



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 1**

**Centro de Software Libres (CESOL)**

**Tema: Navegador de archivos para NovaDroid.**

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

**Autor:**

**Alvaro Castro Pérez**

**Tutores:**

**MSc. Juan Manuel Fuentes Rodríguez**

**Ing. Estela Odelsa Martín Coronel**

**La Habana, mayo de 2022**

**“Año 62 de la Revolución”**



*“El Software es una gran combinación entre el arte e ingeniería”*

*Bill Gates*

## DECLARACIÓN DE AUTORÍA

---

Declaro ser autor de la presente tesis que tiene por título: Navegador de archivos para NovaDroid y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Alvaro Castro Pérez

MSc. Juan Manuel Fuentes Rodríguez

---

Firma del Autor

---

Firma del Tutor

Ing. Estela Odelsa Martín Coronel

---

Firma del Tutor

## DATOS DE CONTACTO

---

### **Autor:**

Alvaro Castro Pérez.

Universidad de las Ciencias Informáticas (UCI)

e-mail: [acastro@estudiantes.uci.cu](mailto:acastro@estudiantes.uci.cu)

### **Tutores:**

Juan Manuel Fuentes Rodríguez

Universidad de las Ciencias Informáticas (UCI)

e-mail: [jrfuentes@uci.cu](mailto:jrfuentes@uci.cu)

Estela Odelsa Martín Coronel

Universidad de las Ciencias Informáticas (UCI)

e-mail: [eomartin@uci.cu](mailto:eomartin@uci.cu)

## DEDICATORIA

---

*Dedico este trabajo a toda mi familia, especialmente a mis padres y hermanos que gracias a ellos soy todo lo que soy en esta vida. A mi amada madre que es lo más grande que tengo, que solo ella encuentra la manera para darme apoyo en los momentos más oscuros y difíciles, esa que con su forma de ser tan especial es capaz de dibujarme una sonrisa en el rostro y con su amor hacerme entender que todo estará bien. A mi padre que aunque la vida nos llevó por distintos caminos de igual forma le tengo un cariño muy especial y un respeto inmenso. A todos mis hermanos que de una forma u otra se han preocupado por mi desde el inicio de esta aventura iniciada ya hace 5 años. A todos mis amigos, los verdaderos, los que me acompañan en los momentos buenos y malos. A mis compañeros de aula que solo ellos fueron capaces de regalarme los momentos más divertidos e inolvidables en la UCI. A mi novia Dailen que de todos los mencionados es la más cercana a este proceso, ella que supo darme calma cuando lo necesitaba y alentarme día a día, a ella que hace de mi mundo un lugar perfecto, gracias por todo mi amor.*

## AGRADECIMIENTOS

---

Quiero agradecer a todos los que creyeron y confiaron en mí desde el principio. Madre gracias por ser la mejor madre del mundo, gracias por estar ahí cuando lo necesite, gracias por ser como eres, por educarme, por llevarme por el buen camino, por ser tan divertida y especial, gracias por todo el esfuerzo que hiciste por mí, gracias por ser madre y padre, por ti es que me esfuerzo cada día por ser alguien mejor. Te amo mamá, gracias por todo.

Papá decirte que espero que estés orgulloso de mi, del hombre en que me he convertido. Mi meta es no defraudarlos y está siendo cumplida. Eres ejemplo a seguir para todo el que te rodea, siempre estuvo ahí cuando hizo falta, no importa lo lejos que vivieras, siempre estabas y no sólo conmigo, sino también con mis hermanos.

Mamá, papá, espero que estén orgullosos de mí, gracias por darme la vida y por darme los hermanos que tengo, ese es el mejor regalo que pudieron hacerme. A ellos quiero agradecerles también porque son parte de mi vida y muy importantes para mí.

Primeramente a mi hermano Maikel, mi compañero de vida cuando no estoy en la escuela, que aunque a veces soy bien duro contigo y parezco bien distante te quiero con mi vida. Que siempre pido sensatez para ti, sabiduría y lo mejor del mundo, que a pesar de todos tus defectos eres un ser humano con un corazón gigante y muy buenos sentimientos. Gracias.

Quiero agradecer a mi hermano, padre, amigo e infinitudes de cosas más Alexis, que no me alcanzan tres vidas para devolverte todo lo que has hecho por mí. Que desde muy temprana me acogiste con un cariño muy especial dándome a entender el ser humano que eres. Gracias por ser tan atento, por seguir esta etapa de mi vida desde bien cerca, por siempre insistir en el estudio, aún recuerdo cuanta ilusión te hacía verme estudiar en el IPVCE la Lenin como tu hiciste. Gracias por todo, por los buenos consejos y por todo lo que me has enseñado, te quiero guacho. Gracias.

Quiero agradecer a mi súper cuñada Dayamis, por ser uno de los seres humanos con el alma más transparente y bella que conozco. Agradecerte por todo el cariño y afecto con que me has recibido en tu vida, por las horas de risa y momentos divertidos. Por ser tan especial con todo el que te rodea, por ser sencillamente tú la mejor cuñada del mundo mundial. Te quiero. Gracias.

A mis hermanos Juvenal, Jorge Luis y Xiomara que siempre estuvieron ahí, lejos pero cerca, gracias por todo el apoyo incondicional, por desearme todo el éxito que pueda alcanzar en esta vida, por ser tan atentos, por formar parte de mi vida, a todos les estoy sinceramente agradecido. Los quiero. Gracias.

Quiero agradecer especialmente a mi novia Dailen, gracias por estar siempre ahí, por pasar tantos momentos juntos, buenos y malos, por las risas, los mimos, las caricias, por darme el amor más puro y sincero. Gracias por ser mi más fiel compañera y amiga por compartir conmigo parte de tu vida,

## AGRADECIMIENTOS

---

*gracias por enseñarme tantas cosas, por sacar la mejor versión de mí. Gracias por todo, mi amor, al lado tuyo siento que soy mejor persona. Te amo. Gracias.*

*Agradecer a David, Yuriem, Miriam, Yudit, Sandy, Darien y Yanet que son grandísimos seres humanos, gracias a cada uno de ustedes por todo el apoyo y la preocupación. Decirles que son muy importantes para mí, que son una hermosa familia que la vida me regalo y me siento feliz por ello, espero disfrutar de todos muchos años más y compartir a su lado muchos recuerdos. Gracias.*

*A mi amigo de toda la vida Andy, que más que un amigo es un hermano de otra madre, gracias por los buenos y malos momentos, por las risas, por los consejos. Gracias por ser tan especial en mi vida, por el apoyo que me diste en este proceso, la preocupación, por ser incondicional conmigo, que se cuento contigo a ciegas, gracias por tu amistad. Gracias.*

*Agradecer a cada uno de mis compañeros de aula, que fueron los mejores amigos que pude haberme encontrado en la UCI, gracias por cada momento de alegría, por las noches de desvelo y largas conversaciones. Agradecerles a todos ya que tuvieron la capacidad que acogirme en sus vidas con un cariño sincero, gracias por juntos conformar esa bella familia y ayudarme a vivir esos cinco años de manera inigualable. Gracias.*

*Agradecer a mis dos tutores Estela y Juan Manuel por toda la ayuda brindada para que este trabajo de diploma obtuviera la mejor calidad posible. Gracias por todos los consejos tanto teóricos como prácticos, gracias por guiarme durante esta investigación. Me llevo buen recuerdo de ambos fue un placer ser su tesista. Gracias.*

*A todos Gracias...*

## RESUMEN

---

NovaDroid es un sistema operativo móvil desarrollado en la Universidad de las Ciencias Informáticas, entidad que actualmente se encuentra desarrollando una nueva variante de este sistema operativo construyéndolo para computadoras de escritorios. Consta de una personalización de Android que tiene como objetivo adaptarlo a las necesidades de los usuarios en Cuba. En la actualidad está siendo instalado y distribuido en las tabletas, laptops producidas por la Empresa Industrial para la Informática, las Comunicaciones y la Electrónica producto de la alianza estratégica UCI-GEDEME. NovaDroid para computadoras no permite la navegación de archivos, imposibilitando así el acceso a los directorios y archivos del sistema operativo. Debido a esta problemática se realizó la presente investigación con el objetivo de desarrollar una aplicación que permita navegar por la estructura de carpetas y archivos de NovaDroid con una estructura de información similar a la de los navegadores de archivos de computadoras de escritorio. Se utilizó UML como lenguaje de modelado, Visual Paradigm como herramienta de modelado, Java como lenguaje de programación y Android Studio como entorno de desarrollo. Para regir el desarrollo de la solución se utilizó la metodología AUP-UCI, adaptada a los procesos productivos de la universidad.

**Palabras clave:** archivos, carpetas, navegación, NovaDroid, sistema de archivos.

## ***ABSTRACT***

---

NovaDroid is a mobile operating system developed at the University of Computer Science, an entity that is currently developing a new variant of this operating system by building it on desktop computers. It consists of an Android customization that aims to adapt it to the needs of users in Cuba. It is currently being installed and distributed in tablets, laptops produced by the Industrial Enterprise for Information Technology, Communications and Electronics product of the strategic alliance UCI-GDEM. NovaDroid for computers does not allow file browsing, thus preventing access to folders and files of the operating system. Due to this problem, the present investigation was carried out with the objective of developing an application that allows browsing the folder and file structure of NovaDroid with an information structure similar to that of file browsers of desktop computers. UML was used as a modeling language, Visual Paradigm as a modeling tool, Java as a programming language and Android Studio as a development environment. The AUP-UCI methodology was used to govern the development of the solution, adapted to the university's production processes.

**Keywords:** file, folder, navigation, NovaDroid, file system.

# ÍNDICE

---

Introducción.....	1
<b>CAPÍTULO 1. Fundamentos de la investigación.....</b>	<b>6</b>
<b>1.    Introducción.....</b>	<b>6</b>
<b>1.1.    Conceptos asociados al dominio del problema.....</b>	<b>6</b>
1.1.1 Metáfora de escritorio.....	6
1.1.2 Navegador de Archivos.....	6
1.1.3 Carpetas o Directorios.....	7
1.1.4 Archivos.....	7
<b>1.2.    Sistema operativo Android.....</b>	<b>8</b>
1.2.1 Características de Android:.....	12
1.2.2 Ventajas y Desventajas de Android:.....	13
<b>1.3.    Estudio de soluciones similares.....</b>	<b>14</b>
1.3.1 ES Explorador de Archivos.....	14
1.3.2 ASTRO.....	14
1.3.3 FX File Explorer.....	15
1.3.4 Amaze File Manager.....	16
1.3.5 Simple File Manager.....	16
FX File Explorer.....	18
1.3.6 Conclusiones del estudio de las posibles soluciones.....	19
<b>1.4.    Selección del lenguaje, tecnologías y herramientas.....</b>	<b>19</b>
1.4.1 Lenguaje de programación.....	19
1.4.2 Lenguaje de modelado.....	20
1.4.3 Herramientas.....	20
<b>1.5.    Metodología de desarrollo.....</b>	<b>21</b>
1.5.1 Metodología de desarrollo de software AUP-UCI.....	22
<b>1.6    Conclusiones Parciales.....</b>	<b>23</b>
<b>CAPÍTULO 2. Análisis y Diseño del navegador de archivo para NovaDroid.....</b>	<b>24</b>
<b>2.1.    Introducción.....</b>	<b>24</b>
<b>2.2.    Propuesta de solución.....</b>	<b>24</b>
<b>2.3.    Captura de requisitos.....</b>	<b>24</b>
2.3.1 Técnicas para la captura de requisitos.....	24
2.3.2 Requisitos funcionales (RF).....	25

# ÍNDICE

---

2.3.3	Requisitos no funcionales (RNF)	27
2.4.	Historias de usuario	28
2.5.	Diseño	35
2.5.1	Descripción de la arquitectura	35
2.5.2	Patrones de diseño	38
2.5.3	Diagrama de clases del diseño	41
2.6.	Conclusiones parciales	42
<b>CAPÍTULO 3. Implementación y prueba del navegador de archivo para NovaDroid</b>		<b>44</b>
3.1.	Introducción	44
3.2.	Código fuente	44
3.2.1	Estándares de codificación	44
3.3.	Diagrama de despliegue	46
3.4.	Validación del sistema	47
3.4.1	Pruebas de unidad	47
3.4.2	Pruebas de validación	48
3.4.3	Pruebas de aceptación	50
3.4.4	Aplicación y resultados de las pruebas	50
3.5.	Conclusiones parciales	52
<b>CONCLUSIONES GENERALES</b>		<b>54</b>
<b>RECOMENDACIONES</b>		<b>55</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b>		<b>56</b>

## Introducción

Los primeros navegadores de archivos fueron creados para sistemas operativos con interfaces de usuario de líneas de comandos (*Comand Line Interface*, CLI). Generalmente representaban las unidades, particiones y directorios en su distribución física real y permitían un número limitado de operaciones sobre estos recursos. El primer navegador de archivos visual que se desarrolló (aunque aún en modo texto) fue Dired, que sentó las bases para los gestores de archivos que surgieron a continuación. Con la introducción de las interfaces gráficas, los navegadores de archivos adquirieron diferentes funcionalidades, como la habilidad de asociar tipos de archivos a programas, y facilitaron la comprensión de conceptos mediante la representación gráfica de cada recurso, identificado con un icono (Howling 2019).

