los resultados recogidos en las fases de inicio y ejecución planteadas por dicha metodología.

2.1. Propuesta de solución

En la presente investigación se propone como solución, el desarrollo de una aplicación que visualice e interprete los paquetes de contenidos educativos para dispositivos móviles que tenga como sistema operativo *Android* con versiones iguales o superiores a 4.0 (*Ice Cream Sandwich*). Tiene como objetivo principal desarrollar el aprendizaje móvil en escenarios conectividad parcial y sin conectividad. Esta solución permitirá que los estudiantes de educación a distancia puedan acceder a los contenidos y recursos educativos que se encuentren en sistemas de gestión de aprendizajes y repositorios educativos.

Para una mayor organización de los paquetes, al iniciarse la aplicación por primera vez, se le solicita al usuario que seleccione un directorio fijo dentro del dispositivo móvil, donde se almacenaran los contenidos educativos. Que luego puede ser modificada a conveniencia del usuario. La aplicación permite importar los paquetes que se encuentren bajo el estándar SCORM, ubicados en el propio dispositivo y desde una url especificada por el usuario; y visualizar en una lista, los paquetes ya importados. Para ello, le brinda al usuario dos maneras de importarlos:

- 1 Si el paquete esta comprimido. La aplicación le pide la ubicación al usuario, para luego ser descomprimido en el directorio escogido anteriormente por el usuario.
- 2 Si el paquete ya está descomprimido. La aplicación copia el paquete descomprimido dentro del directorio escogido anteriormente por el usuario.

Visor de paquetes de contenido educativo para dispositivos móviles

Para la segunda opción, la aplicación descarga el paquete desde la url especificada por el usuario y luego, lo importa mediante las maneras anteriormente explicadas. Para realizar esta acción, la url debe ser pública, de lo contrario, la aplicación no podrá descargar el paquete requerido.

Además, posibilita buscar y eliminar uno o varios paquetes específicos dentro de la lista de paquetes importados; así como, ordenarlos alfabéticamente. Este visor muestra un menú que contiene un índice de contenidos, el cual permite la navegación por los recursos educativos que posee el paquete, y las opciones de ir hacia adelante y hacia atrás sin necesidad de ir al índice de contenido de nuevo.

Para mejor comprensión, a continuación, se muestra una imagen describiendo la propuesta de solución explicada anteriormente.

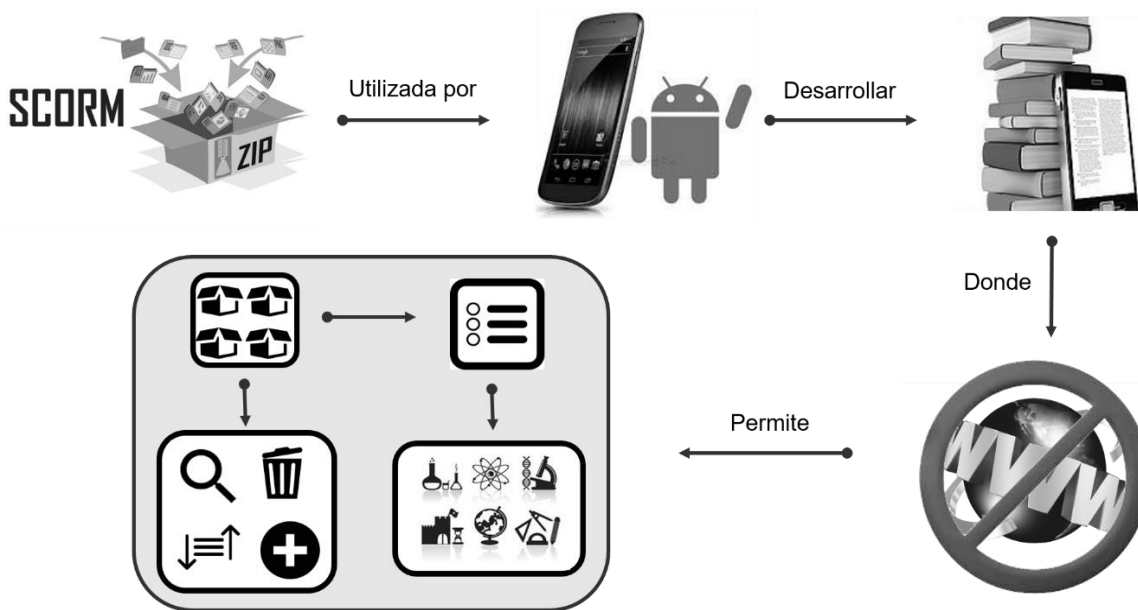


Figura 3: Descripción de la propuesta de solución. Elaboración propia.

2.2. Captura de requisitos

Los requisitos son funcionalidades que posee el sistema con el fin de cumplir un objetivo o darle solución a un problema determinado. Es por ello, que la captura de requisitos es un paso de gran importancia durante el desarrollo de un *software*. Esta labor es necesaria para identificar las necesidades y especificaciones del cliente. Para ello se identifican los requisitos funcionales y no funcionales de la aplicación, a través de uno de los métodos de investigación empíricos como es la entrevista con el cliente.

Visor de paquetes de contenido educativo para dispositivos móviles

2.2.1. Requisitos funcionales

Los requisitos funcionales “son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares”. (55) Con el propósito de lograr el correcto funcionamiento de la solución antes propuesta, a continuación, se exponen los requisitos funcionales del sistema:

Tabla 2: Descripción de los requisitos funcionales. Elaboración propia.

No.	Requisito funcional (RF)	Descripción	Complejidad	Prioridad para el cliente
1	Listar paquetes SCORM importados.	La aplicación debe mostrar al usuario la lista de paquetes SCORM que se encuentren dentro del directorio de almacenamiento determinado por el usuario en la aplicación.	Media	Alta
2	Importar paquete SCORM desde el dispositivo.	La aplicación debe mostrar al usuario una vista donde podrá seleccionar el formato en que se encuentra el paquete que se desee importar. El paquete SCORM puede estar comprimido, como <i>Zip</i> ; o descomprimido, como una carpeta. La vista estará conformada de la siguiente manera: <ul style="list-style-type: none">• Ícono.• Opción.	Alta	Alta

Visor de paquetes de contenido educativo para dispositivos móviles

3	Mostrar paquete SCORM importado.	La aplicación mostrará una vista conformada por los siguientes datos del paquete: <ul style="list-style-type: none"> • Ícono. • Nombre. 	Alta	Alta
4	Mostrar índice de contenidos del paquete SCORM.	La aplicación debe mostrar al usuario una vista en forma de menú, el cual contendrá los siguientes elementos: <ul style="list-style-type: none"> • Logo. • Nombre. • Índice de contenidos. 	Alta	Alta
5	Mostrar contenidos del paquete SCORM.	La aplicación debe mostrar al usuario una vista con los objetos de aprendizaje relacionados con el contenido anteriormente seleccionado.	Media	Alta
6	Seleccionar directorio de almacenamiento de paquetes SCORM.	La aplicación debe mostrar una vista que contiene un explorador de archivos con la posibilidad de que el usuario pueda seleccionar el directorio que desee.	Baja	Media

Visor de paquetes de contenido educativo para dispositivos móviles

7	Modificar directorio de almacenamiento de paquetes <i>SCORM</i> .	La aplicación debe mostrar una vista con un explorador de archivo que le posibilitará al usuario seleccionar un directorio nuevamente.	Media	Media
8	Eliminar paquete <i>SCORM</i> de la aplicación.	La aplicación debe eliminar el o los paquetes seleccionados de la aplicación y del dispositivo.	Alta	Media
9	Ordenar listado de paquetes <i>SCORM</i> importados.	La aplicación debe mostrar el listado de paquetes ordenado alfabéticamente.	Baja	Baja
10	Buscar paquete <i>SCORM</i> de la lista de paquetes.	La aplicación debe buscar un paquete dentro del listado con respecto al nombre del mismo.	Baja	Baja
11	Mostrar resultado de la búsqueda del paquete <i>SCORM</i> de la lista de paquetes importados.	La aplicación debe mostrar una lista con los paquetes que contenga el nombre especificado por el usuario anteriormente.	Baja	Baja

Visor de paquetes de contenido educativo para dispositivos móviles

12	Importar paquete SCORM desde URL.	La aplicación debe mostrar una vista que le permita al usuario escribir la <i>Url</i> donde se ubica el paquete que desea importar. Esta vista se conforma de los siguientes elementos: <ul style="list-style-type: none">• Título.• Mensaje.• Url• Aceptar.• Cancelar.	Alta	Alta
----	-----------------------------------	---	------	------

2.2.2. Requisitos no funcionales

Los requisitos no funcionales son "...restricciones de los servicios o funciones ofrecidos por el sistema, que se les aplican a características o servicios individuales del sistema; se refieren a las propiedades emergentes del sistema como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento..." (56).

Portabilidad:

RNF 1: Dispositivo móvil con sistema operativo *Android* 4.1 o superior. Según *Android Developers*, esta versión de *Android* posee la distribución más alta con respecto al resto de las versiones. (57)

RNF 2: Memoria *RAM* del dispositivo con capacidad mínima de 256 *Mb*.

RNF 3: Almacenamiento interno o tarjeta micro *SD* con capacidad mínima de 10 *Mb*.

2.3. Historias de usuario

De acuerdo con la metodología empleada para el desarrollo de la solución propuesta se decide emplear las historias de usuario para el agrupamiento de los requisitos funcionales del sistema.

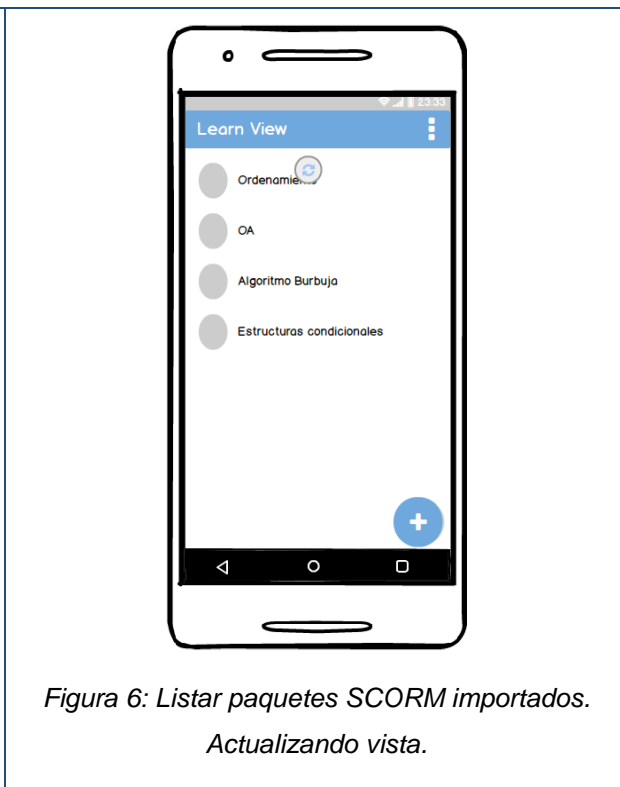
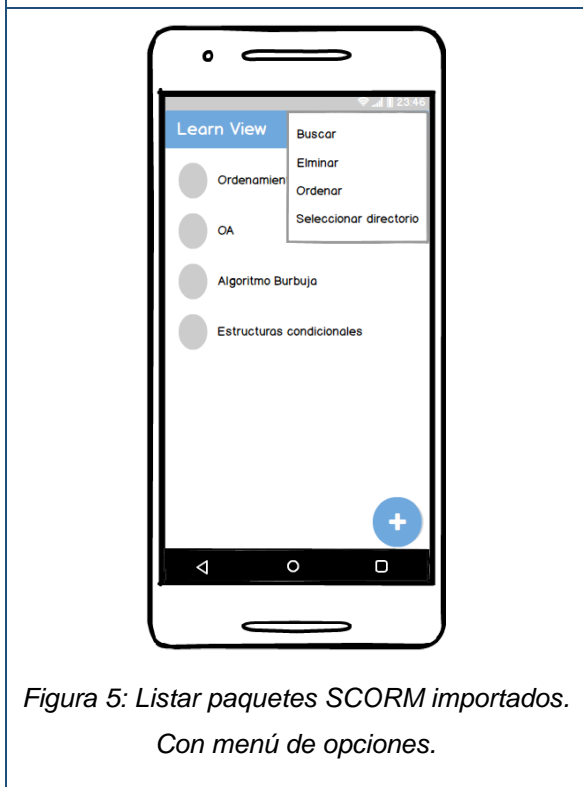
Visor de paquetes de contenido educativo para dispositivos móviles

Para la descripción de los requisitos funcionales de la propuesta de solución planteada, se define una historia de usuario por cada uno de ellos, resultando un total de 11 HU. A continuación, se muestran las HU “3” y “8”, debido a que son las funcionalidades de mayor prioridad para el cliente. El resto de las HU se encuentran en los anexos del documento.

Tabla 3: Historia de Usuario Listar paquetes SCORM importados. Elaboración propia.

Numero: 1	Nombre del requisito: Listar paquetes SCORM importados.	
Programador: Beatriz Ontivero González		Iteración asignada: 6
Prioridad: Alta		Tiempo estimado: 10 días
Riesgo en desarrollo: <ul style="list-style-type: none">✓ Interrupción en el fluido eléctrico.✓ Rotura de la estación de trabajo.✓ Configuración de la estación de trabajo.		Tiempo real: 7 días
Descripción: <p>La aplicación debe mostrar al usuario la lista de paquetes SCORM ya importados dentro de la aplicación. La funcionalidad tiene lugar en la interfaz principal de la aplicación.</p>		
Observaciones: <p>Si el usuario no selecciona un directorio de almacenamiento para los paquetes o no importa paquete alguno, la lista se mostrará vacía.</p>		
Prototipo de interfaz:		

Visor de paquetes de contenido educativo para dispositivos móviles



Visor de paquetes de contenido educativo para dispositivos móviles

Tabla 4: Historia de Usuario: Mostrar índice de contenidos del paquete SCORM. Elaboración propia.

Numero: 4	Nombre del requisito: Mostrar índice de contenidos del paquete SCORM.	
Programador: Beatriz Ontivero González		Iteración asignada: 1
Prioridad: Alta		Tiempo estimado: 20 días
Riesgo en desarrollo: <ul style="list-style-type: none">✓ Interrupción en el fluido eléctrico.✓ Rotura de la estación de trabajo.✓ Configuración de la estación de trabajo.		Tiempo real: 15 días
Descripción: <p>La aplicación debe mostrar al usuario una vista en forma de menú, el cual contendrá los siguientes elementos:</p> <ul style="list-style-type: none">• Logo: Muestra el logo de la aplicación.• Nombre: Muestra el nombre del paquete seleccionado.• Índice de contenidos: Muestra el índice de contenidos del paquete SCORM ya cargado en la aplicación. <p>Si el usuario selecciona un contenido del índice, la aplicación mostrará una vista con la información relacionada a dicho contenido.</p> <p>Si el usuario selecciona afuera del menú, la aplicación mostrará la información del contenido anteriormente mostrado. En caso de que el usuario no seleccionó previamente un elemento del índice, la aplicación mostrará el primer contenido del mismo.</p>		
Observaciones:		

Visor de paquetes de contenido educativo para dispositivos móviles

El índice de contenidos se mostrará, solamente, cuando se seleccione un paquete del listado de paquetes importados.

Prototipo de interfaz:

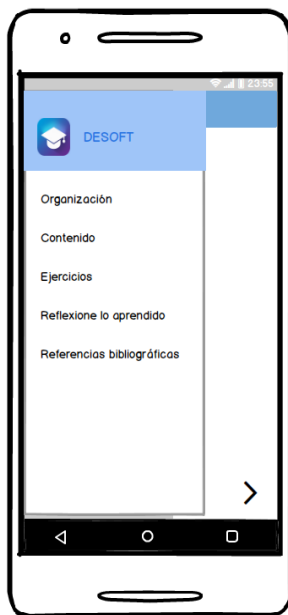


Figura 7: Índice de contenido educativo del paquete SCORM.

2.4. Diseño

Pressman expresó: “...el diseño de *software* se ubica en el área técnica de la ingeniería de software y se aplica sin importar el modelo del proceso que se utilice. El diseño del *software* comienza una vez que se han analizado y modelado los requerimientos, es la última acción de la ingeniería de software dentro de la actividad de modelado y prepara la etapa de construcción (generación y prueba de código)...”. (56)

2.4.1. Descripción de la arquitectura

La *IEEE* expresa que “...la arquitectura de *software* es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución...”. (58)

Visor de paquetes de contenido educativo para dispositivos móviles

La arquitectura del *software* representa los componentes del sistema, sus comportamientos, cómo interactúan entre ellos y como se encuentran estructurados los datos que utilizan; con el objetivo de mejorar la comprensión de dicho sistema.

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. (59) Para ello, MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador. El controlador determina con cual vista es mostrado. Puede haber múltiples vistas por controlador. Las acciones de los disparadores de la vista el controlador las recibe y este modifica el modelo o elige otra vista.

Por otro lado, el Modelo Vista Presentador (MVP) es un patrón donde, toda la lógica de la interfaz de usuario se hace en una clase separada, que se conoce como Presentador. Esta clase hace de intermediario entre la interfaz gráfica de usuario (Vista) y el modelo de datos. A diferencia del MVC, existe una correspondencia uno-a-uno entre la vista y el presentador asociado; el presentador actualiza la vista sobre la base de las medidas solicitadas que realiza en el modelo; posee dos maneras de comunicación con la vista:

1. La vista se comunica con el presentador directamente con llamadas a funciones de una instancia del presentador.
2. El presentador se comunica con la vista hablando con la interfaz de la vista. (60)

En el MVP, el modelo, es la capa donde se gestionan los datos de la aplicación, y la lógica para recuperar y conservar los datos. La vista es la responsable de mostrar la información al usuario. El usuario puede interactuar con los controles en la vista, pero cuando se necesita cierta lógica para iniciarse, la vista está delegado al presentador. El presentador tiene toda la lógica de la vista y es responsable de sincronizar el modelo y la vista. Cuando la vista notifica el presentador que el usuario ha interactuado con la interfaz, el presentador a continuación, actualiza el modelo y sincroniza los cambios entre el modelo y la vista. El presentador no comunica directamente a la vista. En lugar, se comunica a través de una interfaz. De esta forma, el presentador y el modelo se pueden probar en aislamiento. (60)

A continuación, se muestra a través de una figura, la comparación entre las arquitecturas MVC y MVP.

Visor de paquetes de contenido educativo para dispositivos móviles

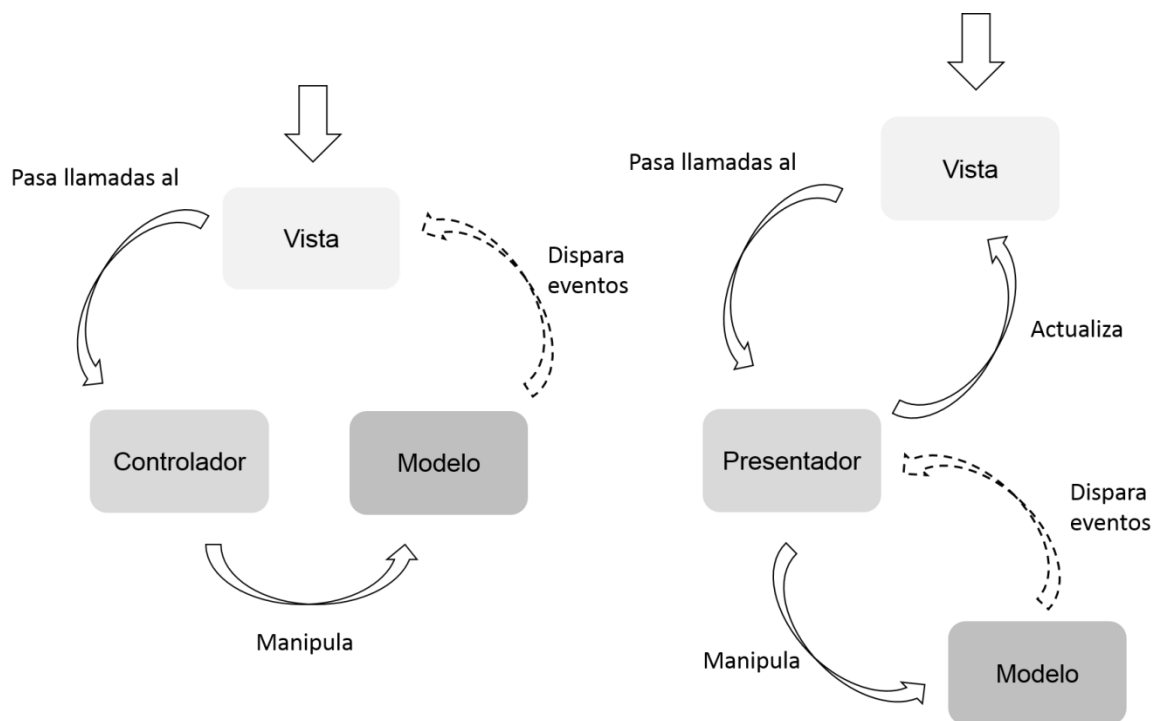


Figura 8: Comparación entre el Modelo Vista Controlador y el Modelo Vista Presentador. Elaboración propia.

Para el presente trabajo se propone una arquitectura Modelo-Vista-Presentador, ya que la autora considera que es el patrón arquitectónico más adecuado a utilizar para el diseño de la aplicación. Debido a que, en la aplicación se cumple que, para cada una de las vistas se comunica con una única clase controladora, en este caso, presentadora. Por tanto, la solución propuesta constará con tres componentes fundamentales: Modelo, Vista y Presentador. A continuación, se muestra una vista que lo ilustra:

Visor de paquetes de contenido educativo para dispositivos móviles

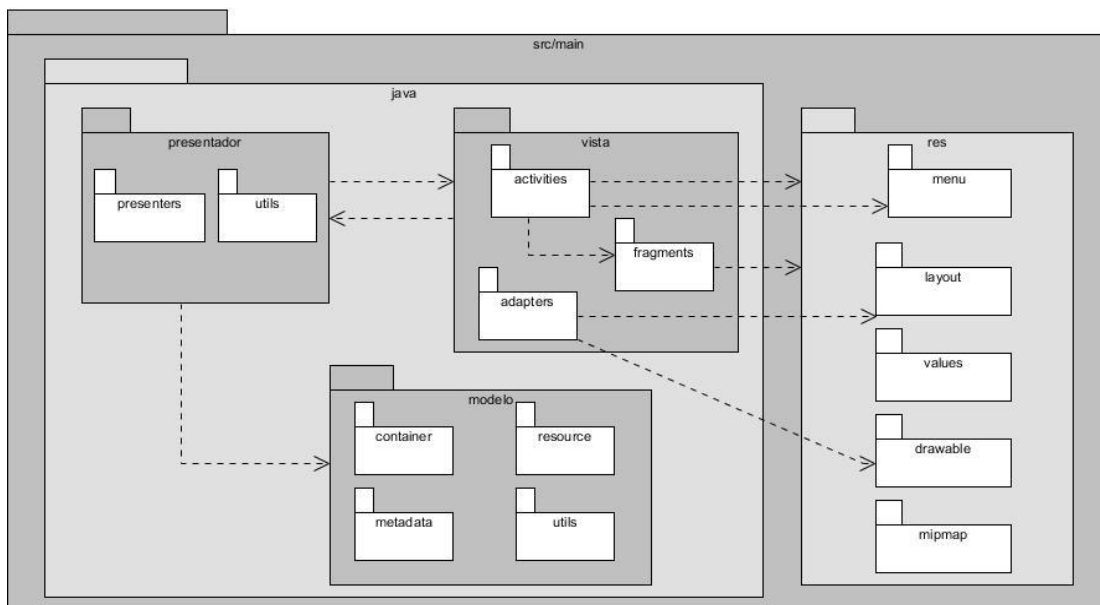


Figura 9: Diagrama de paquetes que ilustra la estructura de la propuesta de solución, basada en MVP. Elaboración propia.

- ✓ **Vista:** Muestra la interfaz de comunicación con el usuario. Es la representación visual de los datos a través del presentador. Dentro de esta capa se encuentran: *ViewerFragment*, que es la vista a través de la cual se muestra el contenido seleccionado en el índice de contenidos; y *PackagesAdapter*, es la clase encargada de interpretar la información del paquete y adaptarla a su representación como componente de la vista correspondiente.
- ✓ **Presentador:** Contienen las clases que interactúan con la vista y con el modelo de datos. Dentro de esta capa se encuentran: *PackagesListActivity*, que es la clase responsable de listar los paquetes SCORM, y *ViewerActivity*, que es la clase donde se interpreta la información dentro del archivo *imsmanifest.xml*.
- ✓ **Modelo:** Es la representación de los datos de la aplicación. Maneja los datos y controla todas sus transformaciones. Dentro de esta capa se encuentran: *ContainerEntity* y *ResourceEntity*. Estas clases gestionan los datos referentes a los contenidos y los recursos del paquete SCORM seleccionado.

Visor de paquetes de contenido educativo para dispositivos móviles

2.4.2. Patrones de diseño

Según Pressman, "...el diseño basado en patrones crea una aplicación nueva, encontrando un conjunto de soluciones comprobadas para un conjunto de problemas delineado con claridad. Cada problema y su solución está descrito por un patrón de diseño catalogado y analizado por otros ingenieros de software que han encontrado el problema e implantado su solución cuando diseñaban otras aplicaciones. Cada patrón de diseño provee un enfoque demostrado para una parte del problema que debe resolverse...". (56)

Patrones GRASP

"Los patrones *GRASP* son patrones de diseño de software para la asignación de responsabilidades. Es el acrónimo de *GRASP* (*object-oriented design General Responsibility Assignment Software Patterns*). Los patrones *GRASP*, más que patrones propiamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software." (61)

- ✓ **Experto:** Posibilita una adecuada asignación de responsabilidades facilitando la comprensión del sistema, su mantenimiento y adaptación a los cambios con re-utilización de componentes. Consiste en la asignación de una responsabilidad a la clase que cuenta con la información necesaria para cumplirla. Ejemplo del uso de este patrón en la propuesta de solución se muestra en las clases *ContainerEntity* y *ResourceEntity*.
- ✓ **Bajo Acoplamiento:** Es una medida de la fuerza con que una clase se relaciona con otras; una clase con bajo acoplamiento no depende de muchas otras. Ejemplo del uso de este patrón en la propuesta de solución se muestra en las clases *PackagesAdapter* y *SearchAdapter*.
- ✓ **Alta Cohesión:** Es una medida que determina cuán relacionadas y adecuadas están las responsabilidades de una clase, de manera que no realice un trabajo colosal; una clase con baja cohesión realiza un trabajo excesivo, haciéndola difícil de comprender, reutilizar y conservar. Ejemplo del uso de este patrón en la propuesta de solución se muestra entre las clases *PackagesListActivity* y *UnzipTask*, donde la primera utiliza la otra solamente para cuando necesita descomprimir el paquete.
- ✓ **Creador:** Aporta un principio general para la creación de objetos, una de las actividades más frecuentes en programación. Este patrón especifica quien debe ser responsable de crear alguna ins-

Visor de paquetes de contenido educativo para dispositivos móviles

tancia de alguna clase. Ejemplo del uso de este patrón en la propuesta de solución se muestra en las clases *ViewerActivity* y *PackagesListActivity*.

- ✓ **Controlador:** Es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Ejemplo del uso de este patrón en la propuesta de solución se muestra en las clases *ViewerActivity* y *PackagesListActivity*.