Un administrador de archivos, gestor de archivos o navegador de archivos es un programa informático que proporciona una interfaz de usuario para administrar archivos y directorios. Las operaciones más comunes realizadas en archivos o grupos de archivos incluyen crear, abrir, ver, reproducir, editar, imprimir, cambiar nombre, copiar, mover, eliminar y buscar archivos; así como modificar sus atributos, propiedades y permisos de acceso. Las carpetas y los ficheros pueden visualizarse en un árbol jerárquico basado en su estructura de directorios. Algunos navegadores de archivos contienen características inspiradas por navegadores web, como los botones de navegación adelante y atrás (Howling 2019).

En la actualidad el navegador de archivos de un sistema operativo es una herramienta sumamente importante, ya que sin él, se vería afectada la navegabilidad del usuario por la inmensa estructura de archivos y carpetas, tanto la del sistema como la estructura que el propio usuario ha creado en torno a sus necesidades y prioridades. Hoy día los sistemas operativos incorporan un navegador de archivos precisamente para que almacenar y encontrar los archivos sea una tarea sencilla y transparente (Informática 2019).

Android es un sistema operativo desarrollado por Google, basado en el kernel de Linux y otros software de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes, automóviles y televisores (ICTEA 2019). Este sistema operativo cuenta con grandes ventajas entre las cuales se puede mencionar:

- Actualmente existen más de 3 000.000 aplicaciones disponibles para teléfonos Android, aproximadamente 2/3 de estas son gratis.

- El sistema Android es capaz de hacer funcionar a la vez varias aplicaciones y además se encarga de gestionarlas, dejarlas en modo suspensión si no se utilizan e incluso cerrarlas si llevan un periodo determinado de inactividad.
- El código de Android es abierto, lo que permite adaptar este a bastantes dispositivos además de teléfonos celulares, incluso existe una versión de Android para Computadoras Personales (PC, por sus siglas del inglés Personal Computer).

Desde hace varios años Cuba está inmersa en una migración al uso del software libre, una tarea que se está llevando a cabo hoy día, lográndose ver en varias instituciones del país. La Universidad de las Ciencias Informáticas (UCI) es una de las instituciones encargadas de impulsar este proceso, donde una de sus ramas más especializadas está dedicada a la investigación y desarrollo de aplicaciones informáticas para satisfacer las necesidades de la migración a plataformas de código abierto. El centro de producción encargado de esta tarea es el Centro de Software Libres (CESOL). Entre los principales productos sobresalientes de la línea de investigación y desarrollo a las que se dedica este centro se puede mencionar NovaDroid.

NovaDroid es una personalización del sistema operativo Android, desarrollada a partir del código fuente y adaptada a las necesidades y realidades de la sociedad cubana. Una de sus principales características es que gracias a la reutilización de uno de los sistemas operativos más usados en el mundo, NovaDroid es compatible con más de 1 millón de aplicaciones disponibles en las diferentes tiendas de aplicaciones de Android. NovaDroid trae consigo aplicaciones desarrolladas por la industria de software cubana, incluye aplicaciones básicas de software libre que le serán útiles en el trabajo cotidiano entre las cuales destacan Firefox, OmniNotes, Lector de documentos (UCI 2019).

Actualmente CESOL se encuentra desarrollando NovaDroid en una nueva variante para los ordenadores, tarea que se enfrenta a grandes retos, dentro de los cuales se pueden mencionar:

- NovaDroid no pudiera sustituir a los sistemas operativos de las computadoras tradicionales, porque existen un gran número de herramientas que necesitan equipos potentes para su buen funcionamiento.
- Una de las principales deficiencias es que la experiencia de usuario brindada por Android no es igual a la metáfora de escritorio seguida por los sistemas operativos para computadora de mesas.
- El uso de este sistema operativo en computadoras de escritorio no se ha extendido debido a que este fue construido para ser explotado en dispositivos con pantallas pequeñas.

En el desarrollo del sistema operativo, se vio evidenciado que uno de las grandes dificultades del usuario en el uso de NovaDroid es la navegación por la estructura de archivos y directorios. A continuación se definen algunas deficiencias que presentan los navegadores de archivos de Android por los que estos no pueden ser instalados en NovaDroid para computadoras de escritorio:

- Estos fueron diseñados para ser usados con pantallas táctiles y gestos, lo que implica que al usuario interactuar a partir del mouse con el navegador de archivo, sus funcionalidades en la mayoría se ven limitadas.
- Los navegadores de archivos como la mayoría de las aplicaciones de Android están diseñadas para dispositivos con pantallas pequeñas, al ser desarrolladas para pantallas pequeñas hacen uso de diferentes técnicas de diseño de interfaces móviles, que no son válidas cuando se usa en pantallas grandes y disminuyen la usabilidad por no seguir la metáfora de escritorio.
- El diseño de estos no cuenta con una estructura de información similar a los navegadores de archivos de computadoras de escritorio.

Finalmente existe la necesidad de crear un navegador de archivo especial, con características específicas para que los usuarios convencionales se puedan adaptar con facilidad al nuevo ambiente de usuario, debido a que los existentes compatibles con Android poseen las deficiencias antes mencionadas.

Teniendo en cuenta lo antes planteado se identifica el siguiente **problema de la investigación**: ¿Cómo permitir la gestión de archivos de Android para computadoras?

Para dar respuesta a la problemática planteada la presente investigación centra su **objeto de estudio** los navegadores de archivos, delimitándose como **campo de acción** los navegadores de archivos para Android.

Se establece entonces como **objetivo general**: desarrollar una aplicación que permita navegar por los archivos para Android X86, con una estructura de información similar a la de los navegadores de archivos de computadoras de escritorio.

Este se desglosa en los siguientes **objetivos específicos**:

1. Investigar los aspectos teóricos fundamentales que sustenta el desarrollo de un navegador de archivos para NovaDroid.
2. Diseñar la aplicación que permita navegar por los archivos del sistema operativo NovaDroid.
3. Implementar la aplicación que permita la navegación por los archivos del sistema operativo NovaDroid.

4. Evaluar el correcto funcionamiento de la solución mediante la aplicación de pruebas.

Para darle solución al problema propuesto para esta investigación, se formularon las siguientes **preguntas científicas**:

1. ¿Cuáles son los referentes teóricos-metodológicos que sustentan el desarrollo de un navegador de archivo para NovaDroid?
2. ¿Cómo diseñar un navegador de archivo con una estructura de información similar a los navegadores de archivos de computadoras de escritorio para NovaDroid?
3. ¿Cómo implementar un navegador de archivo para NovaDroid?
4. ¿Cómo validar la aplicación Android para la navegación de archivos en NovaDroid?

Con el propósito de dar cumplimiento a lo previamente planteado se definen las siguientes **tareas de investigación**:

1. Definición de los elementos teóricos relacionados con el desarrollo de un navegador de archivo para Novadroid.
2. Análisis de diferentes soluciones informáticas para dar solución al tema de investigación en cuestión.
3. Selección de las tecnologías y la metodología más adecuada para el desarrollo de la solución.
4. Identificación de los requisitos de la solución a desarrollar.
5. Diseño de la aplicación que permita navegar por los archivos del sistema operativo NovaDroid.
6. Implementación de la aplicación móvil que permita navegar por los archivos del sistema operativo NovaDroid.
7. Aplicación de pruebas a la solución.
8. Evaluación del resultado obtenido.
9. Corrección de los posibles errores surgidos durante la aplicación de pruebas.

Para dar cumplimiento a las tareas de la investigación se utilizaron los siguientes métodos científicos:

#### **Métodos teóricos:**

- **Analítico – Sintético:** este método propició el proceso de construcción del marco teórico de la investigación, con la descomposición y síntesis asociado con la incorporación de un navegador de archivo para NovaDroid y su utilización en las computadoras de escritorio.
- **Modelación:** se utiliza para crear el proceso de diseño mediante la abstracción de sus elementos fundamentales utilizando un lenguaje de modelado y así desarrollar un modelo para la aplicación a desarrollar a partir de la situación problemática.

**Método empírico:**

- **Observación:** es utilizado para realizar un análisis de las diferentes soluciones que resuelven de forma parcial el problema de investigación planteado.
- **Entrevista:** es utilizado con el propósito de obtener información de especialistas acerca de los navegadores de archivos más utilizados en el mundo y en particular en las instituciones cubanas.

El presente documento está desglosado en los siguientes capítulos:

- **Capítulo I. Fundamentos de la investigación:** Estudio de los principales conceptos y elementos teóricos asociados a la investigación con el objetivo de lograr una mayor comprensión de la misma. Se aborda además el estudio de las soluciones ya existentes que resuelven de forma parcial el problema planteado. Se analiza el estado del arte de las mismas teniendo en cuenta sus ventajas y sus desventajas. Por último se establecen las tecnologías que sirven para el posterior desarrollo de la solución y se selecciona la metodología que guía el proceso de desarrollo.
- **Capítulo 2: Análisis y Diseño del navegador de archivos para NovaDroid:** En este capítulo se realiza la exploración e inicialización de la solución propuesta de acuerdo a la metodología de desarrollo de software. Se describe el modelo de dominio, que permite una mejor comprensión del negocio, los principales conceptos y sus relaciones. Se enumeran y detallan los requisitos funcionales y no funcionales, descritos en historias de usuario, generando una visión para la creación de la aplicación. Se escoge la arquitectura que se debe seguir para el desarrollo de la misma, la cual brindará una organización básica para la implementación.
- **Capítulo III. Implementación y Prueba del navegador de archivos para NovaDroid:** Se describen los artefactos relacionados con la implementación y las pruebas realizadas a la solución con el objetivo de validar su correcto funcionamiento.

## **CAPÍTULO 1. Fundamentos de la investigación**

### **1. Introducción**

En el presente capítulo se analizan los conceptos y elementos teóricos relacionados con el desarrollo de un navegador de archivos. Se plasma además el resultado del estudio realizado a las principales soluciones que resuelven de forma parcial el problema de investigación planteado, estableciendo con estos sistemas homólogos cierta comparación. Por último se describe la metodología seleccionada para guiar el proceso de desarrollo de software y las tecnologías que se seleccionan para el diseño e implementación de la solución.

#### **1.1. Conceptos asociados al dominio del problema**

##### **1.1.1 Metáfora de escritorio**

Una metáfora consiste en el uso de una analogía que puede presentarse en diversos formatos como puede ser una imagen, un símbolo o un texto, para referirse a un concepto que no es denotado de manera literal, sino en sentido figurado sobre la base de determinadas semejanzas entre el concepto y el elemento que se utiliza para representarlo. En diseño, una metáfora se emplea para tratar de lograr que la interacción sea más simple e intuitiva (Universidad Oberta de Catalunya 2017).

Hoy los usuarios tienen la percepción de sistema operativo que se funde en el concepto de escritorio. La potencia de los sistemas operativos modernos se mide en gran medida por la potencia de su interface gráfica de control. La mejora de las interfaces gráficas y la popularización de sistemas como el MacOS y Windows han contribuido al olvido de la línea de comandos. La metáfora del escritorio facilita el uso del ordenador, arrastrar un archivo desde una carpeta de un disco al escritorio lo que en verdad ejecuta una serie de ordenes: copia el archivo al escritorio, borra el archivo de la carpeta original, una secuencia de ordenes pueden irse descomponiendo hasta un lenguaje de ceros y unos, una serie de procesos que para la mayoría de usuarios se convierte en una cuestión de fe (Lichowski 2017).

Ejemplo de metáfora de escritorio es que los directorios se denominan Carpetas, estos son representados con un icono de una carpeta en color amarillo a partir de las carpetas tradicionales de oficina.