Patrones GOF

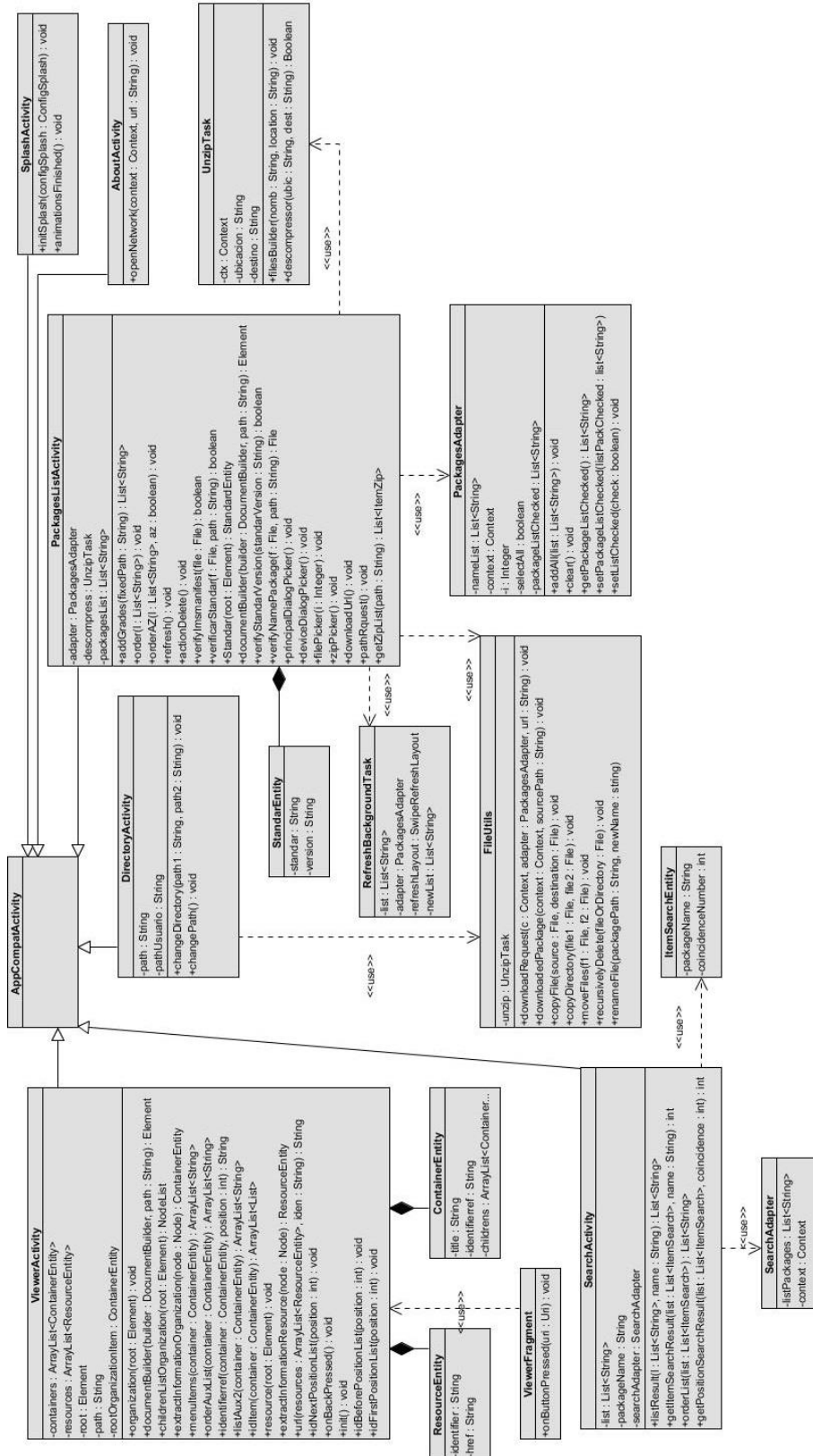
Los patrones *GOF*, o también conocidos como “los patrones de la pandilla de los cuatro” (*Gang of Four*); proporcionan un conjunto de estilos mediante los cuales se pueden construir sistemas orientados a objetos bien diseñados. (56)

Composite: Permite construir objetos complejos mediante composición recursiva de objetos similares. El patrón *Composite* también permite que los objetos del árbol sean manipulados por un manejador consistente, para requerir todos los objetos hay una superclase o un interfaz común. Se empleó en la clase *ContainerEntity*, la cual posee entre sus atributos una lista de objetos de tipo *ContainerEntity*, evidenciándose la recursividad.

2.4.3. Diagrama de clases de diseño

Los diagramas de clases del diseño se emplean en la modelación de las vistas del diseño del sistema para describir las especificaciones de las interfaces y las clases del software, lo que facilita el trabajo de los implementadores. Teniendo en cuenta lo antes expuesto, se propone el siguiente diagrama de clases del diseño:

Visor de paquetes de contenido educativo para dispositivos móviles



Visor de paquetes de contenido educativo para dispositivos móviles

Figura 10: Diagrama de clases del diseño. Elaboración propia.

2.5. Valoraciones finales

A partir del desarrollo del presente capítulo se arribó a las siguientes conclusiones:

1. Se definieron las especificaciones de la aplicación, para un total de 3 requisitos no funcionales y 12 requisitos funcionales, de los cuales 4 requisitos presentan Alta prioridad, 6 con Media y 2 con Baja. Los mismos fueron encapsulados en 11 historias de usuarios, donde cada uno fueron descrito detalladamente, para lograr una mejor comprensión de las funcionalidades de la aplicación que se desea desarrollar.
2. El uso de la arquitectura MVP y de los patrones de diseño GRASP y GOF, contribuyeron al diseño de la aplicación, propiciando una correcta estructura para la misma, la reutilización de código y el empleo de buenas prácticas de programación.
3. Debido a la realización del diagrama de clases del diseño, se obtuvo una visión más exacta de la aplicación en términos de implementación.
4. Con el presente capítulo, queda definida la propuesta de solución para la problemática planteada, descrita a través de los requisitos funcionales y no funcionales, las historias de usuarios, regida por una arquitectura y un diagrama de clases, apoyado a través de patrones de diseño.

Capítulo 3. Implementación y prueba

Para lograr una mejor comprensión del código, es necesario describir los estándares de codificación a utilizar durante la implementación de la aplicación. Además, la definición de la estrategia de prueba, y los casos de pruebas correspondientes a estas. Los resultados de las pruebas permitirán validar el cumplimiento general de los objetivos propuestos y la calidad en la que se lograron.

3.1. Implementación

La codificación de la solución propuesta tiene lugar una vez que se han definido las historias de usuario y se ha concluido el diseño de la aplicación. Cuyos objetivos están encaminados a desarrollar de forma iterativa e incremental un producto completo listo para el despliegue, obteniendo versiones útiles de forma rápida, las que paulatinamente completan el desarrollo de la aplicación.

3.1.1. Estándares de codificación

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. El uso de estándares de codificación trae consigo los siguientes beneficios:

- ✓ Facilitan el mantenimiento de una aplicación.
- ✓ Permiten que cualquier programador entienda y pueda mantener la aplicación.
- ✓ Mejoran la legibilidad del código, al mismo tiempo que permiten su rápida comprensión. (55)

En la propuesta de solución se utilizaron diferentes estilos que se describen de la manera siguiente:

- ✓ Se empleará el idioma inglés para la codificación de la propuesta de solución.
- ✓ Se usará una línea en blanco entre: métodos, variables locales de un método y la primera sentencia, entre diferentes secciones lógicas dentro de un fichero.
- ✓ Los comentarios que abarquen un bloque de instrucciones se escribirán comenzando con los caracteres `/*` y terminaran con `*/`.
- ✓ Los comentarios para una línea comenzaran por `//`.
- ✓ Deben rodearse de espacios en blanco los siguientes operadores:

Asignación (`=`).

Visor de paquetes de contenido educativo para dispositivos móviles

2. Asignación de aumentación (“+=”, “-=”, “*=”, “/=”).
 3. Comparación (“==”, “<”, “>”, “>=”, “<=”, “!=”, “<>”, “in”, “not in”, “is not”).
 4. Expresiones lógicas.
- ✓ Las clases que sean *activities* deberán nombrarse el nombre de la clase y el “*Activity*” después, y la inicial en mayúscula ya sea simple o compuesta su nombre. Ejemplo: “*NameActivity*”.
 - ✓ Los métodos y variables se deberán nombrarse la primera letra de la primera palabra en minúsculas, el resto de las palabras empiezan por mayúsculas. No se utilizará en ningún caso el carácter “_”.
 - ✓ No se usarán nombres de variables que coincidan con palabras reservadas.
 - ✓ No se emplearán caracteres especiales (@, #, \$, %, &, /, * u otros) para la nomenclatura.
 - ✓ Las variables globales deberán ser declaradas como *public static*.
 - ✓ Se debe declarar cada variable en una línea distinta, de esta forma cada variable se puede comentar por separado.

3.2. Pruebas de software

En todas las fases del desarrollo del proyecto hay que probar el *software* que se va construyendo, la etapa de prueba es tan o más importante que todas las realizadas hasta el momento puesto que en ella se refleja la calidad con que ha sido llevada a cabo la construcción del sistema. Las pruebas de *software* consisten en una serie de actividades en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, realizando una evaluación del sistema o del componente probado. Su principal objetivo es evaluar la calidad del producto. Además de ser un elemento crítico para la garantía de la calidad del software. (55) Las pruebas se dividen en los siguientes niveles principales: pruebas de unidad, de integración, de validación, de sistema y de aceptación.

Para la propuesta de solución se utilizó una estrategia de pruebas basada en la ejecución de las mismas tomando como guía tres de estos niveles: unidad, validación y aceptación.

Visor de paquetes de contenido educativo para dispositivos móviles

3.2.2. Estrategia de pruebas

Según Pressman, "...una estrategia de prueba de software debe ser suficientemente flexible para promover un uso personalizado de la prueba. Al mismo tiempo, debe ser suficientemente rígida para alentar la planificación razonable y el seguimiento de la gestión conforme avanza el proyecto. Las pruebas se encuentran clasificadas a través de los siguientes niveles principales: pruebas de unidad, de integración, de validación, de sistema y de aceptación...". (56)

Para el desarrollo de las pruebas a la propuesta de solución se decide utilizar una estrategia de prueba tomando como guía estos niveles pruebas de unidad, de validación y de aceptación.

Pruebas de Unidad

Las pruebas de unidad son el proceso de probar los componentes individuales del sistema, con el objetivo de encontrar los defectos en el mismo. En esta etapa, pueden probarse diferentes tipos de componentes, tales como, funciones individuales o métodos dentro de un objeto; clases de objetos que tienen varios atributos y métodos; y componentes compuestos formados por diferentes objetos y funciones. De ellos, las funciones o métodos individuales son el tipo más simple de componente y sus pruebas son un conjunto de llamadas a estas rutinas con diferentes parámetros de entrada. (56)

El método aplicado para esta prueba fue el de caja blanca, donde emplea pruebas con acceso al código fuente (datos y lógica). Se trabaja con entradas, salidas y el conocimiento interno. Según Pressman, "los casos de prueba se derivan para asegurar que todos los enunciados en el programa se ejecutaron al menos una vez durante las pruebas y que todas las condiciones lógicas se revisaron". (56)

Para automatizar este tipo de pruebas sobre la propuesta de solución se decidió emplear la herramienta *JUnit*, la cual está integrada con el *IDE Android Studio* utilizado para el desarrollo de la aplicación. Se implementaron los casos de prueba a través de la clase *ListarTest*, la cual comprueba el correcto funcionamiento de las principales funcionalidades definidas en la clase *PackagesListActivity*, una de las clases responsables de la lógica de la aplicación; y *UnzipTest* de la clase *UnzipTask*.

Pruebas de validación

Las pruebas de validación verifican que los requerimientos establecidos como parte del modelado fueron cumplidos en el resultado final del *software*. Comienzan después de la realización de las pruebas

Visor de paquetes de contenido educativo para dispositivos móviles

de integración, cuando se descubrieron y corrigieron los errores de interfaz. Las pruebas se orientan en las acciones visibles para el usuario y en las salidas reconocidas por el mismo. (56)

Para realizarlas, el método a emplear fue el de caja negra, se enfoca en los requerimientos funcionales del *software*; derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. (56)

Como técnica se utilizó la de partición de equivalencia, el cual divide el dominio de entrada de un programa en clases de datos de los que pueden derivarse casos de prueba. Un caso de prueba correcto permite descubrir un conjunto de errores de forma directa, simplificando el número de casos de pruebas para encontrar un error general. (56)

Diseño de casos de pruebas

Los casos de pruebas fueron definidos a través de los requisitos funcionales descritas en las historias de usuario. La intención que se persigue con estos artefactos es lograr una comprensión específica de las condiciones que la solución debe cumplir.

Cada planilla de casos de pruebas contiene la especificación de una historia de usuario, a través de una descripción, condición de ejecución, escenario, respuesta del sistema y flujo central; detallando así, las funcionalidades de la aplicación.

A continuación, se muestran los casos de prueba correspondientes a las historias de usuario “Listar paquetes SCORM importados” y “Buscar paquete SCORM de la lista de paquetes importados”, debido a que son funcionalidades fundamentales en el proceso de desarrollo. Los casos de prueba correspondientes al resto de las historias de usuario se encuentran en la sección de anexos, al final del documento.

Abreviaturas utilizadas:

CP: Caso de Prueba.

HU: Historia de Usuario.

EC: Escenario.

Tabla 5: Caso de Prueba de Validación para la Historia de Usuario Listar paquetes SCORM importados. Elaboración propia.

Visor de paquetes de contenido educativo para dispositivos móviles

Código: CP3_HU3		HU_3: Listar paquetes SCORM importados	
Responsable: Beatriz Ontivero González			
Descripción: El caso de prueba se inicia al ejecutarse la aplicación. Se presenta al usuario un listado de los paquetes SCORM que se encuentran importados en la aplicación. El caso de prueba termina cuando el usuario sale de la actividad.			
Condiciones de ejecución: El usuario debe haber seleccionado el directorio donde se serán almacenados los paquetes SCORM.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 3.1 listar paquetes SCORM sin haber seleccionado previamente un directorio de almacenamiento para los paquetes.	El usuario ejecuta la aplicación y esta muestra la lista de paquetes vacía.	La aplicación mostrará la lista de paquetes vacía y mostrará un mensaje que indique al usuario que debe seleccionar un directorio.	Se inicia la aplicación.
EC 3.2 Mostrar el listado de paquetes SCORM en la aplicación con directorio seleccionado, sin haber importado	El usuario ejecuta la aplicación y esta muestra la lista de paquetes vacía.	La aplicación mostrará la lista de paquetes vacía.	Se inicia la aplicación.

Visor de paquetes de contenido educativo para dispositivos móviles

paquetes y el directorio este vacío.			
EC 3.3 Mostrar el listado de paquetes SCORM en la aplicación con directorio seleccionado, sin haber importado paquetes y el directorio contenga ficheros que no sean paquetes SCORM.	El usuario ejecuta la aplicación y esta muestra la lista de paquetes vacía.	La aplicación mostrará la lista de paquetes vacía.	Se inicia la aplicación.
EC 3.4 Mostrar el listado de paquetes SCORM en la aplicación con directorio seleccionado, sin haber importado paquetes y el directorio contenga ficheros que sean y no sean, paquetes SCORM.	El usuario ejecuta la aplicación y esta muestra la lista con los paquetes con el estándar SCORM.	La aplicación mostrará la lista de paquetes SCORM.	Se inicia la aplicación.

Visor de paquetes de contenido educativo para dispositivos móviles

EC 3.5 Mostrar el listado de paquetes SCORM en la aplicación con directorio seleccionado, y que el paquete importado no tenga el estándar SCORM.	El usuario ejecuta la aplicación y esta muestra la lista de paquetes vacía.	La aplicación mostrará la lista de paquetes vacía.	Se inicia la aplicación.
EC 3.6 Mostrar el listado de paquetes SCORM en la aplicación con directorio seleccionado, y que el paquete importado tenga el estándar SCORM.	El usuario ejecuta la aplicación y esta muestra la lista con los paquetes con el estándar SCORM.	La aplicación mostrará la lista de paquetes SCORM.	Se inicia la aplicación.

Pruebas de Aceptación

Las pruebas de aceptación se realizan en la etapa final del proceso de pruebas. Las pruebas de aceptación son las pruebas donde se le entrega una versión del sistema al cliente, implicándolo en las mismas. Donde, si la entrega es suficiente buena, el cliente puede entonces aceptarla para su uso. (55)

Las pruebas de aceptación son realizadas por el usuario final en lugar de los desarrolladores. Estas pueden realizarse durante un período de semanas o meses, y mediante ellas, descubrir errores acumulados que con el tiempo puedan degradar el sistema. Con el fin de descubrir errores que solo el usuario final es capaz de encontrar, las pruebas de aceptación se llevarán a cabo a través de un proceso llamado prueba alfa. Se realiza en el sitio del desarrollador por un grupo representativo de usuarios finales, permitiendo que el desarrollador registre errores y problemas de uso.

3.1.2. Ejecución y resultados de las pruebas de software

Para la ejecución de las pruebas de validación y de aceptación se utilizaron las siguientes terminales:

Visor de paquetes de contenido educativo para dispositivos móviles

✓ Teléfono celular:

- Modelo: *Alcatel One Touch 5042T*
- Versión de SO *Android: 4.4.4 (Kitkat)*
- SDK: 26
- CPU: *Quad-core 1.2 GHz Cortex-A53*
- RAM: 1 Gb
- Almacenamiento interno: 1.80 Gb
- Almacenamiento externo: 14.83 Gb

✓ Teléfono celular:

- Modelo: *Galaxi Nexus*
- API: 25
- Versión de SO *Android: 7.1.1 (Nougat)*
- SDK: 26
- CPU: *Intel Atom*
- RAM: 1 Gb
- Almacenamiento interno: 2 Gb

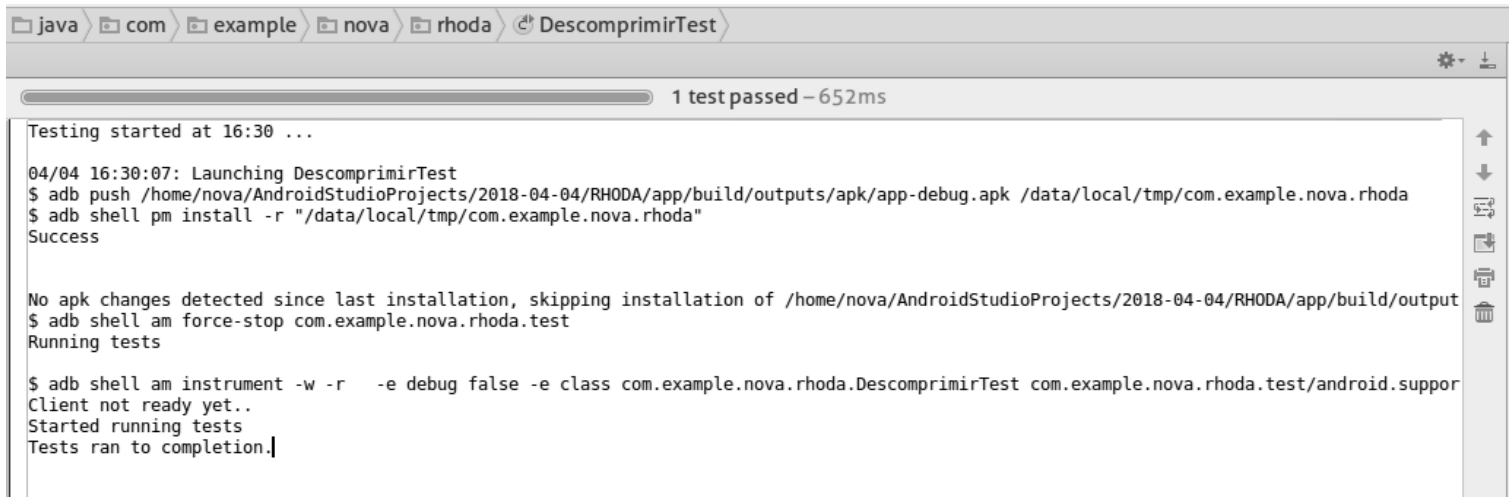
✓ Emulador *Genymotion*:

- Modelo: *Genymotion _vbox86p_4.4_150216_21300*
- Versión de SO *Android: 4.3 (Jelly Bean)*
- SDK: 26
- *Intel® Core™ i3-2120 @3.30 GHz*
- RAM: 512Mb

Visor de paquetes de contenido educativo para dispositivos móviles

- Almacenamiento interno: 2024 Mb
- Almacenamiento externo: 8189 Mb

A continuación, se muestran imágenes que ilustran el desarrollo de las pruebas unitarias de los resultados obtenidos para los casos de pruebas implementados.

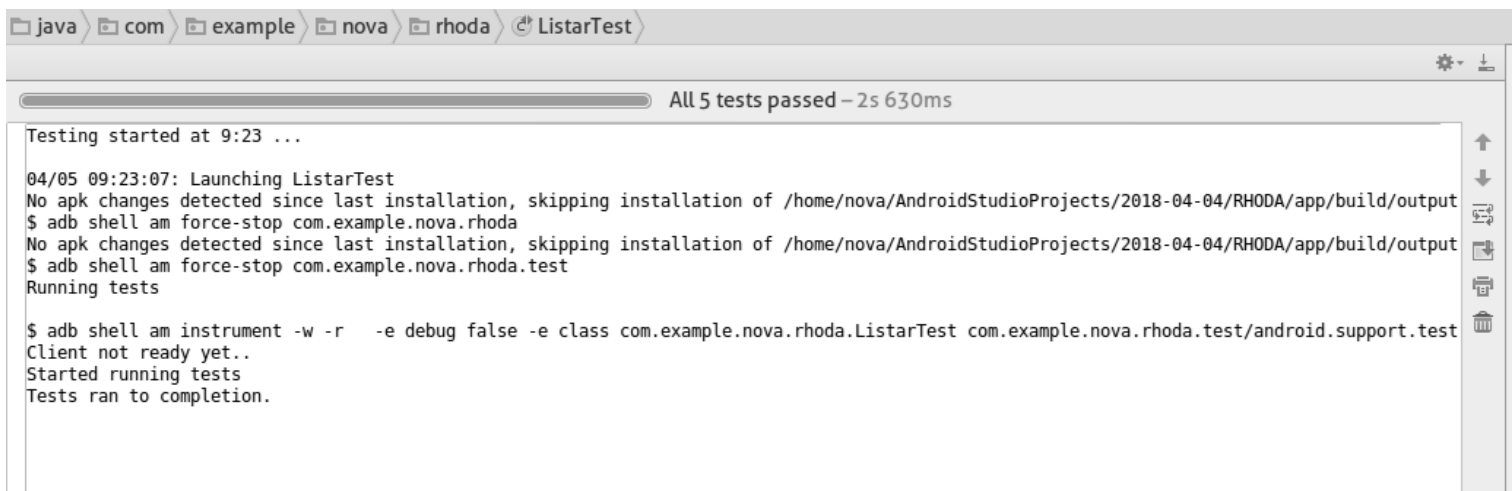


```
java > com > example > nova > rhoda > DescomprimirTest
1 test passed - 652ms
Testing started at 16:30 ...
04/04 16:30:07: Launching DescomprimirTest
$ adb push /home/nova/AndroidStudioProjects/2018-04-04/RHODA/app/build/outputs/apk/app-debug.apk /data/local/tmp/com.example.nova.rhoda
$ adb shell pm install -r "/data/local/tmp/com.example.nova.rhoda"
Success

No apk changes detected since last installation, skipping installation of /home/nova/AndroidStudioProjects/2018-04-04/RHODA/app/build/output
$ adb shell am force-stop com.example.nova.rhoda.test
Running tests

$ adb shell am instrument -w -r -e debug false -e class com.example.nova.rhoda.DescomprimirTest com.example.nova.rhoda.test/android.support
Client not ready yet..
Started running tests
Tests ran to completion.]
```

Figura 11: Resultados de JUnit en la clase UnzipTest.



```
java > com > example > nova > rhoda > ListarTest
All 5 tests passed - 2s 630ms
Testing started at 9:23 ...
04/05 09:23:07: Launching ListarTest
No apk changes detected since last installation, skipping installation of /home/nova/AndroidStudioProjects/2018-04-04/RHODA/app/build/output
$ adb shell am force-stop com.example.nova.rhoda
No apk changes detected since last installation, skipping installation of /home/nova/AndroidStudioProjects/2018-04-04/RHODA/app/build/output
$ adb shell am force-stop com.example.nova.rhoda.test
Running tests

$ adb shell am instrument -w -r -e debug false -e class com.example.nova.rhoda.ListarTest com.example.nova.rhoda.test/android.support.test
Client not ready yet..
Started running tests
Tests ran to completion.
```

Figura 12: Resultados de JUnit en la clase ListTest.

A partir de la interpretación de dichos resultados se puede afirmar que los mismos garantizan el correcto funcionamiento de los métodos de la clase *UnzipTask* y *ListTest*.

Visor de paquetes de contenido educativo para dispositivos móviles

Los problemas detectados en el período de pruebas de validación y aceptación se clasificaron en: No conformidades significativas (NCS) y en No conformidades no significativas (NCNS).

Las **NCS** son las no conformidades referentes a las funcionalidades de la aplicación; como son validaciones incorrectas o respuestas de la aplicación diferentes a lo descrito a lo descrito previamente en las historias de usuario.

Las **NCNS** son las no conformidades en cuanto al diseño de la propuesta de solución y errores ortográficos.

Fueron realizadas 3 iteraciones de pruebas, realizando pruebas de regresión a cada una de ellas con el fin de asegurar que después de resolver las no conformidades encontradas, estas no introdujeran nuevos errores en la misma iteración.

En la primera iteración se detectaron 8 NCS, donde se encontraron 4 errores de funcionalidad, 3 validaciones incorrectas y 1 respuesta diferente; y 5 NCNS, donde se encontraron 3 no conformidades en el diseño y 2 de errores ortográfico. Las mismas fueron resueltas satisfactoriamente en la misma iteración.

En la segunda iteración se identificaron 5 NCS, donde se descubrieron 2 de funcionalidad, 2 validaciones incorrectas y 1 respuesta incorrecta; y 3 NCNS, 2 no conformidades en el diseño y 1 de errores ortográficos.

En la tercera iteración no se detectaron NCS ni NCNS, por lo que se demostró que la aplicación cumple con los requisitos funcionales establecidos.

A continuación, se muestra una gráfica de barras, describiendo las no conformidades encontradas durante las 3 iteraciones como resultado de las pruebas de *software* realizadas; divididas en significativas y no significativas.

Visor de paquetes de contenido educativo para dispositivos móviles

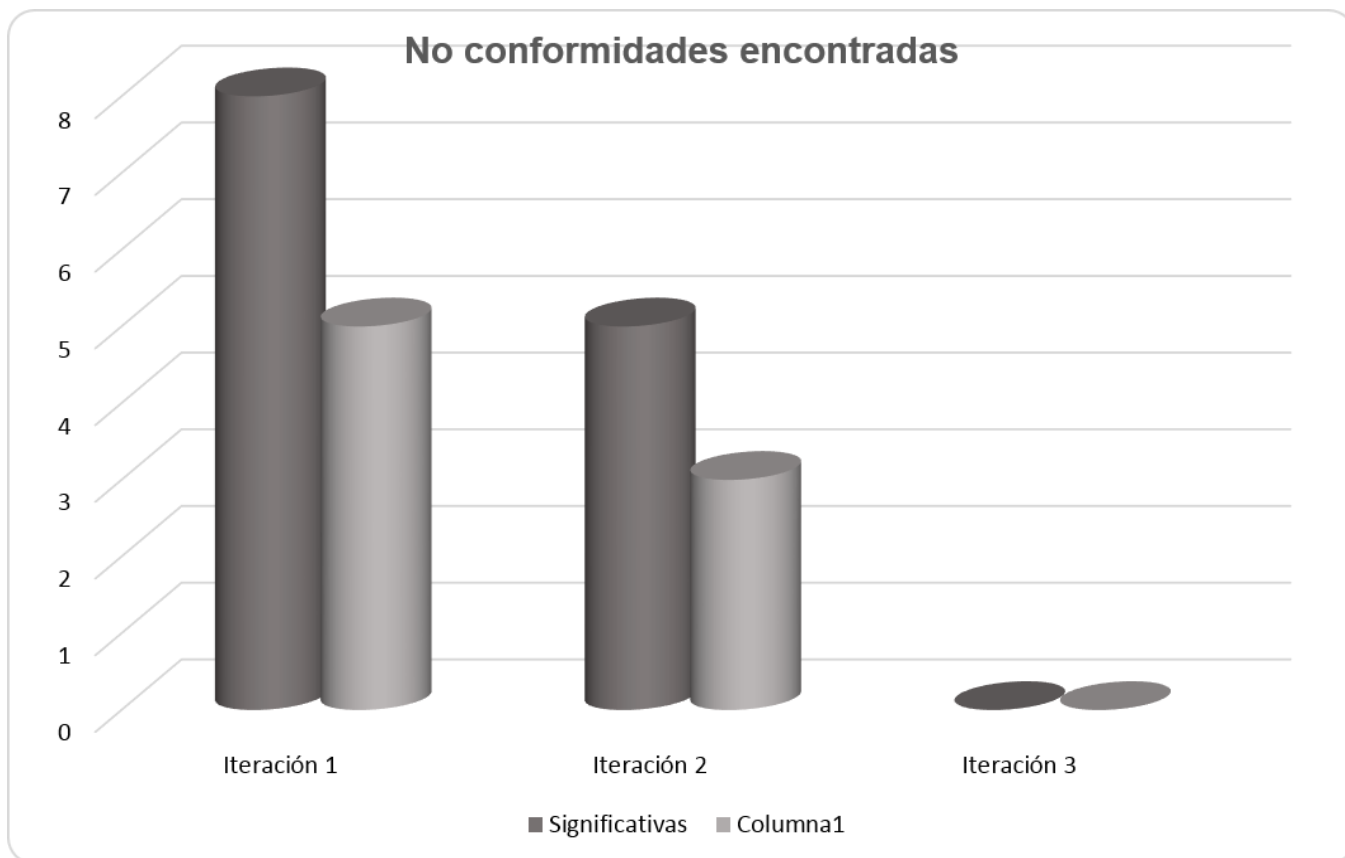


Figura 13: No conformidades encontradas por iteración en el proceso de pruebas. Elaboración propia.