##### **1.1.2 Navegador de Archivos**

Un navegador de archivos es cualquier aplicación encargada de la manipulación de ficheros en dispositivos de almacenamiento. Estos programas hacen más fácil el manejo de los archivos en discos

duros, discos flexibles, discos ópticos, entre otros. Las funciones básicas que permite un administrador de archivos son: copiar, pegar, mover, crear directorios, navegar, borrar, renombrar y buscar. Opciones avanzadas pueden incluir distintos tipos de vistas, ordenación de archivos, previsualización de archivos multimedia, compartir archivos en una red, acceso a archivos remotos, (*File Transfer Protocol*, FTP), impresión, modificación de atributos de archivos, modificación de permisos de archivos, etcétera (Alegsa 2016a).

### 1.1.3 Carpetas o Directorios

Un directorio es una agrupación de archivos de datos, atendiendo a su contenido, a su propósito o a cualquier criterio que decida el usuario. Sirven para organizar mejor los archivos en un medio de almacenamiento como un disco duro, un dispositivo de almacenamiento USB (por sus siglas en inglés Universal Serial Bus), un CD (por sus siglas en inglés Compact Disc), entre otros. Dentro de un directorio pueden existir también otros directorios, llamados subdirectorios o subcarpetas; de hecho, todos los directorios son subdirectorios del directorio raíz de la unidad lógica (Alegsa 2016b).

Los directorios o carpetas permiten al usuario almacenar todo tipo de información dentro de un ordenador, siendo clasificada directamente por el sistema operativo o diciéndole el propio usuario al sistema dónde y cómo se clasifique esa información en el ordenador, para así de este tener todo organizado a su gusto (tecnologicon 2015).

### 1.1.4 Archivos

En informática se conoce como archivo o fichero a un conjunto organizado de unidades de información (bits) almacenados en un dispositivo. Se les denomina de esa manera como metáfora de a partir de los archivos tradicionales de oficina, escritos en papel, y que vendrían a ser su equivalente digital. Cada archivo posee una identificación única o nombre, la cual puede ser modificada o asignada a voluntad del usuario o del programador, y una extensión que determina qué tipo de archivo es y qué funciones cumple (Raffino 2019). Un archivo tiene generalmente los siguientes atributos: nombre, tipo, ubicación, tamaño, protección, hora, fecha. Estos se pueden estructurar de varias maneras las más comunes son:

**Secuencia de bytes:** El archivo es una serie no estructurada de bytes, posee máxima flexibilidad y el sistema operativo no sabe que contiene el archivo (Osnaya 2012).

**Secuencia de registros:** El archivo es una secuencia de registros de longitud fija, cada uno con su propia estructura interna (Osnaya 2012).

**Árbol:** El archivo consta de un árbol de registros, no necesariamente de la misma longitud, cada registro tiene un campo llamado *key* (llave o clave) en una posición fija del registro. El árbol se ordena mediante el campo clave para permitir una rápida búsqueda de una clave particular (Osnaya 2012).

## 1.2. Sistema operativo Android

El sistema operativo Android es una plataforma de código abierto que admite la construcción de aplicaciones para, de manera sencilla, manejar, configurar y personalizar el dispositivo móvil; esto hace que se pueda disfrutar de una interfaz amigable en la que se puede priorizar necesidades como correo, chat, redes sociales, multimedia y mensajería (Vanegas 2012).

Fue desarrollado por Android Inc., empresa que en 2005 fue comprada por Google, aunque no fue hasta 2008 cuando se popularizó, gracias a la unión al proyecto de Open Handset Alliance, un consorcio formado por 48 empresas de desarrollo de hardware, software y telecomunicaciones, que decidieron promocionar el software libre. Pero ha sido Google quien ha publicado la mayor parte del código fuente del sistema operativo, gracias al software Apache, que es una fundación que da soporte a proyectos de software de código abierto (Báez 2012).

Desde abril de 2009, las versiones de Android han sido desarrolladas bajo un nombre en clave y siguen un orden alfabético: Apple Pie, Banana Bread, Cupcake, Donut, Éclair, Froyo, Gingerbread, Honeycomb, Ice CreamSandwich, JellyBean, KitKat, Lollipop, Marshmallow, Nougat, Oreo, Pie y finalmente Android 10 en su última versión ya que Google les dice adiós a los postres. Para dar solución a las mejoras que los dispositivos requerían, tanto de hardware (pantallas más grandes, desaparecían los teclados), como de software (nuevas funcionalidades y requerimientos de los dispositivos).

Cada nueva versión de Android introduce mejoras y nuevas funcionalidades. También se corrigen bugs<sup>1</sup> y errores destacados. Se identifican por una numeración. A los dos primeros dígitos de esta se le da el nombre de un postre popular en inglés. A cada versión generalmente se le agregan modificaciones, surgiendo pequeñas actualizaciones menores (Versiones de Android 2018).

Tabla 1. Versiones Android (Versiones de Android 2018).

Versión / Nombre	Fecha	Características
Android 10.0	3 de Septiembre 2019	Novedades que incluye android 10 son: - Modo oscuro en todo el sistema. - Soporte para operar con banda de redes 5G.

		<ul style="list-style-type: none"> <li>- Introducción de Focus Mode<sup>1</sup>.</li> <li>- Introducción de Live Caption<sup>2</sup>.</li> <li>- Introducción de una nueva función de llamada Burbujas (Bubbles).</li> <li>- Control por intensidad de presión en pantalla.</li> <li>- Compartir la contraseña de una red wifi a la que esté conectado mediante el código QR.</li> <li>- Selección de colores en el tema.</li> <li>- Mayor velocidad y eficiencia en Google Assistant (Asistente de Google).</li> <li>- Modo escritorio.</li> <li>- Soporte para Wifi 6 y WPA3 (Acceso Wi-Fi protegido).</li> <li>- Mejora la privacidad usuario.</li> <li>- Panel de configuración flotante.</li> <li>- Soporte para TLS 1.3 (Seguridad de la capa de transporte).</li> <li>- Compatibilidad con el códec de video AV1, el formato de video HDR10+ y el códec de audio Opus.</li> <li>- Nuevos gestos de la barra de navegación.</li> <li>- Mejora la seguridad del dispositivo.</li> </ul>
Android 9.0 Pie	6 de Agosto 2018	<p>Conocido antes de su lanzamiento oficial como Android P, la versión nombrada como Pie (tarta), posee las siguientes funcionalidades:</p> <ul style="list-style-type: none"> <li>- Nueva interfaz de usuario Material Design 2<sup>3</sup>.</li> <li>- Nuevo sistema de navegación por deslizamiento.</li> <li>- Nuevo Panel de notificaciones.</li> </ul>

<sup>1</sup> **Focus Mode:** herramienta de Google que permite desactivar temporalmente las aplicaciones que más distraen.

<sup>2</sup> **Live Caption** es una herramienta que permite agregar subtítulos automáticamente a videos, *podcasts* y mensajes de audio.

<sup>3</sup> **Material Design** es un sistema de diseño de código abierto que ayuda a los equipos a construir fácilmente experiencias digitales de alta calidad

		<ul style="list-style-type: none"> <li>- Soporte para teléfonos con muescas en pantalla.</li> <li>- Función de Batería Adaptativa, que prioriza el consumo de batería para las aplicaciones más utilizadas.</li> <li>- Función de Brillo Adaptativo, que adapta el brillo de pantalla en base a preferencias y entornos.</li> <li>- Función App Actions <sup>4</sup> , que añade sugerencias contextuales.</li> <li>- Soporte para HDR VP9 (Alto Rango Dinámico).</li> <li>- Soporte para codificación HEIF (formato de archivo de imagen de alta eficiencia).</li> </ul>
Android 8.0 Oreo	21 de Agosto 2017	<p>Conocido antes de su lanzamiento oficial como Android O, Oreo proporciona las siguientes funcionalidades:</p> <ul style="list-style-type: none"> <li>- Mejor gestión de notificaciones.</li> <li>- Fluid Experience (Experiencias fluidas permite que Android sea más rápido y tenga mejor gestión de la batería).</li> <li>- Iconos adaptativos.</li> <li>- Modo Picture in Picture<sup>5</sup> (una ventana flotante de vídeo encima de cualquier aplicación, pudiendo interactuar con ambas).</li> <li>- Project Treble<sup>6</sup> (actualizaciones más sencillas y seguras).</li> <li>- Selección de texto inteligente.</li> <li>- Nuevos emojis.</li> <li>- Autorrelleno de texto nativo.</li> <li>- Desaparece "Orígenes desconocidos" (es necesario autorizar manualmente aplicaciones para instalar archivos</li> </ul>

<sup>4</sup> **App Actions:** aplica la inteligencia artificial a las acciones que realizas dentro de los *apps* en tu teléfono o tableta

<sup>5</sup> **Picture in Picture (PiP)** es una característica de algunos televisores o dispositivos similares. Un canal de televisión es mostrado a pantalla completa mientras que otro es mostrado en una ventana menor. El sonido escuchado generalmente corresponde al programa principal.

<sup>6</sup> **Project Treble** es una nueva arquitectura para el sistema operativo que presumiblemente debería reducir la fragmentación al hacer las actualizaciones del sistema más rápidas y sencillas para los fabricantes.

		APK (Android Application Package, significado en español: Paquete de Aplicación Android)).
Android 7.0 Nougat (Turrón)	Agosto 2016	Soporte para ventanas múltiples, mejoras del centro de notificaciones, un consumo más eficiente de batería, actualizaciones más rápidas del sistema y de las aplicaciones e inicio del sistema más rápido. Liberada para los Nexus 6, 5x, 6P, 9, Nexus Player, Pixel C y Android One. La actualización menor 7.1 llegó en diciembre.
Android 6.0 Marshmallow (Malvavisco)	Octubre 2015	Versión liberada previamente en mayo bajo el nombre de Android M para los teléfonos Nexus 5 y Nexus 6 y la tableta Nexus 9. Introduce muchas funcionalidades que hacen de Android un sistema sólido.
Android 5.0 Lollipop (Piruleta)	Noviembre 2014	Introduce grandes cambios en la interface de usuario con el uso de "material design", un nuevo diseño con iconos planos (flat) y más sencillos, creados especialmente para la web. Además más intuitiva respuesta del contenido al tocar la pantalla, transiciones, transparencias especiales, nueva tipografía, imágenes de esquina a esquina y colores vivos. Las notificaciones se muestran en la pantalla de bloqueo. Un nuevo Gmail y nueva aplicación de mensajes SMS/MMS (por las siglas del inglés Short Message Service/ Multimedia Message Service). Mejoras en rendimiento, consumo de la batería y más opciones de configuración. Lollipop tuvo cuatro actualizaciones.
Android 4.4 KitKat	Noviembre 2013	Fue lanzado con el teléfono Nexus 5 de Google y LG. Tuvo 4 actualizaciones.
Android 4.3 Jelly Bean (Michel)	Julio de 2013	Fue lanzada la segunda generación del Nexus 7 con conectividad 4G LTE (Evolución a Largo Plazo, por sus siglas en inglés Long Term Evolution).

Android 4.2 Jelly Bean (Gummy Bear)	Noviembre 2012	Lo incluyeron el teléfono Nexus 4 desarrollado por Google en colaboración con LG y la tableta Nexus 10 con Samsung.
Android 4.1 Jelly Bean (Gomita Confitada o Gominola)	Julio de 2012	La tableta Nexus 7 fue el primer dispositivo en correr Jelly Bean.
Android 4.0 Ice Cream Sandwich (Sándwich de helado)	Octubre de 2011	Fue liberada con el Samsung Galaxy Nexus. Fue un sistema operativo para teléfonos inteligentes y tabletas. Tuvo tres actualizaciones posteriores.
Android 3.0 / 3.1 / 3.2 Honeycomb (Panal de miel)	Febrero de 2011	Primera actualización exclusiva para tabletas.
Android 2.3 Gingerbread (Pan de jengibre)	Diciembre 2010	Incluyó posteriormente cinco actualizaciones.
Android 2.2 Froyo (Yogur helado)	Mayo 2010	Incluyó posteriormente tres actualizaciones.
Android 2.0 Eclair	Octubre 2009	Incluyó posteriormente dos actualizaciones. En enero del 2010 fue lanzado el Nexus One usando esta versión. Fue el primero de una serie de teléfonos de Google que en lo adelante serían la nave insignia de Android.
Android 1.6 Donut	Septiembre 2009	Fue lanzado el SDK (acrónimo de "Software Development Kit" (Kit de desarrollo de software)) de Android, que permitió a los desarrolladores la creación de aplicaciones.
Android 1.5 Cupcake	Abril 2009	Primera versión con nombre de pastelería, que es una actualización de la anterior.
Android 1	Septiembre 2008	Primera versión comercial lanzada con el primer dispositivo Android, el HTC Dream. Posteriormente fue lanzada en febrero del 2009 la actualización 1.1, que corrigió algunos fallos y errores.