3.3. Validación de la investigación

Para la validación de la investigación y partiendo de la hipótesis planteada se diseñó un cuasi experimento que permitiera comprobar la utilidad de la aplicación para estudiantes de Educación a Distancia que estuvieran en diferentes escenarios de conectividad.

Para llevar a cabo el experimento se escogió los 25 estudiantes de primer año del CUM¹¹ de San Nicolás de Bari en la carrera de Ingeniería en Procesos Agroindustriales, primera carrera de ciencias técnicas que se estudia en la modalidad a distancia utilizando el nuevo modelo. En este caso la población y la muestra coinciden ya que se les proporcionó la aplicación a todos los estudiantes los cuales contaban con un dispositivo móvil con Android entre teléfonos y Tablet.

¹¹ CUM: Centro de la Universitario Municipal.

Visor de paquetes de contenido educativo para dispositivos móviles

Luego de dos meses de utilizar la aplicación de manera intencionada con la colaboración de los profesores del CUM se le aplicó una encuesta (Ver anexo 2) que arrojó los siguientes resultados:

No	Criterio	Valor
1	Escenario de conectividad según Modelos de EaD: <ul style="list-style-type: none"> ✓ Sin conectividad o baja conectividad. ✓ Total conectividad. 	21 – 84% 4 – 16%
2	Facilidad de uso de la aplicación: <ul style="list-style-type: none"> ✓ Muy fácil. ✓ Fácil. ✓ Difícil. 	8 – 32% 16 – 64% 1 – 4%
3	Frecuencia de uso de la aplicación: <ul style="list-style-type: none"> ✓ Mucha. ✓ Poca. ✓ Ninguna. 	20 – 80% 4 – 16% 1 – 4%
4	Lugares donde más se utilizó la aplicación para consultar contenido: <ul style="list-style-type: none"> ✓ CUM. ✓ Centro de Trabajo. ✓ Casa. ✓ Otros. 	3 – 12% 7 – 28% 12 – 48% 3 – 12%
5	Utilidad para el estudio individual:	

Visor de paquetes de contenido educativo para dispositivos móviles

✓ Muy útil.	14 – 56%
✓ Útil.	11 – 44%
✓ Sin utilidad.	0 – 0%

De los datos en la tabla anterior se puede llegar a la conclusión de que los estudiantes que más utilizaron la aplicación son los que se encuentran mayormente en escenarios de baja o ninguna conectividad. Además, lograron tener los recursos disponibles para el estudio individual en su casa como lugar donde mayormente realizaron el mismo.

3.4. Valoraciones finales.

A partir del desarrollo de presente capítulo se arribaron las siguientes conclusiones:

1. La descripción del proceso de implementación de la aplicación, a través de la definición de los estándares utilizados para la codificación, posibilitando más legibilidad del código y haciéndolo más comprensible.
2. Las pruebas de caja blanca, caja negra y alfa; permitieron, a través del diseño de casos de usos, evaluar el cumplimiento de los requerimientos del cliente y validar el correcto funcionamiento del código de la aplicación.

Visor de paquetes de contenido educativo para dispositivos móviles

Conclusiones generales.

Con el cumplimiento de los objetivos trazados y a raíz de los resultados obtenidos, se puede arribar a las siguientes conclusiones:

1. El análisis de los elementos teóricos relacionados con los visores de paquetes de contenido educativo para dispositivos móviles posibilitó una mejor comprensión de los conceptos asociados.
2. El estudio de las principales aplicaciones para dispositivos móviles que visualizan e interpretan paquetes de contenido educativo rigiéndose por el estándar *SCORM*, permitió comprobar que la existencia de aplicaciones con estas características, es escasa, y las existentes no brindan una solución completa al problema a resolver.
3. La definición de los requisitos funcionales y no funcionales, descritos a través de historias de usuario, y la posterior codificación, facilitó la comprensión de cada uno de los requerimientos que precisaba el cliente.
4. La implementación de la aplicación Visor de paquetes de contenido educativo para dispositivos móviles posibilitó darles respuesta a los requisitos definidos y con ello a las necesidades del cliente.
5. La ejecución de pruebas al Visor de paquetes de contenido educativo para dispositivos móviles permitió la validación de todos los requisitos identificados y la aceptación por parte del cliente. (Ver anexo 3).

Visor de paquetes de contenido educativo para dispositivos móviles

Recomendaciones.

1. Permitir la interpretación y visualización de otros estándares para empaquetado de objetos de aprendizaje.
2. Incorporar otros idiomas a la aplicación para permitirle al usuario seleccionar el idioma que desee.
3. Establecer un sistema de clasificación basados en los metadatos de los paquetes.
4. En la funcionalidad de importar paquete desde el dispositivo en formato *Zip*, la aplicación debe permitir visualizar una lista con todos los ficheros que se encuentren en este formato ubicados en el dispositivo móvil.

Visor de paquetes de contenido educativo para dispositivos móviles

Referencias Bibliográficas

1. Falcón Villaverde, Marianela. *La educación a distancia y su relación con las nuevas tecnologías de la información y las comunicaciones*. Centro Provincial de Información, Universidad de Ciencias Médicas. Cienfuegos. Cuba : Medisur, 2013. págs. 280 - 295. ISSN 1727-897X.
2. Ramírez Montoya, María Soledad. *Recursos Tecnológicos para el Aprendizaje Móvil (m-Learning) y su Relación con los Ambientes de Educación a Distancia: implementaciones e investigaciones*. s.l. : RIED. Revista Iberoamericana de Educación a Distancia., 2009. ISSN: 1138-2783.
3. Menéndez, Víctor H., Prieto, Manuel E. and Zapata, Alfredo. *Sistemas de Gestión Integral de Objetos de Aprendizaje*. s.l. : IEEE-RITA, 2010. págs. 56 - 62.
4. Alfonso Sanchez, Ileana R. *La educación a distancia*. Ciudad de La Habana. Cuba : ACIMED, 2003. ISSN 1024-9435.
5. Vidal Ledo, María J., y otros. *Aprendizaje Móvil*. Escuela Nacional de Salud Pública. Habana. Cuba : s.n., 2015. págs. 669 - 679. ISSN 1561-2902..
6. Ávila, Dr. Carlos González. *El libre uso del lenguaje en los formatos Publicitarios*. V Congreso Internacional Latina de Comunicación, Universidad de La Laguna. España : s.n., 2013.
7. Alonso, Arturo Baz, Artime, Irene Ferreira, Rodríguez, María Álvarez and Baniello, Rosana García. *Dispositivos móviles*. EPSIG Ing. Telecomunicación Universidad de Oviedo. España : s.n., 2011.
8. *17 Tecnologías de la Información*. ONEI. Cuba : s.n., 2017.
9. Tips, Tecno. *Impacto de la tecnología móvil en la educación*. Veracruz, México : UNIVERSO: el periódico de los universitarios, 2012.
10. Fombona Cadavieco, Javier, Pascual Sevillano, María Ángeles y Ferreira Amador, María Filomena. *Realidad aumentada, una evolución de las aplicaciones de los dispositivos móviles*. Universidad de Sevilla. Sevilla. España : Pixel-Bit, 2012. págs. 197 - 210, Revista de Medios y Educación. ISSN: 1133-8482.
11. Cantillo Valero, Carmen y Roura Redondo, Margarita y Sánchez Palacín, Ana. *Tendencias actuales en el uso de dispositivos móviles en educación*. Departamento de Desarrollo Humano, Educación y Cultura,

Visor de paquetes de contenido educativo para dispositivos móviles

Comunicación y Educación en la Red: de la Sociedad de la Información a la Sociedad del Conocimiento. España : Portal Educativo de las Américas, 2012. ISSN 0013-1059.

12. *Centro Nacional de Educación a Distancia*. Universidad de las Ciencias Informáticas. Cuba : s.n.

13. *RHODA. Bienvenido al repositorio de objetos de aprendizaje*. Cuba : s.n., 2017.

14. Nuñez Padron, Sandy y Pérez González, Dianelys. *SCORMVIEW: Visor De Paquetes Scorm Para Xauce Movil*. Universidad de las Ciencias Informáticas. Cuba : s.n., 2015. Tesis de grado.

15. Waldo Gerard, Ralph. *Applied Science and Technological Progress: A Report to the Committee on Science and Astronautics*. House of Representatives, Government Printing. Washington, D.C.: U.S : s.n., 1967.

16. Polsani, Pithamber R. *Use and Abuse of Reusable Learning Objects*. s.l. : Journal of Digital Information, 2006.

17. *IEEE Learning Technology Standards Committee (LTSC)*. s.l. : IEEE STANDARDS ASSOCIATION, 2013.

18. Wiley II, David A. *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy*. 2003. pág. 35.

19. Villegas Dianta, Adrián. *Los Objetos Digitales de Aprendizaje (ODAs)*. s.l. : E-Historia, 2010.

20. García Peñalvo, Francisco José. *Estado actual de los sistemas e-learning. Teoría de la Educación. Educación y Cultura en la Sociedad de la Información*, Universidad de Salamanca. España : Sistema de Información Científica, 2005. ISSN: 1138-9737.

21. Ortuño Campos, Rosalynn Argelia, Morgado Morales, Erla Mariela y Rodríguez Orozxo, Claudia Margarita. *Diseño de Objetos de Aprendizaje adaptados para cuatro estilos de aprender: un estudio de caso*. s.l. : UFMT. Revista de Educación Pública, 2016. págs. 548 - 572. ISSN 2238-2097.

22. García Aretio, Lorenzo. *Objetos de aprendizaje. Características y repositorios*. Titular de la Cátedra UNESCO de Educación a Distancia (CUED). s.l. : UNED, 2013. ISSN: 2340-552X.

Visor de paquetes de contenido educativo para dispositivos móviles

23. Janson, Andreas y Söllner, Matthias and Leimeister, Jan Marco. *Individual Appropriation of Learning Management Systems - Antecedents and Consequences*. *Transacciones AIS en interacción humano-computadora*. 2017. págs. 173 - 201.
24. Chaubey, Aabha and Bhattacharya, Bani. *Learning Management System in Higher Education*. s.l. : Revista Internacional de Ciencia, Tecnología e Ingeniería, 2015. págs. 158 - 162.
25. Watson, Sunnie Lee and Watson, William R. *An Argument for Clarity: What are Learning Management Systems, what are They Not, and What Should They Become?* Universidad. Indiana : s.n., 2012. pág. 18.
26. Goikolea, Markos. *¿Qué son los sistemas de gestión del aprendizaje?* s.l. : Noticias Iberestudios. Internacional., 2013.
27. *Características, tipos y plataformas más utilizadas para estudiar a distancia*. Universidad. Valencia. España : s.n., 2015.
28. Fainholc, Beatriz. *Optimizando las Posibilidades de las TICs en Educación*. s.l. : Edutec. Revista, 2006.
29. Area, Manuel y Adell, Jordi. *E-Learning: enseñar y aprender en espacios virtuales*. 2009. págs. 391 - 424.
30. De la Peña Esteban, Francisco David y Burgos García, María Concepción. *Modelo Práctico De Aplicación (App) Para Dispositivo Móvil*. Universitaria De Enseñanza A Distancia. Madrid. España : EDUTEC. Revista Electrónica de Tecnología Educativa, 2015. ISSN: 1135-9250.
31. Foix, Christian y Zavando, Sonia. *Estándares e-Learning: Estado del arte*. Centro de Tecnologías de Información. Chile : s.n., 2002.
32. Boneu, Josep M. *Plataformas abiertas de e-Learning para el soporte de contenidos educativos abiertos*. Universidad Oberta de Catalunya. Barcelona. España : RUSC. Revista de la Universidad y Sociedad del Conocimiento, 2007. págs. 34 - 47.
33. Morales Morgado, Erla Mariela. *Gestión del conocimiento en sistemas E-Learning, Basado en objetos de aprendizaje, cualitativa y pedagógicamente definidos*. 2007.

Visor de paquetes de contenido educativo para dispositivos móviles

34. Ocsa, Alexander, y otros. *Propuesta para el diseño y desarrollo de aplicaciones M-Learning: caso, apps de historia del Perú como objetos de aprendizaje móviles*. s.l. : Nuevas Ideas en Informática Educativa TISE, 2014.
35. Emilio Biscari, Carlos. *Los estándares de e-Learning*. Universidad de Palermo. págs. 49 - 74.
36. Diego Hevia, Darien, Galvez Alonso, Arlan y Molina Toledo, Pablo. *Visor de contenidos para la propuesta de reproductor de archivos SCORM*. Universidad de las Ciencias Informáticas. Cuba : s.n., 2014. Tesis de grado.
37. Acosta calvo, Yenny Paola y Ráquira Empaque, Maria E. *Diseño de ambiente virtual para apoyar los procesos de auto-evaluación y acreditación en la Facultad de Ingeniería de la Universidad Distrital Francisco José de Caldas*. Universidad Distrital Francisco José de Caldas. Bogotá D.C. : s.n., 2017.
38. Del Castillo Zabala, Cristian Camilo y Rodríguez Zamora, José Leonardo. *Diseño de un recurso educativo abierto (rea) para la asignatura de bases de datos en el área de ingeniería de sistemas de la Universidad Libre de Colombia*. Universidad Libre de Colombia. Bogotá : s.n., 2017.
39. Díaz Vega, Liena y Jorge Santana, Edwin. *Editor de objetos de aprendizaje basado en SCORM para el marco de trabajo Xalix*. 2016.
40. *Why Android. Android*. s.l. : Google, 2017.
41. *SCORM Player*. s.l. : Google Play, 2018.
42. *G-OpenLMS*. s.l. : GooglePlay.
43. KINASH, Shelley, BRAND, Jeffrey y MATHEW, Trishita. *Challenging mobile learning discourse through research: Student perceptions of Blackboard Mobile Learn and iPads*. s.l. : Australasian Journal of Educational Technology, 2012. págs. vol. 28, no. 4. 1449-5554.
44. *TeachByApp SCORM Player*. s.l. : GooglePlay, 2018.
45. García Domene, Ana. *Lector SCORM para Android*. 2014.
46. Rodríguez Sánchez, Tamara. *Metodología de desarrollo para la actividad productiva de la UCI*. Universidad de las Ciencias Informáticas. La Habana. Cuba : s.n.

Visor de paquetes de contenido educativo para dispositivos móviles

47. Larman, Craig. *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. 2da Edición.*
48. *What is Visual Paradigm?* s.l. : Visual Paradigm, 2017.
49. *Información de Java 8.* s.l. : Oracle Corporation, 2017.
50. *Getting Started with Gradle for Android Build I Gradle.* s.l. : Gradle INC.
51. *Genymotion Plugin for Android Studio Guide - Plugin-for-Android-Studio-1.0.7-Guide.* s.l. : Genymotion, 2017.
52. *Conoce Android Studio.* s.l. : Android Developers, 2017.
53. *Balsamiq Mockups | Balsamiq.* s.l. : Balsamiq Mockups, 2016.
54. *JUnit - About.* s.l. : JUnit, 2016.
55. Sommerville, Ian. *Ingeniería del software. Séptima edición.* Madrid : s.n., 2005. Libro. ISBN: 84-7829-074-5.
56. S. Pressman, Roger. *Ingeniería de Software. Un enfoque práctico. Séptima edición.* México : s.n., 2010. Libro.
57. *Versions.* s.l. : Android Developers, 2018.
58. *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.* s.l. : IEEE, 2000. ISBN 0-7381-2519-9.
59. Díaz González, Yanette y Fernández Romero, Yenisleidy. *Patrón Modelo-Vista-Controlador.* Departamento de Telemática. Facultad de Telecomunicaciones y Electrónica, Universidad Tecnológica de La Habana "José Antonio Echeverría", CUJAE. Cuba : s.n., 2012. págs. 47 - 57. ISSN 1729-3804.
60. Carrera Guanoluisa, Jenny Germania. *Análisis comparativo de la productividad entre los patrones de diseño Modelo Vista Controlador (MVC) y Modelo Vista Presentador (MVP) aplicado al desarrollo del Sistema Nómina de Empleados y Rol de Pagos de la "Distribuidora Soria C.A."*. Facultad de Informática y Electrónica, Escuela de Ingeniería en Sistema. Riobamba. Ecuador : s.n., 2014.

Visor de paquetes de contenido educativo para dispositivos móviles

61. Botero Tabares, Ricardo. *Patrones Grasp y Anti-Patrones: un Enfoque Orientado a Objetos desde Lógica de Programación*. Entre Ciencia e Ingeniería. Facultad de Ciencias Básicas e Ingeniería., Universidad Católica de Pereira. Pereira. Colombia : s.n., 2010. págs. 161 - 173.

Visor de paquetes de contenido educativo para dispositivos móviles

Anexos.

Anexo 1. Entrevista al cliente

- 3 ¿Qué es la Educación a Distancia?
- 4 ¿Cómo se manifiesta esa modalidad en la Universidad y en el país?
- 5 ¿Cuáles son los escenarios tecnológicos que posee la EaD?
- 6 ¿Cómo considera que un dispositivo móvil pueda apoyar a las horas de estudio de los estudiantes en los escenarios tecnológicos mencionados anteriormente?
- 7 ¿Qué estándares se utilizan para el empaquetado de los contenidos educativos en la Universidad y en el país?
- 8 ¿Qué estándar, de los anteriormente mencionados, es el más utilizado en la Universidad y en el país?
- 9 ¿Existen visores de paquetes de contenido educativo en la Universidad o en el país?

Visor de paquetes de contenido educativo para dispositivos móviles

Anexo 2. Encuesta a los estudiantes del CUM

Encuesta

Estimado estudiante de la Educación a Distancia, le pedimos que conteste con la mayor sinceridad las siguientes preguntas. El objetivo de esta encuesta es para determinar si la aplicación móvil Visor de paquetes de contenido educativo para dispositivos móviles les resulta útil, y cómo podríamos mejorarla para ofrecerles un mejor servicio.

- 10 ¿En cuál de los escenarios considera usted que se encuentra? (Total, Baja, Sin conectividad).
- 11 Valore en una escala de 1 a 3, la facilidad de uso de la aplicación. Teniendo en cuenta que:
1 – Difícil. 2 – Fácil. 3 – Muy fácil.
- 12 ¿Con qué frecuencia utilizó la aplicación? (Mucha, Poca, Ninguna).
- 13 ¿Dónde consultó con mayor frecuencia los recursos utilizando la aplicación? (CUM, Centro de trabajo, Casa, Otros).
- 14 Valore en una escala de 1 a 3 la utilidad de la aplicación para su estudio individual. Teniendo en cuenta que:
1 – Sin utilidad. 2 – Útil. 3 – Muy útil.
- 15 Si tiene alguna sugerencia o recomendación para los desarrolladores de la aplicación, expóngala a continuación.

Visor de paquetes de contenido educativo para dispositivos móviles

Anexo 3. Acta de aceptación.

 **CENTRO DE SOLUCIONES LIBRES
CESOL**

ACTA DE ACEPTACIÓN

De una parte, el Centro Nacional de Educación a Distancia en lo sucesivo CENED, representado en este acto por su directora Dra. Lidia Ruiz Ortiz y de **otra parte** el Centro de Soluciones Libres de la Universidad de las Ciencias Informáticas, en lo adelante CESOL, representado por su director MSc. Yoandy Pérez Villazón.

Primero: A raíz de una necesidad del CENED y respondiendo a los escenarios tecnológicos del Modelo de Educación a Distancia para la Educación Superior Cubana, se acuerda el desarrollo de un visor de paquetes SCORM para dispositivos móviles con SO Android para ser utilizado en escenarios con conectividad parcial o limitada o sin conectividad.

Considerando: Que los hitos realizados han sido desarrollados con la calidad requerida y que los requisitos funcionales pactados han sido implementados en su totalidad.

Considerando: Que las pruebas realizadas en entornos de trabajo real han arrojado valores y criterios positivos.

Considerando: Que algunas de las recomendaciones realizadas por estudiantes y profesores de la educación a distancia a esta versión constituirán nuevos requisitos a implementar en una versión 2 del producto.

Por tanto: Las partes acuerdan formalizar mediante la presente acta, la aceptación del producto **Visor de paquetes de contenido educativo SCORM para dispositivos móviles** en su versión 1.0.

Y para que así conste se extiende la presente acta en dos ejemplares rubricados por ambas partes.


MSc. Yoandy Pérez Villazón
Director CESOL
Entrega
Centro CESOL Facultad 1


Dra. Lidia Ruiz Ortiz
Directora CENED
Recibe



Visor de paquetes de contenido educativo para dispositivos móviles

Anexo 4. Historias de Usuario

Tabla 6: Historia de Usuario Importar paquete SCORM desde el dispositivo. Elaboración propia.

Número: 2	Nombre del requisito: Importar paquete SCORM desde el dispositivo.	
Programador: Beatriz Ontivero González	Iteración Asignada: 4	
Prioridad: Alta	Tiempo Estimado: 5 días	
Riesgo en Desarrollo: <ul style="list-style-type: none">✓ Interrupción en el fluido eléctrico.✓ Rotura de la estación de trabajo.✓ Configuración de la estación de trabajo.	Tiempo Real: 5 días	
Descripción: <p>La aplicación debe mostrar al usuario una vista donde podrá seleccionar el formato en que se encuentra el paquete que se desee importar. El paquete SCORM puede estar comprimido, como <i>Zip</i>; o descomprimido, como una carpeta.</p> <p>Si el usuario selecciona afuera de la vista, esta se cancelará y se mostrará nuevamente el listado de paquetes.</p> <p>Si el usuario selecciona una de las opciones explicadas anteriormente, la aplicación mostrará una vista donde muestra un explorador intuitivo y fácil de navegar para seleccionar el paquete SCORM dentro del dispositivo móvil.</p> <p>La vista estará conformada de la siguiente manera:</p> <ul style="list-style-type: none">• Ícono: Muestra un ícono que represente la opción correspondiente.• Opción: Muestra la el formato en que se puede encontrar el paquete a importar.		

Visor de paquetes de contenido educativo para dispositivos móviles

Observaciones:

El usuario debe haber seleccionado la opción de importar el paquete desde el dispositivo en la vista anterior.

Si el usuario no ha seleccionado el directorio de almacenamiento para los paquetes SCORM, la aplicación mostrará un mensaje indicando que debe seleccionar el directorio de almacenamiento primero.

Si el paquete a importar está comprimido y en el explorador correspondiente, el usuario selecciona otro archivo que no sea *.zip*, la aplicación mostrará un mensaje y le pedirá al usuario que seleccione nuevamente un paquete. Si no selecciona ningún paquete y el usuario toca el botón “aceptar”, la aplicación mostrará un mensaje especificando que debe seleccionar un paquete. Si el usuario selecciona varios paquetes y el usuario toca el botón “aceptar”, la aplicación mostrará un mensaje especificando que debe seleccionar solamente un paquete.

Si el paquete a importar esta descomprimido, es decir, en una carpeta; y el usuario no selecciona ningún paquete y el usuario toca el botón “aceptar”, la aplicación mostrará un mensaje especificando que debe seleccionar un paquete. Si el usuario selecciona varios paquetes y el usuario toca el botón “aceptar”, la aplicación mostrará un mensaje especificando que debe seleccionar solamente un paquete.

Prototipo de interfaz:

Visor de paquetes de contenido educativo para dispositivos móviles



Figura 14: Importar un paquete SCORM. Elaboración propia.



Figura 15: Seleccionar formato en que se encuentra el paquete SCORM a importar. Elaboración propia.

Visor de paquetes de contenido educativo para dispositivos móviles

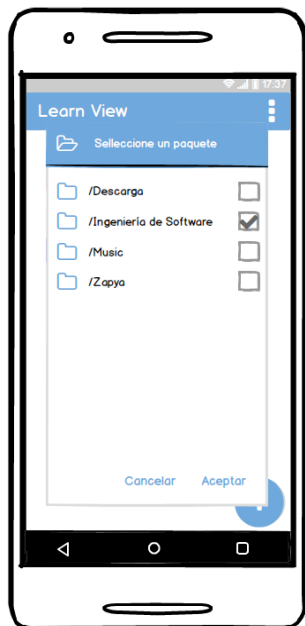


Figura 16: Seleccionar paquete en formato de carpeta dentro del dispositivo. Elaboración propia.

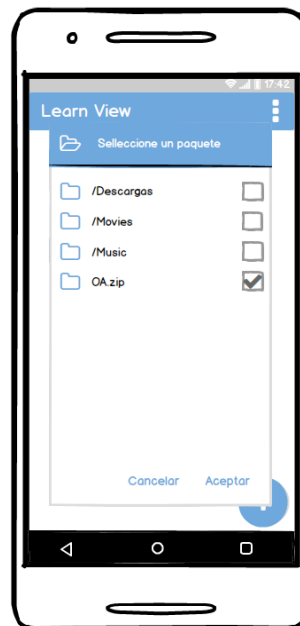


Figura 17: Seleccionar paquete en formato comprimido dentro del dispositivo. Elaboración propia.

Tabla 7: Historia de Usuario Mostrar paquete SCORM importado. Elaboración propia.

Número: 3	Nombre del requisito: Mostrar paquete SCORM importado.
Programador: Beatriz Ontivero González	Iteración Asignada: 5
Prioridad: Alta	Tiempo Estimado: 7 días.
Riesgo en Desarrollo: <ul style="list-style-type: none"> ✓ Interrupción en el fluido eléctrico. ✓ Rotura de la estación de trabajo. ✓ Configuración de la estación de trabajo. 	Tiempo Real: 5 días.

Visor de paquetes de contenido educativo para dispositivos móviles

Descripción:

Luego de que el usuario importe el paquete *SCORM*, la aplicación mostrará una vista conformada por los siguientes datos del paquete:

- Ícono: Muestra el ícono representando un paquete *SCORM*.
- Nombre: Muestra el nombre del paquete.

Si el usuario selecciona el paquete, la aplicación mostrará una vista que contendrá un índice de contenidos del paquete *SCORM* seleccionado.

Observaciones:

Si el paquete que busco no es un paquete con estándar *SCORM*, este no se mostrará en la lista de paquetes.

Prototipo de interfaz:

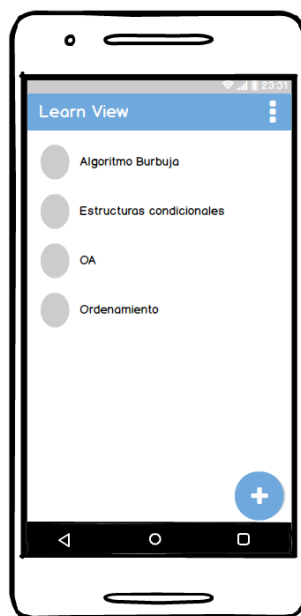


Figura 18: Paquete importado. Elaboración propia.

Tabla 8: Historia de Usuario Mostrar contenidos del paquete SCORM. Elaboración propia.