### 1.2.1 Características de Android:

Las principales características del sistema operativo Android, según (Castillo 2019) son:

- Código abierto.
- Núcleo basado en el Kernel de Linux.
- Adaptable a muchas pantallas y resoluciones.
- Utiliza SQLite para el almacenamiento de datos.
- Ofrece diferentes formas de mensajería.
- Soporte de Java y muchos formatos multimedia.
- Soporte de HTML, HTML5, Adobe Flash Player.
- Catálogo de aplicaciones gratuitas o pagas en el que pueden ser descargadas e instaladas.
- Bluetooth.
- Wifi.

### **1.2.2 Ventajas y Desventajas de Android:**

A continuación se mencionan las principales ventajas y desventajas del sistema operativo Android según (Tomás 2013). Entre las ventajas más notables se encuentran:

- Da completa libertad al propietario de un terminal a instalar cualquier aplicación en el formato APK. Google liberó la mayoría de su código bajo la licencia apache, una licencia libre y de código abierto. Cuenta con una de las comunidades más grande de desarrolladores.
- Dos tercios de sus aplicaciones son gratuitas y disponibles en su tienda oficial de aplicaciones: Google Play.
- Es completamente personalizable.
- Posee sistema de multitarea inteligente.

Entre sus desventajas se pueden mencionar:

- El acceso a las actualizaciones de cada versión, depende del fabricante.
- La mayoría de software maliciosos creados para Linux están destinados a dispositivos con este sistema operativo.
- A partir de la versión 8.0 Oreo la API para HotSpot dejó de ser pública.
- No puede ser usado out of the box (listo para usarse) en una computadora sin afectar la usabilidad.

### 1.3. Estudio de soluciones similares

#### 1.3.1 ES Explorador de Archivos

Uno de los exploradores para Android más completos es el ofrecido por la compañía china EStrong Inc. desde 2013. ES Explorador de Archivos está disponible en más de 30 idiomas diferentes. Según el fabricante, actualmente lo utilizan más de 300 millones de usuarios en todo el mundo para gestionar sus propios documentos y contenidos multimedia, así como para las aplicaciones instaladas en dispositivos Android. La potente aplicación de gestión de archivos, que requiere unos 26 MB de espacio en disco, es gratuita en su versión básica, pero la empresa china ofrece una versión de pago que contiene temas adicionales para la interfaz de usuario de la aplicación. ES Explorador de Archivos es considerado un excelente gestor de archivos para Android, sobre todo, por las numerosas funciones que ofrece para la organización de archivos. Por ejemplo, hay tres tipos de visualizaciones diferentes (iconos, lista o detalles) y para cada uno de ellos se pueden seleccionar tres tamaños (pequeño, mediano y grande) y varias opciones de clasificación (por nombre, tipo, tamaño o fecha de modificación). También permite cortar, renombrar, comprimir o incluso encriptar archivos. Su atractivo diseño y la posibilidad de abrir varias pestañas para diferentes directorios permite a los usuarios mantener una visión general sin problema (IONOS 2019).

#### Ventajas

- Ofrece soluciones automatizadas para la optimización del espacio del disco.
- Excelentes capacidades de transferencia de archivos a otros dispositivos.

#### Desventajas

- No se puede desactivar el cuadro de diálogo eliminar.
- La publicidad intrusiva solo puede desactivarse parcialmente.

#### 1.3.2 ASTRO

El gestor de Archivos ASTRO (también conocido como ASTRO Filemanager) es el explorador de archivos para Android de la compañía americana de software Metago Inc. La aplicación, lanzada en 2008, fue una de las primeras soluciones de gestión de archivos para el sistema operativo y cuenta con un extenso historial de desarrollo en el que el diseño y la funcionalidad han sido revisados y optimizados en profundidad. En la tienda Google Play Store, el galardonado programa, que ahora permite la gestión de aplicaciones instaladas como ES Explorador de Archivos, ya ha registrado más de 86 millones de descargas. ASTRO File Manager es gratuito y no tiene publicidad. ASTRO se centra en funciones

básicas que los usuarios necesitan para organizar sus archivos. Los botones como el símbolo de tres puntos o la lupa para llevar a cabo una búsqueda disponen de funciones adicionales. Por ejemplo, durante la búsqueda puedes especificar también la ubicación o el tamaño y la fecha de modificación de los archivos. También es posible renombrar y comprimir archivos y carpetas o crear otras nuevas. En lugar de la vista de cuadrícula predeterminada, los usuarios pueden cambiar a la vista de lista, grupo o imagen, así como ordenar los archivos por nombre, fecha, tamaño y tipo (IONOS 2019).

### **Ventajas**

- Permite búsquedas detalladas de archivos y carpetas.
- Interfaz clara e intuitiva.

### **Desventajas**

- No tiene función para cortar.
- No cuenta con modo de dos paneles.

### **1.3.3 FX File Explorer**

En 2012, la empresa NextApp Inc. con sede en EE.UU, sacó al mercado su propio gestor de archivos para Android conocido como FX File Explorer. La aplicación, que también se puede descargar desde GooglePlay Store, no solo ofrece excelentes funciones de gestión de archivos y aplicaciones, sino también un diseño que proporciona al usuario una verdadera sensación de pantalla de inicio. Además, se incluyen varias herramientas adicionales como un editor de texto, un reproductor multimedia o un visor de imágenes. La edición estándar es gratuita y tiene publicidad pero se puede actualizar a la versión plus por una pequeña cantidad de dinero. Esta versión de pago permite habilitar funciones para la transferencia de archivos desde y hacia otras plataformas. FX File Explorer permite crear nuevos documentos y carpetas y adaptar los nombres de los elementos existentes. También es posible cortar, archivar y comprimir archivos (.zip,.gzip,.tar). Si se desea mover archivos de una ubicación a otra, el modo de dos paneles opcional facilita el seguimiento de los mismos. Dependiendo del contenido de la carpeta, se pueden seleccionar diferentes formatos de visualización como cuadrícula, símbolo, lista o vista previa de imágenes y diversos modos de ordenación como por orden alfabético o por fecha. Mediante la función de búsqueda integrada, los usuarios pueden iniciar búsquedas individuales de texto completo basadas en criterios libremente seleccionables como la ubicación de almacenamiento, el tipo de archivo, la fecha o el tamaño (IONOS 2019).

### **Ventajas**

- La posibilidad de tener multiventanas.

- Muy buen diseño.

### **Desventajas**

- Enlaces a otras plataformas/servicios en línea disponible para la versión Plus Edition.

### **1.3.4 Amaze File Manager**

Amaze File Manager es una herramienta de gestión de archivos de código abierto que permite explorar todos los directorios de un terminal Android, mover archivos y carpetas, renombrar documentos, copiar y pegar archivos, etcétera. Una de sus características más llamativas de este navegador de archivo es que ofrece una interfaz limpia, sencilla y elegante, de hecho, los usuarios podrán elegir entre diferentes patrones de color para la aplicación dentro de las opciones. Además de las características básicas que se le asumen a cualquier gestor de archivos (cortar, copiar, eliminar, comprimir, extraer...), Amaze File Manager cuenta con algunas prestaciones destinadas a que trabajar con nuestros archivos sea una tarea más cómoda. El cajón de navegación rápida, por citar un ejemplo, permite tener una especie de accesos directos a los directorios que más usemos, crear copias de seguridad o revisar las propiedades de las aplicaciones instaladas. Otra prestación interesante es el gestor de aplicaciones, que nos permitirá echar un vistazo a todas las aplicaciones que tenemos instaladas, y borrarlas si ya no las queremos en la memoria de nuestro terminal Android. Amaze File Manager es un sencillo pero potente gestor de archivos, que pese a estar en una fase temprana de desarrollo, ofrece unas prestaciones y una interfaz más que dignas (Uptodown Technologies 2018a).

### **Ventajas**

- Las entradas de carpetas y archivos se pueden personalizar visualmente (fuente y tamaño, color de fondo, entre otras).

### **Desventajas**

- No cuenta con Modo dos paneles (vertical y/o horizontal).

### **1.3.5 Simple File Manager**

Simple File Manager es un navegador de archivo de código abierto a través del cual podrás gestionar todos los ficheros almacenados en un dispositivo. Mediante una interfaz sencilla, esta herramienta forma parte del paquete desarrollado por Simple Mobile Tools. Simple File Manager permite crear carpetas y archivos a gusto del usuario con el objetivo de crear una estructura de mucho más organizada. Además, la herramienta permite renombrar, copiar, mover o eliminar cualquier registro almacenado tanto en la memoria interna como en la tarjeta SD (por sus siglas en inglés Secure Digital).

Simple File Manager tiene como objetivo gestionar de forma intuitiva toda la información almacenada en un dispositivo (Uptodown Technologies 2018b).

### **Ventajas**

- Interfaz de usuario muy bien estructurada.
- Cuenta con una función “Seleccionar todo”.

### **Desventajas**

- No dispone de modo multiventana.
- Función de búsquedas sin opciones de filtro.

### **1.3.6 MK Explorer**

MK Explorer es un navegador de archivos de código abierto que no destaca por ser muy avanzado, sino que está pensado para realizar las tareas más comunes de un usuario de Android intermedio. Aun así, incorpora características avanzadas como *root access*, animaciones fluidas y otras. Permite gestionar todo el almacenamiento del dispositivo en subcarpetas que él mismo de organiza, como descargas, fotografías o vídeos. MK Explorer genera una primera división que es de memoria interna y externa, aunque permite decidir que muestre ficheros en la raíz en la opción de ajustes. La herramienta incorpora funciones para comprimir o descomprimir archivos ZIP o RAR, copiar, navegar, mover, borrar o buscar archivos y generar atajos en el lanzador de ciertas rutas para acceder rápidamente a ellas desde la pantalla principal. Integra un editor de texto básico, con atajos de teclado y hasta su propia galería. Es una herramienta ligera guiado para realizar tareas básicas de un usuario común (Andro4all 2019).

### **Ventajas**

- Cuenta con función para doble venta.
- Incorpora la posibilidad de personalizar íconos y colores de la aplicación.

### **Desventajas**

- Es muy simple y no incorpora funciones complejas.
- La publicidad intrusiva solo puede desactivarse parcialmente.

### **1.3.7 Solid Explorer File Manager**

Solid Explorer File Manager es un navegador de archivos de código abierto que permite explorar todos los directorios de un terminal Android, su interfaz por completa es basada en material designer. Integra modo dos paneles para la navegación entre archivos con la posibilidad de arrastrar y soltar archivos

entre paneles, permite personalizar íconos, colores y temas de la aplicación. Incorpora soporte para FTP (por sus siglas en inglés: File Transfer Protocol), SFT (System File Table), WebDAV (Web-based Distributed Authoring and Versioning) y SMB/CIFS (Samba/ Common Internet File System). Integra soporte para servicios en la nube. Solid Explorer File Manager es un navegador de archivos simple pero a la vez tiene funcionalidades esenciales para realizar tareas comunes en el almacenamiento interno de un dispositivo.

### Ventajas

- Cuenta con modo dos paneles.
- Conexión con la nube.

### Desventajas

- Los terminales con Android stock no lo incluyen.

Tabla 2. Análisis de las posibles soluciones

	Código Abierto	Operaciones con archivos	Conexión con la nube	Comprimir archivos	Navegación entre directorios
ES Explorador de archivos	no	sí	sí	sí	sí
Gestor de archivos ASTRO	no	sí	sí	sí	sí
FX File Explorer	no	sí	sí	sí	sí
Amaze File Manager	sí	sí	si	sí	sí
Simple File Manager	sí	sí	no	no	sí
MK Explorer	sí	sí	no	sí	sí
Solid Explorer File Manager	sí	sí	sí	no	sí

### **1.3.6 Conclusiones del estudio de las posibles soluciones**

Una vez concluido el estudio de las anteriores aplicaciones, se determina que las mismas no cumplen con las características necesarias para dar solución al problema planteado, sino que cumplen este parcialmente. Esto se debe a que todos los navegadores de archivos fueron diseñados para dispositivos móviles con pantallas pequeñas e interactuar con estos a partir de gestos y pantallas táctiles. Se extraen de las aplicaciones estudiadas funcionalidades claves (mover, copiar, cortar, eliminar, entre otras) para el desarrollo de la solución propuesta. Finalmente se decide la reutilización de Amaze File Manager, debido a que es la más completa y la licencia no es privativa por lo que se puede modificar o ajustar su código a las necesidades del proyecto en cuestión.