Visor de paquetes de contenido educativo para dispositivos móviles

Número: 5	Nombre del requisito: Mostrar contenidos del paquete SCORM.
Programador: Beatriz Ontivero González	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 15 días
Riesgo en Desarrollo: <ul style="list-style-type: none">✓ Interrupción en el fluido eléctrico.✓ Rotura de la estación de trabajo.✓ Configuración de la estación de trabajo.	Tiempo Real: 10 días
Descripción: <p>Cuando el usuario seleccione un elemento del índice de contenido del paquete SCORM; si es un título y no contiene contenido alguno, la aplicación cerrará el menú y mostrará el contenido del siguiente elemento; de lo contrario, se cerrará el menú, y se visualizará el contenido seleccionado, permitiéndole al usuario la navegación por el mismo.</p> <p>La vista muestran dos botones representados por imágenes, indicando la navegación hacia el siguiente contenido y hacia el contenido anterior. En el momento en que se está visualizando el primer contenido del paquete, solo se mostrará el botón de hacia adelante. En el momento en que se está visualizando el último contenido, solo se mostrará el botón de hacia atrás. En ninguno de los dos casos anteriores, se mostrarán los dos botones.</p>	

Visor de paquetes de contenido educativo para dispositivos móviles

Observaciones:

Si el usuario selecciona afuera del menú y no ha seleccionado ningún contenido anteriormente, la aplicación cerrará el menú, y mostrará una vista vacía.

Si el usuario selecciona afuera del menú y ya ha seleccionado al menos un contenido anteriormente, la aplicación cerrará el menú, y mostrará una vista con la información del contenido previamente mostrada.

Prototipo de interfaz:

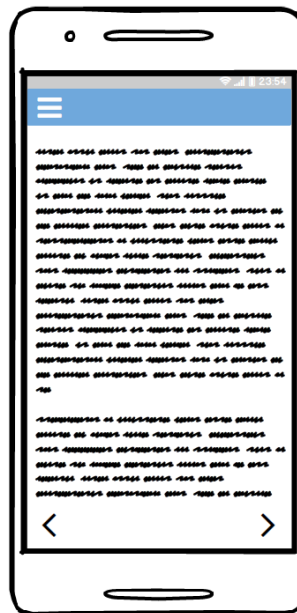


Figura 19: Visor de contenido educativo. Elaboración propia.

Tabla 9: Historia de Usuario Seleccionar directorio de almacenamiento de paquetes. Elaboración propia.

Número: 6	Nombre del requisito: Seleccionar directorio de almacenamiento de paquetes.
Programador: Beatriz Ontivero González	Iteración Asignada: 7

Visor de paquetes de contenido educativo para dispositivos móviles

Prioridad: Media	Tiempo Estimado: 7 días
Riesgo en Desarrollo: <ul style="list-style-type: none">✓ Interrupción en el fluido eléctrico.✓ Rotura de la estación de trabajo.✓ Configuración de la estación de trabajo.	Tiempo Real: 5 días
Descripción: <p>Cuando se abre la aplicación por primera vez, la aplicación debe mostrar una vista que le indique al usuario que es necesario que seleccione un directorio fijo para almacenar de forma organizada los paquetes SCORM dentro del dispositivo, posibilitando una mejor administración de los paquetes importados en la aplicación. Esta vista contendrá los siguientes elementos:</p> <ul style="list-style-type: none">• Ícono: Muestra un icono que le representa al usuario “información”.• Mensaje: Muestra un mensaje expresando que el usuario debe seleccionar un directorio de almacenamiento antes.• Aceptar: Muestra un botón con el nombre aceptar, que si se presiona indica que el usuario está de acuerdo con el mensaje mostrado. <p>Si el usuario presiona afuera de la vista, esta se cierra y la aplicación mostrará la vista anterior.</p> <p>Si el usuario presiona el botón aceptar, la aplicación mostrará una vista que contiene un explorador de archivos con la posibilidad de que el usuario pueda seleccionar el directorio que desee.</p> <p>Al seleccionarlo, si el usuario presiona el botón “aceptar”, la aplicación guardará la dirección del directorio seleccionado, se cerrará el explorador y se mostrará la vista que contiene el listado de paquetes; de lo contrario, si selecciona “cancelar”, se cerrará el explo-</p>	

Visor de paquetes de contenido educativo para dispositivos móviles

rador y se mostrará la vista que contiene el listado de paquetes.

Observaciones:

En el explorador de archivo, solo podrá seleccionar un solo directorio, de seleccionar más de una, se le mostrará al usuario un mensaje notificándole que no es una acción permitida.

Si el usuario presiona “aceptar” sin haber seleccionado un directorio, se mostrará un mensaje especificando que se debe seleccionar un directorio.

Si el usuario presiona “cancelar”, se cerrará la vista y se mostrará el listado de paquetes.

Si el usuario selecciona un paquete que no posea el estándar SCORM, la aplicación mostrará una vista informándole al usuario que el paquete seleccionado no es un paquete SCORM.

Si el usuario selecciona un paquete que ya se encuentra importado, la aplicación mostrará una vista donde informa que ese paquete ya se encuentra importado.

Prototipo de interfaz:

Visor de paquetes de contenido educativo para dispositivos móviles

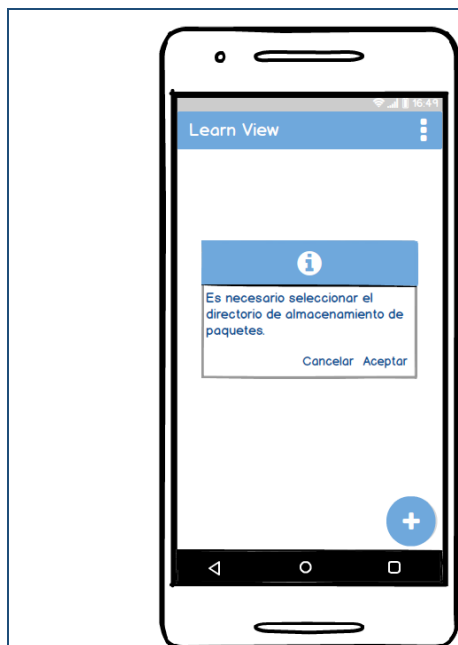


Figura 20: Mensaje para seleccionar el directorio de almacenamiento. Elaboración propia.

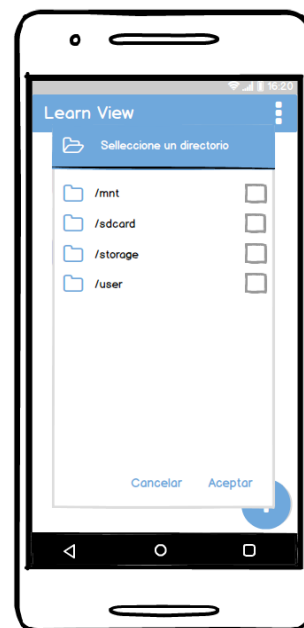


Figura 21: Seleccionar directorio de almacenamiento. Elaboración propia.

Tabla 10: Historia de Usuario Modificar directorio de paquetes. Elaboración propia.

Número: 7	Nombre del requisito: Modificar directorio de paquetes.	
Programador: Beatriz Ontivero González	Iteración Asignada: 8	
Prioridad: Baja	Tiempo Estimado: 10 días	
Riesgo en Desarrollo:	Tiempo Real: 7 días	
<ul style="list-style-type: none"> ✓ Interrupción en el fluido eléctrico. ✓ Rotura de la estación de trabajo. ✓ Configuración de la estación de trabajo. 		

Visor de paquetes de contenido educativo para dispositivos móviles

Descripción:

En la vista donde se encuentra el listado de paquetes, en la esquina superior derecha, la aplicación posee un ícono que contiene un conjunto de opciones, dentro de ellos, la opción: “Seleccionar directorio”.

Cuando el usuario presione esta opción, se mostrará una vista que contendrá la ruta del directorio fijo que el usuario escogió la primera vez que abrió la aplicación, y al lado derecho, se encuentra un botón “...” que permite cambiar ese directorio por uno nuevo que el propio usuario desee. La vista, además, posee, en la esquina superior derecha, un botón, que le posibilitará al usuario volver a la vista anterior; lo mismo sucederá se presiona el botón atrás del teléfono.

Al presionarlo, se mostrará un explorador de archivo que le posibilitará al usuario seleccionar un directorio nuevamente. Una vez seleccionado, si el usuario presiona el botón “aceptar”, la aplicación copiará los paquetes que se encontraban en el directorio antiguo para el nuevo; lo que posibilita que el usuario todavía pueda visualizar los paquetes que tenía previamente importados; de lo contrario, si el usuario presiona el botón “cancelar”, se cerrará la vista y se mostrará nuevamente la vista con el directorio escogido anteriormente. Finalmente, la ruta del directorio pasado se cambiará por la ruta actual.

Observaciones:

Si la primera vez que se abre la aplicación el usuario no seleccionó el directorio de almacenamiento, en la vista, en vez de mostrar la ruta, tendrá un mensaje especificando al usuario que seleccione un directorio fijo.

Prototipo de interfaz:

Visor de paquetes de contenido educativo para dispositivos móviles

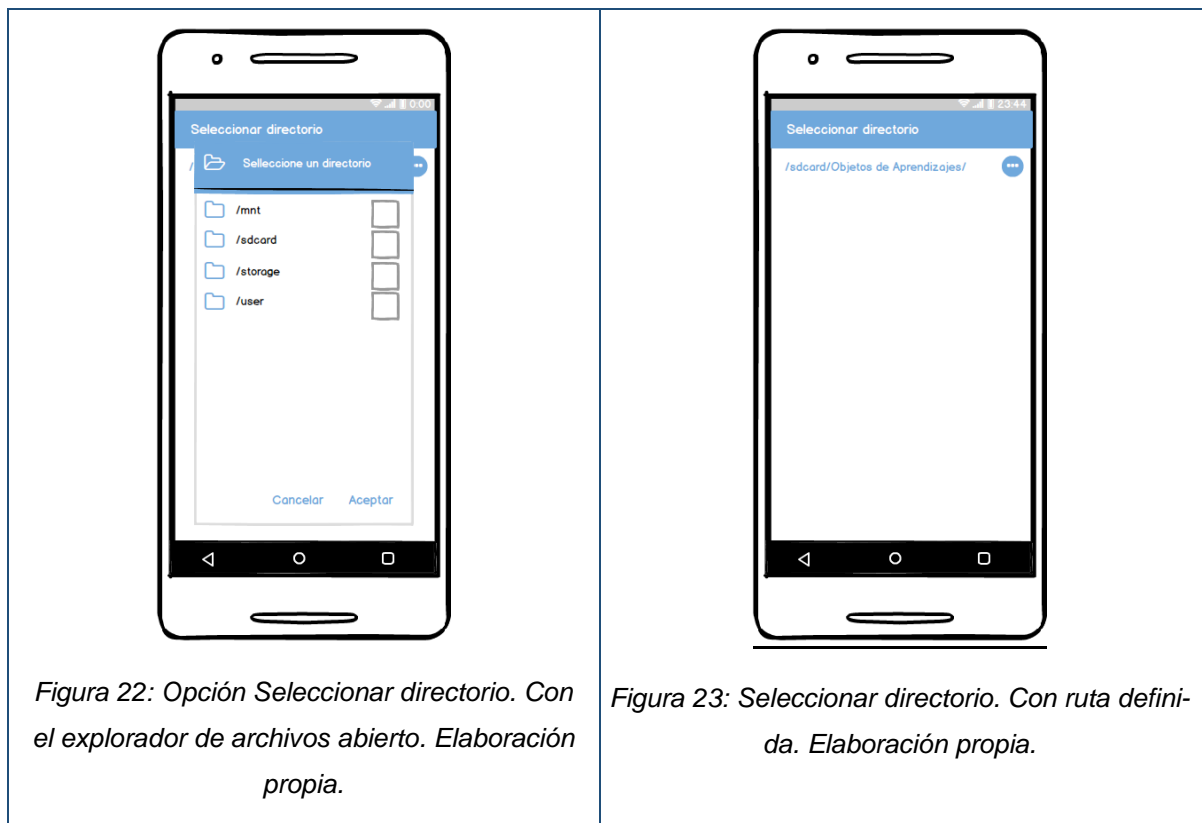


Tabla 11: Historia de Usuario Eliminar paquete SCORM de la aplicación. Elaboración propia.

Número: 8	Nombre del requisito: Eliminar paquete SCORM de la aplicación.		
Programador: Beatriz Ontivero González		Iteración Asignada: 9	
Prioridad: Media		Tiempo Estimado: 10 días	
Riesgo en Desarrollo:		Tiempo Real: días	
<ul style="list-style-type: none"> ✓ Interrupción en el fluido eléctrico. ✓ Rotura de la estación de trabajo. ✓ Configuración de la estación de trabajo. 			

Visor de paquetes de contenido educativo para dispositivos móviles

Descripción:

En la vista que contiene el listado de paquetes, en la esquina superior derecha, la aplicación posee un ícono que contiene un conjunto de opciones, dentro de ellos, la opción: “Eliminar”. Cuando el usuario presione esta opción, aparecerá un pequeño cuadro en la esquina superior derecha de cada paquete de la lista. Además, se visualizará una barra abajo, donde mostrará 3 opciones: seleccionar todos los paquetes, eliminar el o los paquetes seleccionados y una imagen que indica cancelar.

Si el usuario presiona la opción de seleccionar todos, se mostrarán todos los paquetes del listado con una palomita arriba del cuadrado. Si después de esto, el usuario lo vuelve a presionar, se desmarcarán todos.

Si el usuario presiona la opción de cancelar, la aplicación ocultará la barra que contiene las opciones de eliminar, y también los cuadrados que se encontraban arriba de cada paquete.

Si el usuario presiona la opción de eliminar, la aplicación mostrará una vista que contendrá los siguientes elementos:

- Ícono: Muestra un ícono que le representa al usuario “información”.
- Mensaje: Muestra un mensaje que le pregunta al usuario si se encuentra seguro que desea eliminar el o los paquetes seleccionados anteriormente.
- Aceptar: Muestra un botón con el nombre aceptar, que si se presiona indica que el usuario está seguro con la opción seleccionada.
- Cancelar: Muestra un botón con el nombre cancelar, que si se presiona indica que el usuario no está seguro de la opción que seleccionó.

Si el usuario presiona el botón aceptar, la aplicación eliminará el o los paquetes seleccionados de la aplicación y del dispositivo; y se ocultará el cuadrado que se encontraba arriba del paquete y se actualizará el listado. Si el usuario presiona el botón cancelar, se cerrará la vista, volviéndose a mostrar el listado de los paquetes todavía seleccionados.