## **1.4 . Selección del lenguaje, tecnologías y herramientas**

La selección del lenguaje, tecnologías y herramientas constituye un factor clave en el desarrollo de una solución informática. Esta elección está guiada principalmente por los requerimientos que presenta el software a desarrollar. En la presente sección se definen las tecnologías y herramientas que se escogieron para la implementación.

### **1.4.1 Lenguaje de programación**

Un lenguaje de programación es una interfaz de usuario con la cual los seres humanos pueden elaborar procesos que serán ejecutados por una máquina de cómputo. Los lenguajes de programación tienen reglas y parámetros establecidos para poder realizar diferentes acciones, existen diferentes lenguajes de programación y dependen del tipo de programación (Bellas, Unanue y Fernández 2016).

Existen varios lenguajes de programación que pueden ser utilizados para desarrollar aplicaciones Android, entre estos se encuentran: Java, Kotlin y C++. Para el desarrollo de la aplicación será utilizado Java, por ser el lenguaje nativo que usa Android.

#### **Java v1.8.0**

Lenguaje de programación de propósito general, concurrente y orientado a objeto. Permite que el programa se escriba una sola vez y se ejecute en diferentes dispositivos, es decir el código que se ejecuta en una plataforma no tiene que volverse a compilar para ejecutarse en otra diferente. Su sintaxis se deriva principalmente de los lenguajes C y C++ mejorando principalmente la facilidad de uso y con la diferencia de ser orientado a objetos por completo (Flanagan 2005).

Java es un lenguaje orientado a objetos, eso implica que su concepción es muy próxima a la forma de pensar humana. También posee otras características importantes según (Exes 2015):

- Es un lenguaje compilado, que genera ficheros de clases, las que son en realidad interpretadas por la máquina virtual de Java, siendo esta la que mantiene el control sobre las clases que se estén ejecutando.
- Es un lenguaje multiplataforma: el mismo código escrito en Java que funciona en un SO, funcionará en cualquier otro que tenga instalada la máquina virtual de Java.
- Es un lenguaje seguro: la máquina virtual, al ejecutar el código escrito en Java, realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras, como por ejemplo los punteros.
- Gracias al API (Interfaz de Programación de Aplicaciones) de Java, se puede ampliar el lenguaje para que sea capaz de comunicarse con equipos mediante red, acceder a bases de datos, crear páginas HTML dinámicas y crear aplicaciones visuales.

Se decide la utilización de Java teniendo en cuenta principalmente que posee un poderoso conjunto de herramientas de desarrollo de software (Android SDK), desarrolladas por Google que permite la implementación de soluciones de calidad para Android pudiendo ser ejecutadas en dispositivos con diferentes configuraciones de hardware.

### **1.4.2 Lenguaje de modelado**

#### **UML v2.0**

Lenguaje de Modelado Unificado (por sus siglas en inglés UML) es un lenguaje de modelado estandarizado que consiste en un integrado conjunto de diagramas; desarrollados para ayudar a desarrolladores de software y de sistema en la especificación, visualización, construcción u documentación de los artefactos de los sistemas de software, así como también para el modelado de negocio y otros sistemas (What is Unified Modeling Language (UML)? 2017).

Para modelar los artefactos generados se utilizará UML pues el equipo de desarrollo posee amplia experiencia en el uso de este lenguaje de modelado.

### **1.4.3 Herramientas**

Las herramientas de desarrollo de software son diversos productos informáticos que dan soporte a una tarea concreta dentro de las actividades de desarrollo de software, facilitando y asegurando entregar un sistema con calidad.

#### **Visual Paradigm v8.0**

Herramienta de software diseñada para modelar los sistemas de información empresarial y gestionar el proceso de desarrollo de software. Soporta lenguajes de modelado y estándares claves como UML, SysML (System Modeling Language), BPMN (Business Process Model and Notation), entre otros. Ofrece además un conjunto completo de herramientas de software para tareas como la captura de requisitos, análisis de procesos, diseño de sistemas y diseño de base de datos (Visual Paradigm Frequently Asked Questions 2017).

Se selecciona esta herramienta debido al conjunto de funcionalidades útiles que ofrece para llevar a cabo el proceso de desarrollo de software y a su adecuada integración con UML, el lenguaje de modelado seleccionado.

### **Android Studio v3.5**

Entorno de Desarrollo Integrado (por sus siglas en inglés IDE) oficial para el desarrollo de aplicaciones Android, desarrollado por Google publicado de forma gratuita bajo la licencia de código abierto Apache 2.0. Está basado en el IDE IntelliJ IDEA de la compañía JetBrains y está disponible para las plataformas Windows, MacOS y GNU/Linux (Android Studio Release Notes | Android Studio 2017).

La selección de este IDE se basa principalmente en la experiencia del equipo de desarrollo en su uso, y además en que es el recomendado por Google para el desarrollo de aplicaciones Android puesto que está implementado con tal propósito.

### **Gradle v5.4.1**

Herramienta de construcción enfocada en la construcción automática y el soporte para el desarrollo en diferentes lenguajes de programación. Proporciona un modelo flexible que puede soportar todo el ciclo de desarrollo, desde compilar y empaquetar hasta publicar sitios web. Soporta la construcción automática para diferentes lenguajes como Java, Scala, Android, C/C++ y Groovy; además se integra estrechamente con varios IDEs como Eclipse, IntelliJ y Android Studio. Además ofrece funcionalidades como la gestión de dependencias, facilitando su resolución a través de repositorios centrales publicados en Internet (gradle: Adaptable, fast automation for all 2019).

Esta selección está condicionada principalmente a que es utilizada por defecto en Android Studio para la construcción de proyectos y la gestión de dependencias ahorrando así tiempo de configuración.

## **1.5. Metodología de desarrollo**

En ingeniería de software una metodología de desarrollo constituye un conjunto de métodos, procedimientos, técnicas, herramientas y soportes documentales utilizados para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información (Pinzón, Guevara Bolaños 2006).

Para la adecuada implementación de la solución propuesta es necesario la selección de una metodología que guíe el ciclo de vida del proyecto para asegurar un producto de calidad. Se selecciona en consecuencia la metodología AUP-UCI teniendo en cuenta que es la metodología adaptada al ciclo de vida de los proyectos productivos de la universidad, es ampliamente usada en el área y es extremadamente flexible al proceso de desarrollo de software.

### 1.5.1 Metodología de desarrollo de software AUP-UCI

AUP-UCI constituye una variante de AUP (Proceso Unificado Ágil, por sus siglas en inglés) surge con el objetivo de ser una metodología que se adapte al ciclo de vida definido por la actividad productiva en la universidad. Se elaboró teniendo en cuenta el Modelo CMMI-DEV v1.3 que constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora, estas prácticas se centran en el desarrollo de productos y servicios de calidad (Ramírez 2018).

Esta metodología propone tres fases para el desarrollo del software (Inicio, Ejecución y Cierre), siete disciplinas (Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas internas, Pruebas de liberación, Pruebas de aceptación) y once roles (Jefe de proyecto, Planificador, Analista, Arquitecto de información, Desarrollador, Administrador de la configuración, Cliente o proveedor de requisitos, Administrador de calidad, Probador, Arquitecto de software y Administrador de bases de datos (Ramírez 2018).

AUP-UCI propone además cuatro escenarios a utilizar para modelar el sistema en los proyectos, estos se explican a continuación (Ramírez 2018):

- **Escenario No. 1:** modelar el sistema con CUS (Caso de Uso del Sistema) cuando el negocio se modele con CUN (Caso de Uso del Negocio) y MC (Modelo Conceptual).
- **Escenario No. 2:** modelar el sistema con CUS cuando el negocio se modele con MC.
- **Escenario No 3:** modelar el sistema con DRP (Descripción de Requisitos por Proceso) cuando el negocio se modele con DPN (Descripción de Proceso de Negocio) y con MC.
- **Escenario No 4:** modelar el sistema con HU (Historia de Usuario) cuando no se realice modelado de negocio.

De los escenarios descritos anteriormente se selecciona para el modelado del sistema de la solución a desarrollar el escenario número cuatro. Para esta elección se tuvo en cuenta principalmente que no se realiza modelado del negocio puesto que este no está bien definido, y que la solución a desarrollar no consta de un proyecto muy extenso lo que permite que las historias de usuario no posean demasiada información y sean lo más concretas posible.

## 1.6 Conclusiones Parciales

- Con el estudio de los principales conceptos asociados al dominio del problema se logra obtener una mejor interpretación del mismo.
- Mediante el estudio de los navegadores de archivos existentes se obtiene como resultado que los mismos no constituyen una solución para el problema planteado, por lo que se requiere del desarrollo de una nueva solución reutilizando una de las aplicaciones similares que cumplen parcialmente con el problema de la investigación. Se identificaron funcionalidades comunes en dichos sistemas que deberán ser incorporadas en la nueva solución, para aumentar su valor competitivo.
- El análisis realizado sobre las diferentes tecnologías, incluyendo la selección de la metodología de desarrollo AUP-UCI permite la creación del entorno de trabajo para el desarrollo de la solución.

## **CAPÍTULO 2. Análisis y Diseño del navegador de archivo para NovaDroid**

### **2.1. Introducción**

En el presente capítulo se exponen las principales características de la propuesta de solución a desarrollar. Como parte de la disciplina de Requisitos definida por la metodología de desarrollo AUP-UCI se realiza la captura de los requisitos funcionales y no funcionales, y se describen las historias de usuarios correspondientes a estos. Por último, como parte de la disciplina Análisis y Diseño se presentan la arquitectura de la solución, los patrones de diseño utilizados y el diagrama de clases del diseño.

### **2.2. Propuesta de solución**

Se propone como solución una aplicación para dispositivos con sistema operativo NovaDroid que permita navegar y administrar la estructura de carpetas y archivos, tanto la del sistema como las que ha creado el usuario para satisfacer sus necesidades.

### **2.3. Captura de requisitos**

Según el estándar 1233 de la IEEE (Instituto de Ingenieros Eléctricos y Electrónicos): Guía para el desarrollo de Especificaciones de Requerimientos de Sistemas, un requisito se define como:

- Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
- Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.

Es posible concluir que los requisitos de software son características y funcionalidades que debe poseer un sistema y están enfocados hacia lo que debe hacer el software. Además, pueden ser clasificados en funcionales y no funcionales.

#### **2.3.1 Técnicas para la captura de requisitos**

En el proceso de desarrollo de un sistema el equipo de desarrollo siempre se enfrenta al problema de la identificación de requisitos. La definición de estos es un proceso complejo, pues hay que identificar los requisitos que el sistema debe cumplir en orden de satisfacer las necesidades de los usuarios finales y clientes. Para realizar este proceso existen diferentes técnicas, su selección y resultados dependen en gran medida del equipo de desarrollo, como de los propios usuarios o clientes que participen en ellas. Se muestran a continuación las técnicas utilizadas para identificación de los requisitos:

#### **Análisis de sistemas existentes:**

Mediante el análisis de sistemas existentes es posible estudiar aplicaciones similares a la que se necesita obtener. Una vez que se tiene la concepción del funcionamiento de un software similar en cuanto a funcionalidades y características es más sencillo identificar los requisitos del sistema que se necesita implementar. Durante la investigación se realizó un estudio de aplicaciones similares a la solución a desarrollar, en las cuales se observaron los diseños de sus interfaces, las funcionalidades que ofrecen, el grado de dificultad a la hora de interactuar con la aplicación, entre otros rasgos importantes que contribuyeran a obtener un producto con la mejor calidad posible.

**Tormenta de ideas:**

Es una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador (Escalona 2002).

El equipo de desarrollo se reunió con el personal técnico de NovaDroid y a partir de un conjunto de ideas propuestas se identificaron importantes requisitos con los que debería cumplir la solución.

**2.3.2 Requisitos funcionales (RF)**

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, cómo debe comportarse en situaciones específicas. En algunos casos también pueden plantear explícitamente qué no debe hacer el sistema (Ian Somerville 2005).

A partir de lo antes planteado se definieron los siguientes requisitos funcionales:

Tabla 3: Descripción de los requisitos funcionales

No.	Requisito Funcional	Descripción	Complejidad	Prioridad para el cliente
1.	Crear carpeta.	La aplicación deberá permitir al usuario crear una carpeta la cual inicialmente se crea vacía y con el nombre de "Nueva Carpeta".	Alta	Alta
2.	Crear archivo.	La aplicación deberá permitir al usuario crear un archivo el cual puede ser un texto, un dibujo, un video etc. Estos se diferencian por el icono que los representa.	Alta	Alta

3.	Cambiar el nombre de una carpeta o archivo.	La aplicación deberá permitir al usuario renombrar una carpeta o un archivo.	Media	Baja
4.	Seleccionar carpetas o archivos.	La aplicación deberá permitir al usuario seleccionar una o varias carpetas, así como uno o varios archivos.	Media	Baja
5.	Mover carpetas o archivos.	La aplicación deberá permitir al usuario mover una o varias carpetas, así como uno o varios archivos, eliminando este de la ubicación original y solo guardándolo en la ubicación de destino.	Alta	Media
6.	Copiar carpetas o archivos.	La aplicación deberá permitir al usuario copiar una o varias carpetas, así como uno o varios archivos, creando dos copias del mismo o duplicando este, luego de este procedimiento ya sea una carpeta o un archivo se encontrara en la ubicación original y en la de destino.	Alta	Alta
7.	Eliminar carpetas o archivos.	La aplicación deberá permitir al usuario eliminar carpetas o archivos de forma permanente.	Alta	Alta
8.	Buscar carpeta o archivo.	La aplicación deberá permitir al usuario buscar cualquier archivo o carpeta. Contará con un buscador en la parte superior.	Alta	Media
9.	Comprimir carpetas o archivos.	La aplicación deberá permitir al usuario comprimir carpetas o archivos permitiendo así reducir el tamaño de la información.	Alta	Media
10.	Extraer carpetas o archivos.	La aplicación deberá permitir al usuario extraer carpetas o archivos.	Alta	Media
11.	Administrar aplicaciones.	La aplicación deberá permitir al usuario administrar aplicaciones para realizar operaciones sobre las mismas.	Alta	Alta

12.	Compartir carpetas o archivos.	La aplicación deberá permitir al usuario compartir carpetas y archivos. Mostrará una ventana auxiliar para definir la vía por la que este será compartido.	Alta	Media
13.	Ordenar carpetas y archivos.	La aplicación deberá permitir al usuario ordenar las carpetas o archivos de su ordenador ya sea por orden alfabético o por la fecha.	Media	Baja
14.	Mostrar propiedades de una carpeta o archivo.	La aplicación deberá permitir al usuario visualizar las propiedades de una carpeta o archivo, ya sea tamaño, localización, tipo de archivo, entre otras.	Alta	Media
15.	Cambiar vista.	La aplicación deberá permitir al usuario cambiar la forma de visualizar los archivos y carpetas.	Alta	Baja
16.	Añadir atajo.	La aplicación deberá permitir al usuario añadir un atajo de una carpeta o archivo en caso de querer acceder a ella luego de manera sencilla.	Alta	Baja
17.	Abrir carpetas o archivos.	La aplicación deberá permitir al usuario abrir una carpeta o un archivo, este último a partir del tipo que sea, abrirá la aplicación adecuada al tipo de fichero que se desea ejecutar.	Alta	Media
18.	Listar opciones del menú.	La aplicación deberá permitir al usuario hacer clic en el botón del menú y luego esta listara las opciones contenidas dentro del menú.	Alta	Media

### 2.3.3 Requisitos no funcionales (RNF)

Los requisitos no funcionales son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades de este como fiabilidad, tiempo de respuesta y la capacidad de almacenamiento. Incluyen además restricciones de tiempo, sobre el proceso de

desarrollo y estándares (SOMMERVILLE 2011). A continuación, se definen los requisitos no funcionales que debe cumplir la aplicación basándose en los establecido por las normas ISO 25000 Calidad del Producto de Software, específicamente la ISO/IEC 25010 que define las características de calidad que se tienen en cuenta al evaluar las propiedades de un producto de software:

### **Confiabilidad**

RNF1- La aplicación debe ser capaz de manejar los errores y recuperarse.

### **Hardware**

RNF2- Memoria RAM del dispositivo con capacidad mínima de 256 Mb.

RNF3- Almacenamiento interno con capacidad mínima de 100 Mb.

RNF4- Soporte para conexiones WIFI.

RNF5- Soporte para conexiones Bluetooth.

### **Interfaz**

RNF6- La aplicación deberá poseer interfaz de usuario basada en Material Design.

RNF7- La aplicación tendrá una interfaz de usuario sencilla e intuitiva.

### **Portabilidad**

RNF8- La aplicación deberá funcionar correctamente en todas las versiones del sistema operativo NovaDroid.

RNF9- La aplicación deberá soportar los idiomas inglés y español.

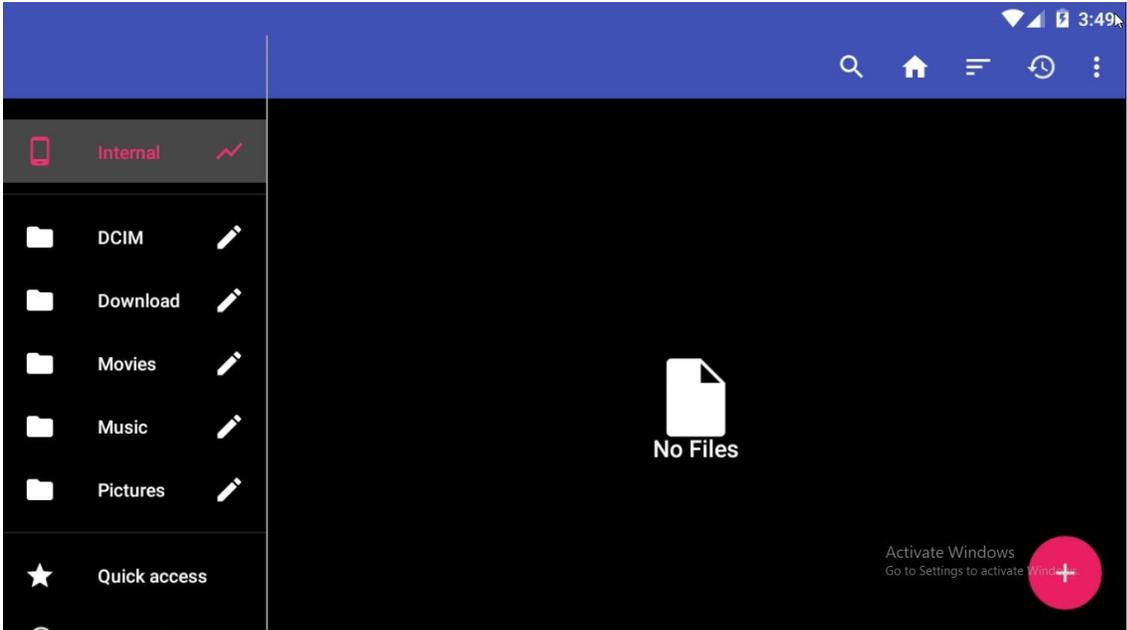
## **2.4. Historias de usuario**

Las historias de usuario (HU) constituyen una forma de administración de requisitos sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Las historias de usuario son cortas descripciones de una funcionalidad desde la perspectiva de la persona que la desea, usualmente un usuario o cliente. Las mismas son escritas utilizando el lenguaje común. Son empleadas en las metodologías de desarrollo ágiles para la especificación de requisitos (COHN 2018).

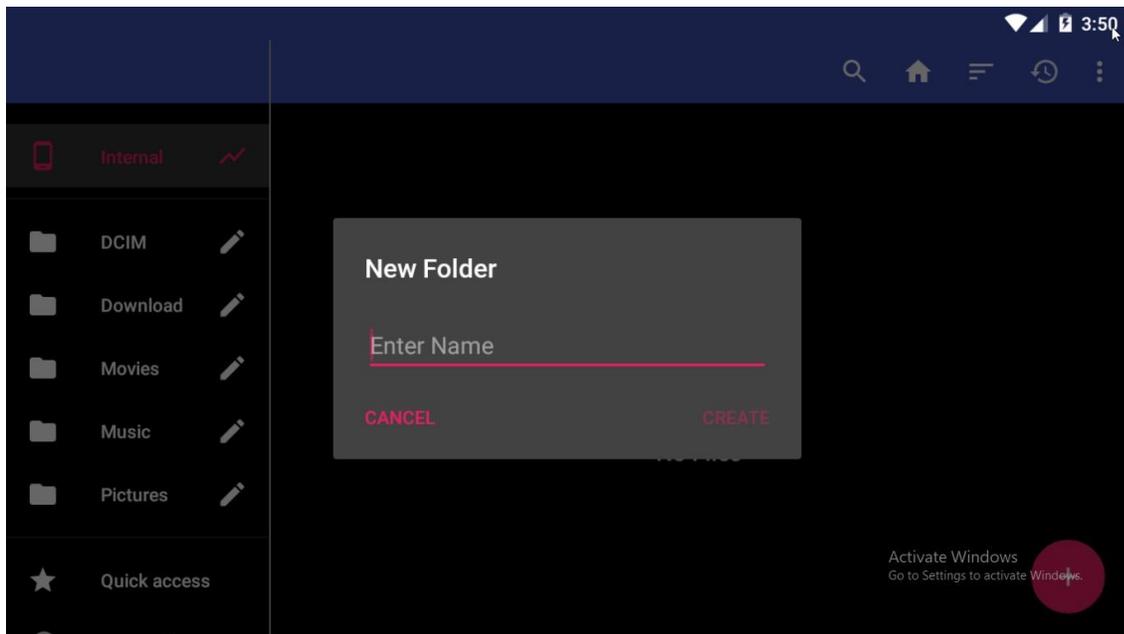
En correspondencia con la selección del escenario número cuatro de la metodología empleada se procede a modelar el sistema con historias de usuario, donde se define una por cada requisito funcional, lográndose un total de 18 HU. Se muestran a continuación las HU “Crear Carpeta”, “Copiar carpetas y

archivos” y “Eliminar carpetas o archivos” por ser las que describen las funcionalidades que poseen mayor prioridad para el cliente.

Tabla 4: Historia de usuario “Crear Carpeta”.

<b>Número:</b> HU – 1	<b>Nombre del requisito:</b> Crear Carpeta.
<b>Programador:</b> Alvaro Castro Pérez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 3
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 1
<p><b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario crear carpetas. Se listan a continuación las funcionalidades requeridas:</p> <ul style="list-style-type: none"> <li>• Permitir al usuario navegar hasta la ubicación donde desea crear la carpeta.</li> <li>• Mostrar el botón de menú de opciones.</li> <li>• Listar opciones del menú luego de hacer clic en el botón.</li> <li>• Permitir al usuario elegir en el menú la opción “Crear Carpeta”.</li> <li>• Permitir al usuario introducir el nombre de la carpeta.</li> <li>• Mostrar la opción cancelar y crear.</li> <li>• Listar la nueva carpeta.</li> </ul>	
<p><b>Prototipo de interfaz:</b></p>  <p>The screenshot shows a mobile application interface for file management. At the top, there is a blue header bar with navigation icons (search, home, list, refresh, and menu) and the time 3:49. Below the header, a dark sidebar on the left contains a list of folders: 'Internal' (with a red checkmark), 'DCIM', 'Download', 'Movies', 'Music', and 'Pictures'. Each folder has a white folder icon and a pencil icon for editing. At the bottom of the sidebar is a 'Quick access' section with a star icon. The main content area is dark and currently displays 'No Files' with a white document icon. In the bottom right corner, there is a red circular button with a white plus sign and the text 'Activate Windows Go to Settings to activate Windows'.</p>	

Luego de seleccionar la opción de crear carpeta.



Luego de introducir el nombre de la carpeta en la interfaz de diálogo con el usuario y listando esta en la ubicación deseada.