Visor de paquetes de contenido educativo para dispositivos móviles

Si el usuario mantiene presionado por unos segundos el paquete, la aplicación mostrará el mensaje explicado anteriormente.

Observaciones:

Si el usuario elimina uno o varios paquetes, deberá actualizar la lista deslizando el dedo por la pantalla, de arriba hacia abajo; para que se actualice la vista.

Prototipo de interfaz:

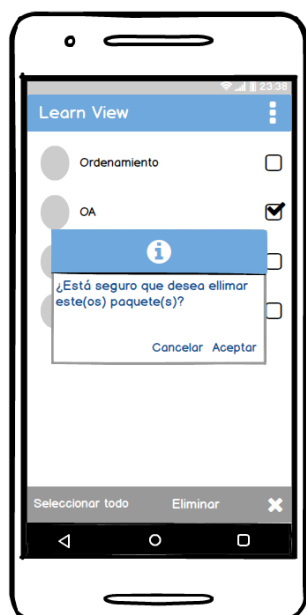


Figura 24: Mensaje de verificación de eliminar.
Elaboración propia.

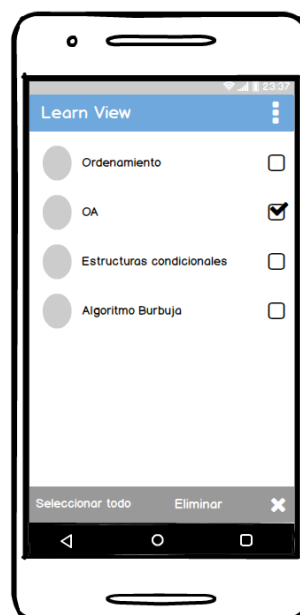


Figura 25: Eliminar. Con elemento marcado. Elaboración propia.

Tabla 12: Historia de Usuario Ordenar listado de paquetes SCORM importados. Elaboración propia.

Numero: 9	Nombre del requisito: Ordenar listado de paquetes SCORM importados.
Programador: Beatriz Ontivero González	Iteración Asignada: 11

Visor de paquetes de contenido educativo para dispositivos móviles

Prioridad: Baja	Tiempo Estimado: días
Riesgo en Desarrollo: <ul style="list-style-type: none">✓ Interrupción en el fluido eléctrico.✓ Rotura de la estación de trabajo.✓ Configuración de la estación de trabajo.	Tiempo Real: 7 días
Descripción: <p>En la esquina superior derecha, la aplicación posee un ícono que contiene un conjunto de opciones, dentro de ellos, la opción: “ordenar”. Cuando el usuario presiones esta opción, la aplicación le mostrará la lista de paquetes importados en la misma vista, pero ordenados alfabéticamente (A-Z y Z-A).</p>	
Observaciones: <p>Por defecto, la lista de paquetes estará ordenada en orden A-Z. Cuando el usuario seleccione la opción ordenar y la lista este seleccionada en el orden A-Z, la aplicación ordenará la lista en el orden Z-A; y viceversa.</p>	
Prototipo de interfaz:	

Visor de paquetes de contenido educativo para dispositivos móviles

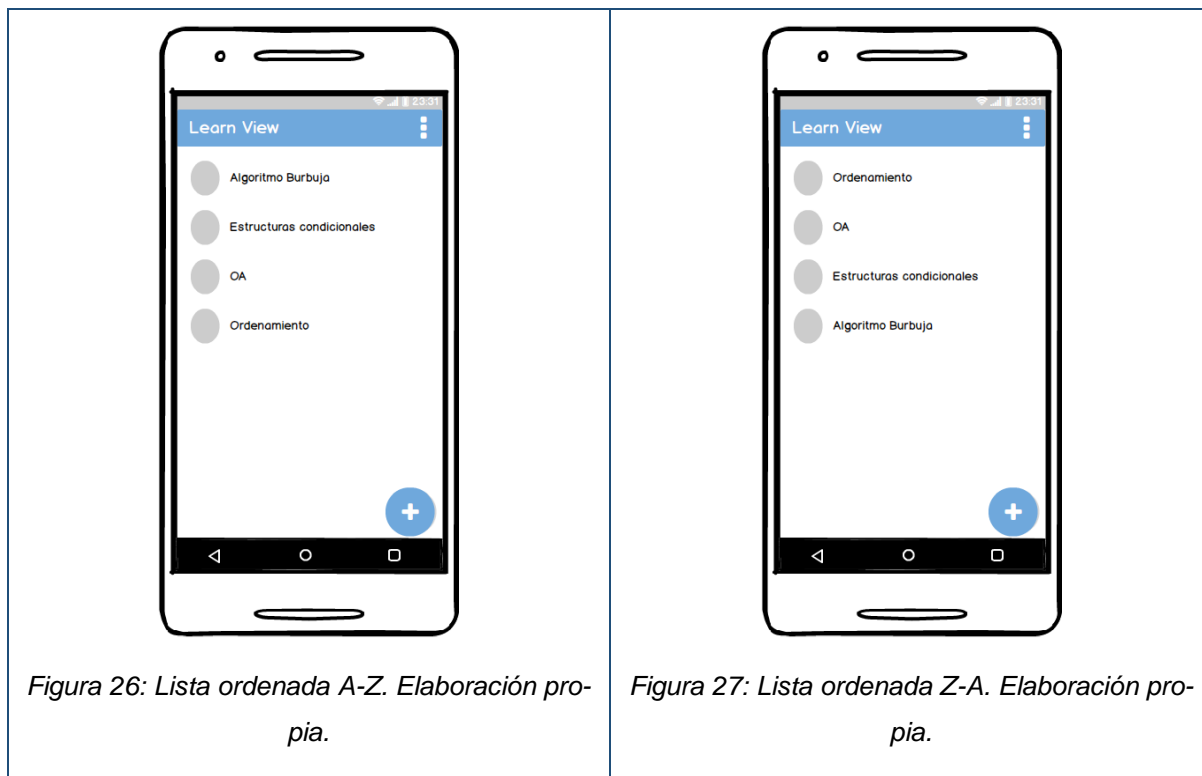


Figura 26: Lista ordenada A-Z. Elaboración propia.

Figura 27: Lista ordenada Z-A. Elaboración propia.

Tabla 13: Historia de Usuario Buscar paquete SCORM de la lista de paquetes. Elaboración propia.

Número: 10	Nombre del requisito: Buscar paquete SCORM de la lista de paquetes.	
Programador: Beatriz Ontivero González	Iteración Asignada: 10	
Prioridad: Baja	Tiempo Estimado: 10 días	
Riesgo en Desarrollo: <ul style="list-style-type: none"> ✓ Interrupción en el fluido eléctrico. ✓ Rotura de la estación de trabajo. ✓ Configuración de la estación de trabajo. 	Tiempo Real: 5 días	

Visor de paquetes de contenido educativo para dispositivos móviles

Descripción:

La aplicación debe mostrar al usuario la opción de buscar un paquete *SCORM* deseado dentro del menú ubicado en la interfaz del listado de paquetes importados en la aplicación, en la esquina superior derecha. Cuando el usuario seleccione la opción buscar, la aplicación mostrará una vista. Esta vista contiene los siguientes elementos:

- Campo de texto: Muestra un campo de texto para que el usuario pueda escribir el nombre del paquete que desea buscar.
- Buscar: Muestra un botón con una imagen con una lupa que indica al usuario la opción de buscar.
- Resultado: Muestra una lista con el resultado de la búsqueda.

El usuario debe escribir el nombre del paquete que desea buscar. Al terminar debe tocar el botón, y se mostrará debajo una lista con los paquetes que coincidan con la búsqueda del usuario ordenada por la cantidad de coincidencia de los encontradas. La búsqueda ignora las mayúsculas y minúsculas; y no exige poner el nombre exacto.

Si el usuario presiona un paquete del resultado de la búsqueda, la aplicación mostrará una vista que contendrá un índice de contenidos del paquete *SCORM* seleccionado.

Observaciones:

Si el usuario presiona el botón de buscar sin haber escrito el nombre, la aplicación mostrará un mensaje indicando al usuario que el campo de texto no debe de estar vacío.

Si el nombre escrito por el usuario no coincide con ninguno de los paquetes, la aplicación mostrará un mensaje indicando al usuario que no se encontró el paquete con ese nombre.

Prototipo de interfaz:

Visor de paquetes de contenido educativo para dispositivos móviles

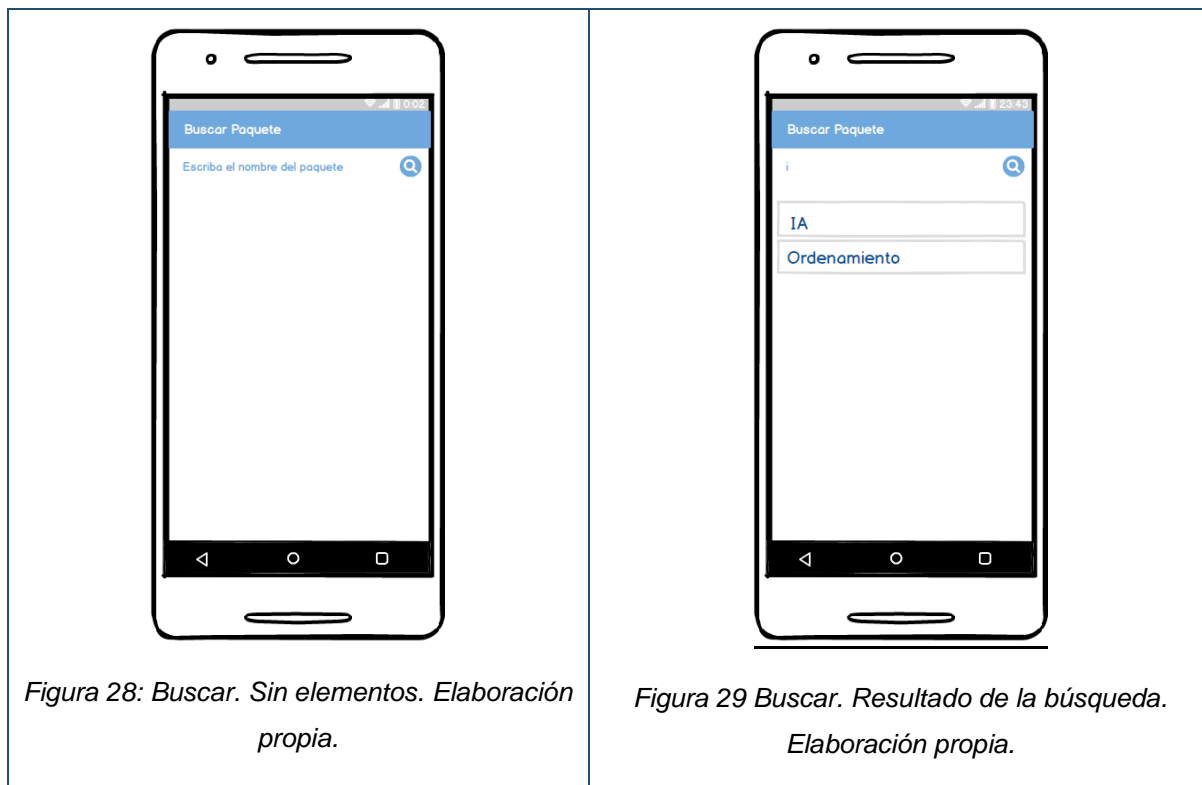


Figura 28: Buscar. Sin elementos. Elaboración propia.

Figura 29 Buscar. Resultado de la búsqueda. Elaboración propia.

Tabla 14: Historia de Usuario Importar paquete SCORM desde URL. Elaboración propia.

Número: 11	Nombre del requisito: Importar paquete SCORM desde URL.	
Programador: Beatriz Ontivero González	Iteración Asignada: 12	
Prioridad: Alta	Tiempo Estimado: 10 días	
Riesgo en Desarrollo: <ul style="list-style-type: none"> ✓ Interrupción en el fluido eléctrico. ✓ Rotura de la estación de trabajo. ✓ Configuración de la estación de trabajo. 	Tiempo Real: 7 días	

Visor de paquetes de contenido educativo para dispositivos móviles

Descripción:

Cuando el usuario selecciona la opción de importar paquete desde una *URL*, la aplicación muestra al usuario una vista donde debe escribir la *URL* donde se encuentra el paquete *SCORM* a importar. Esta vista contiene los siguientes elementos:

- **Título:** Muestra el título de la vista, esta indica al usuario la opción escogida.
- **Mensaje:** Muestra un mensaje con el nombre *URL*, que indica al usuario que debe hacer en esa vista.
- **Url:** Muestra un campo de texto donde el usuario deberá escribir la *URL* donde desea importar el paquete.
- **Aceptar:** Muestra un botón con el nombre aceptar que indica que la aplicación importará el paquete desde la *URL* escrita anteriormente.
- **Cancelar:** Muestra un botón con el nombre cancelar que indica que el usuario ya no desea realizar esa opción.

Si el usuario presiona la opción cancelar, se cerrará la vista y se mostrará la vista del listado de paquetes.

Si el usuario presiona la opción de aceptar, la aplicación deberá descargar el paquete para luego importarlo a la aplicación.

Visor de paquetes de contenido educativo para dispositivos móviles

Observaciones:

El usuario debe haber seleccionado la opción de importar paquete desde *URL* en la vista anterior.

La aplicación para poder importar el paquete desde una *URL*, esta debe ser pública; de lo contrario, la aplicación mostrará un mensaje indicando al usuario los posibles errores que pueden haber ocurrido. Estos errores serán:

- La *URL* escrita por el usuario no es correcta.
- La *URL* escrita por el usuario no es pública.
- El dispositivo no tiene conexión.

Prototipo de interfaz:



Figura 30: Importar paquete SCORM. Elaboración propia.

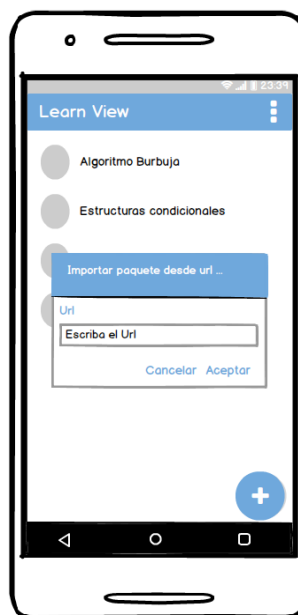


Figura 31: Importar paquete desde Url. Elaboración propia.