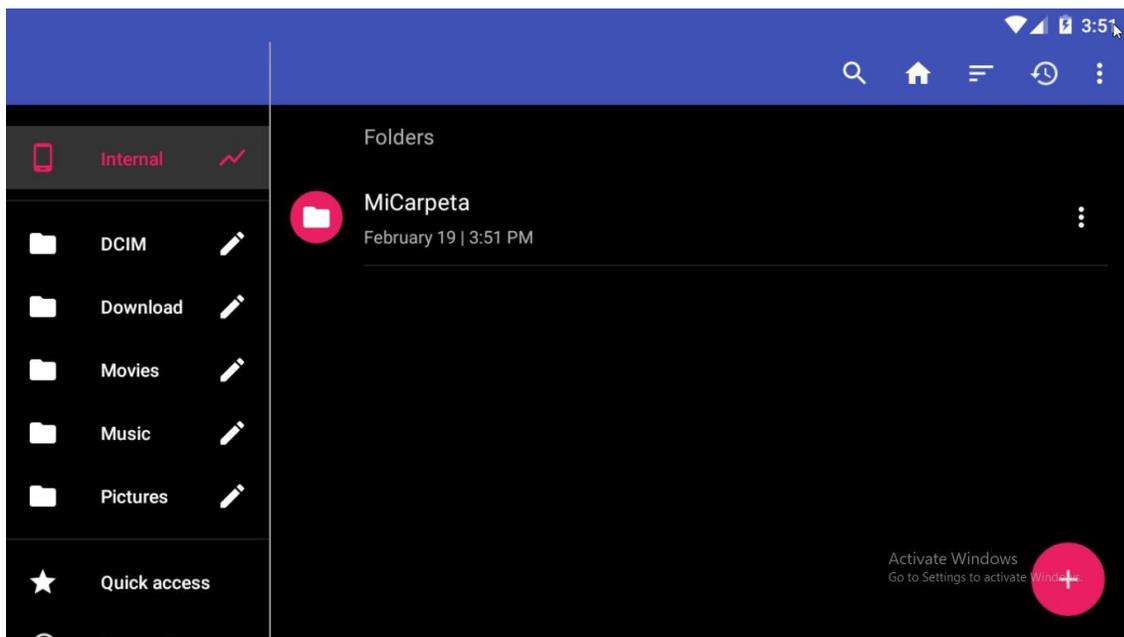
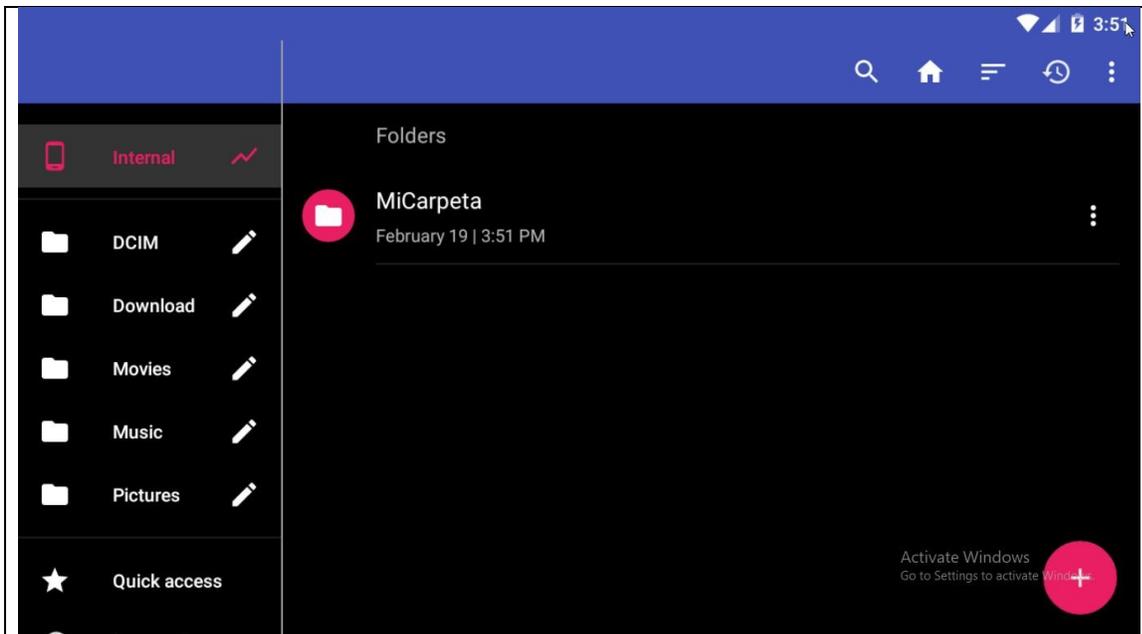
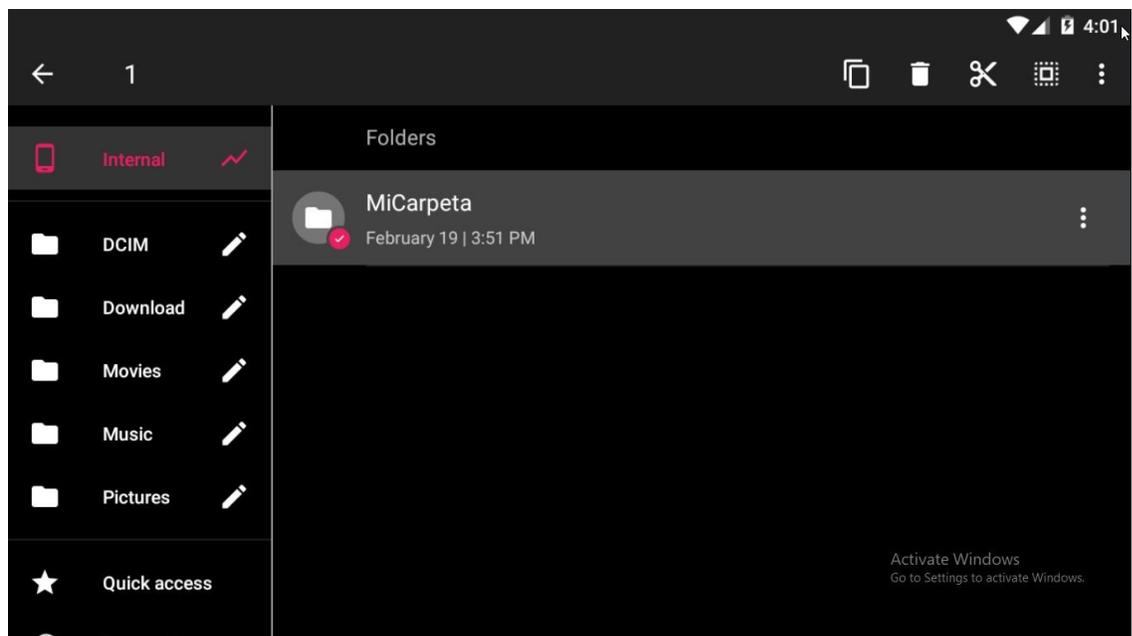


Tabla 5: Historia de usuario “Copiar carpetas o archivos”.

<b>Número:</b> HU – 6	<b>Nombre del requisito:</b> Copiar carpetas o archivos.
<b>Programador:</b> Alvaro Castro Pérez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 2
<p><b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario copiar carpetas o archivos desde una ubicación de origen a una ubicación de destino.</p> <p>Se listan a continuación las funcionalidades requeridas:</p> <ul style="list-style-type: none"> <li>• Permitir al usuario navegar hasta la ubicación de donde desea copiar la carpeta o el archivo.</li> <li>• Posibilitar al usuario seleccionar la carpeta o el archivo que desea copiar en otra ubicación.</li> <li>• Mostrar el botón de menú de opciones.</li> <li>• Listar opciones del menú luego de hacer clic en el botón.</li> <li>• Permitir al usuario elegir en el menú la opción “Copiar”.</li> <li>• Permitir al usuario navegar hasta la unidad de almacenamiento o la carpeta donde desea copiar el elemento seleccionado.</li> <li>• Permitir al usuario elegir en el menú la opción “Pegar”.</li> </ul>	
<b>Prototipo de interfaz:</b>	



Al seleccionar la carpeta que el usuario desea copiar.



Luego de seleccionar la opción de pegar y copiar la carpeta en su nueva ubicación.

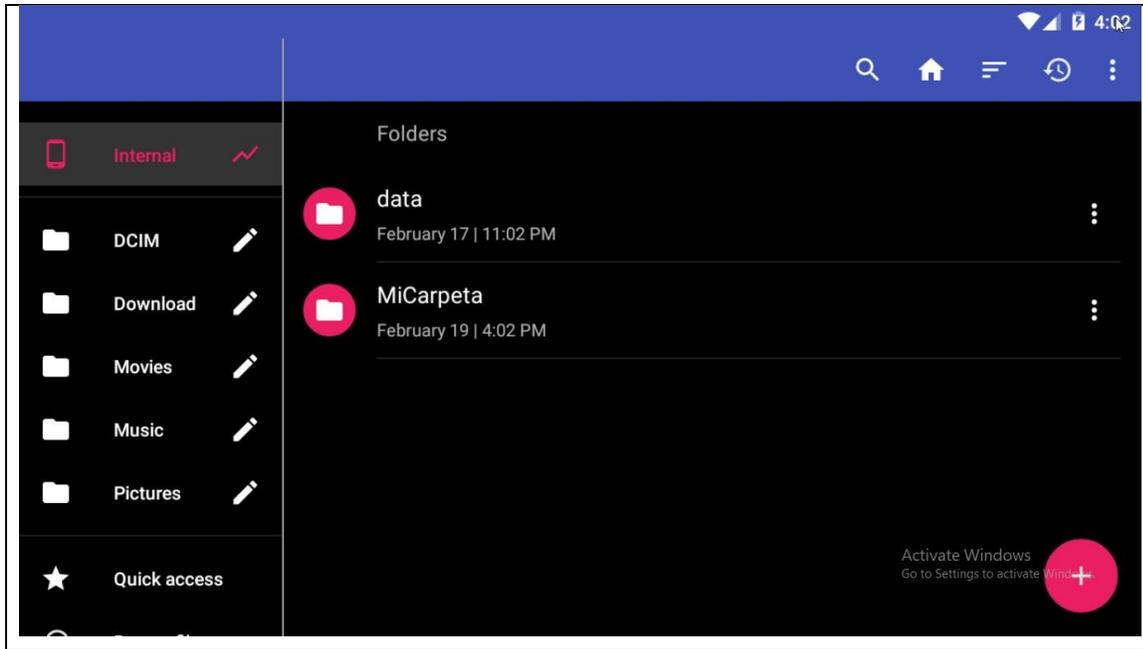
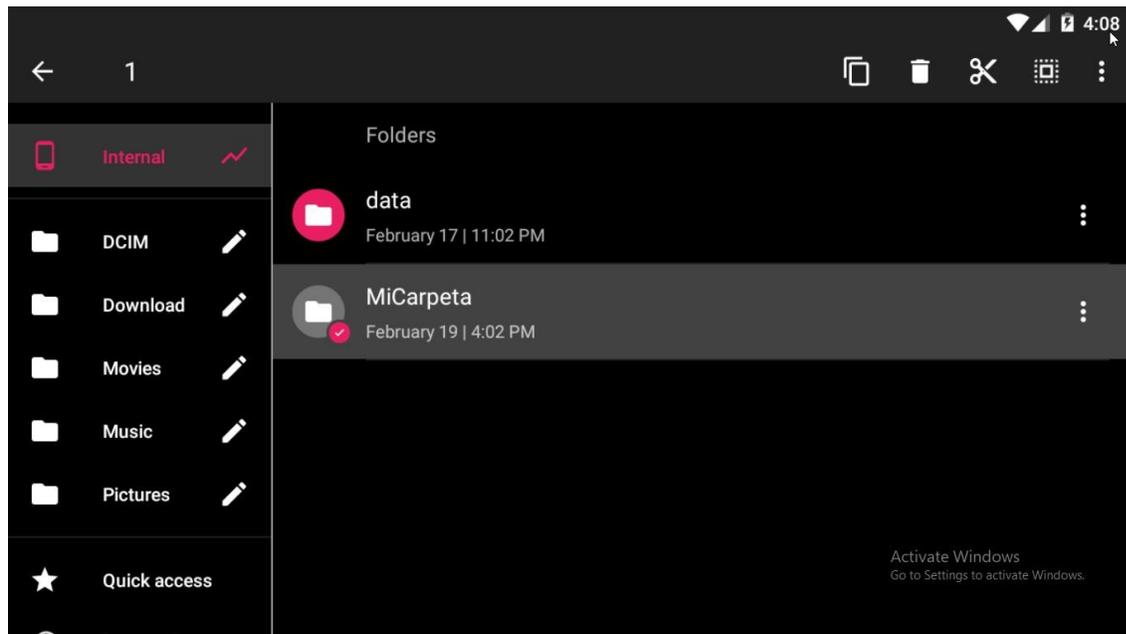


Tabla 6: Historia de usuario “Eliminar carpetas o archivos”.

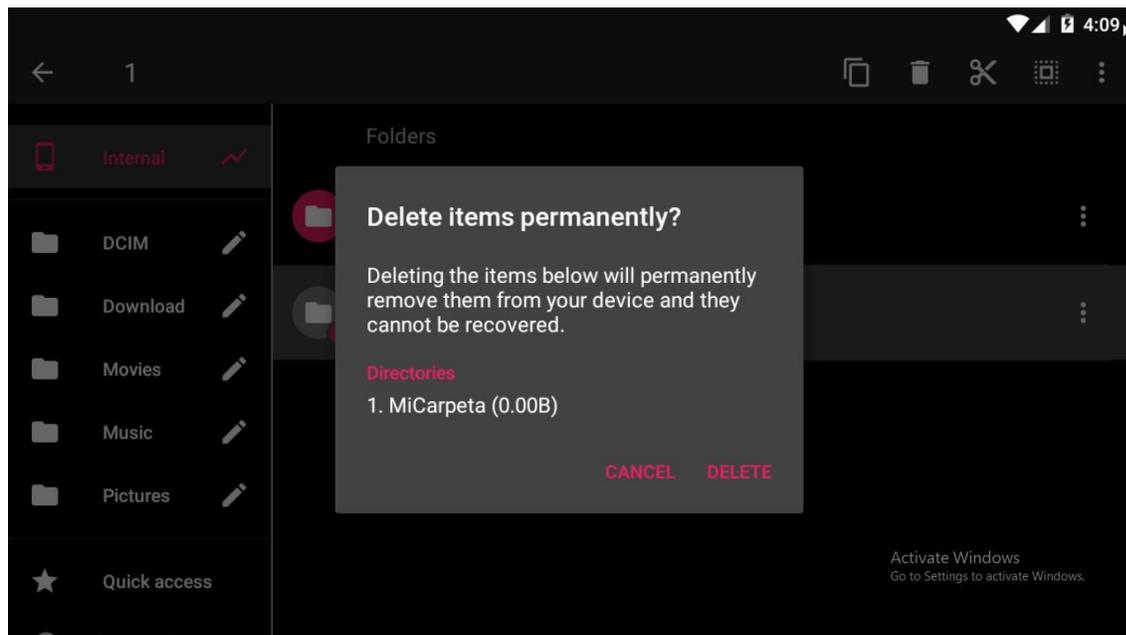
<b>Número:</b> HU – 7		<b>Nombre del requisito:</b> Eliminar carpetas o archivos.	
<b>Programador:</b> Alvaro Castro Pérez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 3	
<b>Riesgo en Desarrollo:</b> Medio		<b>Tiempo Real:</b> 2	
<p><b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario eliminar carpetas o archivos del dispositivo.</p> <p>Se listan a continuación las funcionalidades requeridas:</p> <ul style="list-style-type: none"> <li>• Permitir al usuario navegar hasta la ubicación de donde desea eliminar la carpeta o el archivo.</li> <li>• Posibilitar al usuario seleccionar la carpeta o el archivo que desea eliminar.</li> <li>• Mostrar el botón de menú de opciones.</li> <li>• Listar opciones del menú luego de hacer clic en el botón.</li> <li>• Permitir al usuario elegir en el menú la opción “Eliminar”.</li> <li>• Mostrar una caja de diálogo para confirmar la acción, esta confirmación significa una eliminación para los elementos del disco duro.</li> <li>• Traspasar los elementos a la papelera de reciclaje, donde podrás recuperarlos si</li> </ul>			

- posteriormente lo deseas.

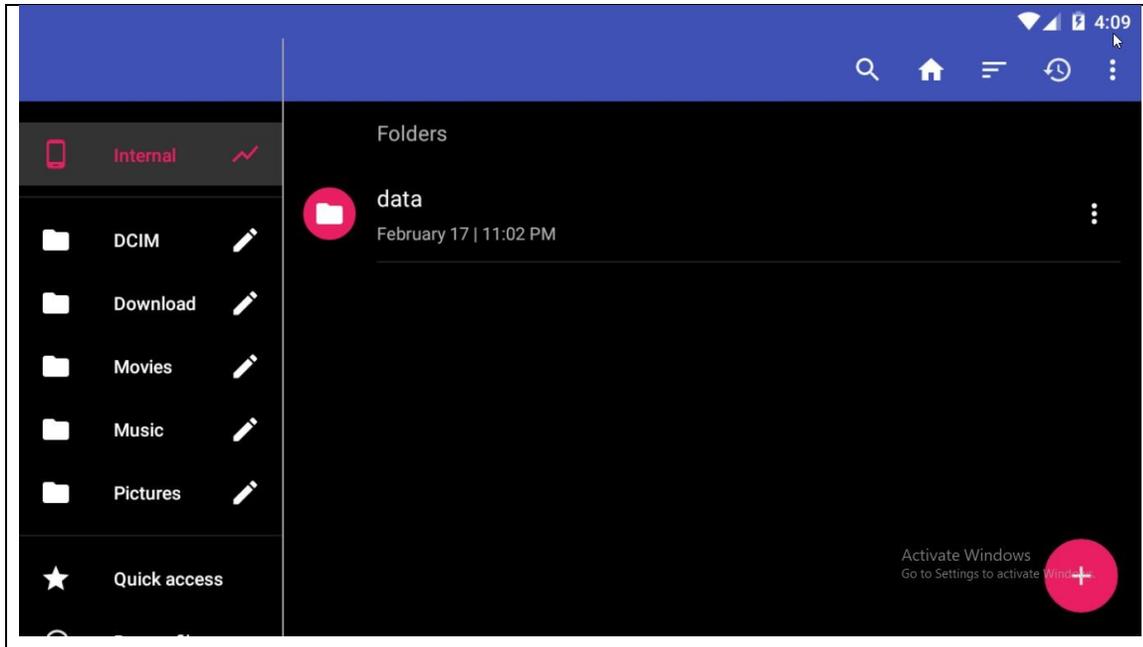
**Prototipo de interfaz:**



Luego de seleccionar la carpeta y hacer clic en la opción eliminar.



Luego de seleccionar la opción eliminar en la interfaz de dialogo de usuario.



## 2.5. Diseño

El diseño permite modelar el sistema o producto que se va a construir. Este modelo se evalúa respecto de la calidad y su mejora antes de generar código; después, se efectúan pruebas y se involucra a muchos usuarios finales. El diseño es el lugar en el que se establece la calidad del software (Pressman 2005a). Esta actividad del ciclo de vida de ingeniería de software debe describir la arquitectura de construcción (Luca Crisan 2019).

### 2.5.1 Descripción de la arquitectura

El diseño arquitectónico es un proceso creativo en el que se intenta establecer una organización del sistema que satisfaga los requisitos funcionales y no funcionales del propio sistema. Debido a que es un proceso creativo, las actividades dentro del proceso difieren radicalmente dependiendo del tipo de sistema a desarrollar, el conocimiento y la experiencia del arquitecto del sistema, y los requisitos específicos del mismo (Sommerville 2005).

Es decir, la arquitectura de software es una forma de representar sistemas mediante el uso de la abstracción, de forma que aporte el más alto nivel de comprensión de los mismos. Esta representación incluye los componentes fundamentales del software, su comportamiento y formas de interacción para satisfacer los requisitos del sistema. Según Roger Pressman: “En su forma más simple, la arquitectura del software es la estructura u organización de los componentes del programa, la manera en que estos interactúan y la estructura de datos que utilizan” (Pressman 2005a).

Para el desarrollo de la solución propuesta en el presente trabajo de diploma se propone una arquitectura Modelo-Vista-Presentador (MVP, Model-View-Presenter) teniendo en cuenta principalmente su bien definida separación de conceptos, su adecuada adaptación al Android SDK y a su amplio uso en el desarrollo de aplicaciones Android.

Se presenta a continuación una vista lógica de esta arquitectura utilizando el diagrama de paquetes proporcionado por UML.

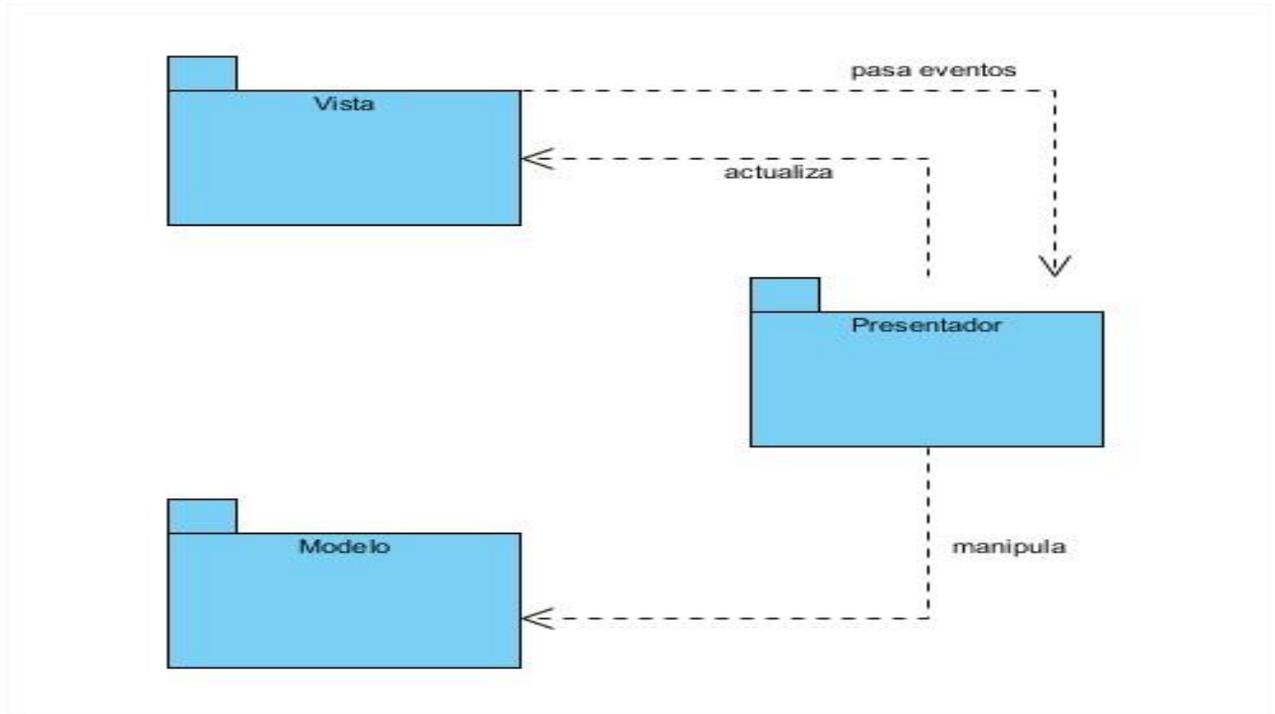


Figura1.Vista lógica de la arquitectura MVP

Como es posible observar en la Figura 1, el sistema se estructura en tres componentes lógicos que interactúan entre sí. El presentador es responsable de actuar como hombre en el medio entre la vista y el modelo, este recupera datos del modelo y los devuelve con un formato específico a la vista, en diferencia con la arquitectura Modelo-Vista-Controlador (MVC, Model-View-Controller), este también decide que sucede cuando el usuario interactúa con ella. La vista, posee referencia al presentador, lo único que debe hacer es llamar a un método del presentador cada vez que ocurre una acción en la interfaz. Por último, el modelo debe ser solamente un portal a la lógica de negocio, se puede ver como el proveedor de los datos que se desean mostrar en la vista.

**Vista:** se encarga de manejar los aspectos visuales. Mantiene una referencia a su Presentador, al cual le delega la responsabilidad del manejo de los eventos.

**Presentador:** contiene la lógica para responder a los eventos y manipula el estado de la vista. El Presentador utiliza el modelo para saber cómo responder a los eventos y es responsable de establecer y administrar el estado de una vista.

**Modelo:** está compuesto por los objetos que conocen y manejan los datos dentro de la aplicación.

Para el desarrollo del diseño arquitectónico se tuvo como base la arquitectura descrita en distintos proyectos desarrollados por Google. Incluyen una serie de ejemplos con diferentes arquitecturas con el objetivo de demostrar estrategias para ayudar a resolver problemas comunes como son clases extensas, inconsistente nombrado de esquemas así como también desajustes o arquitecturas faltantes (A collection of samples to discuss and showcase different architectural tools and patterns for Android apps 2018).

Se muestra a continuación el diagrama de paquetes de la solución a desarrollar.

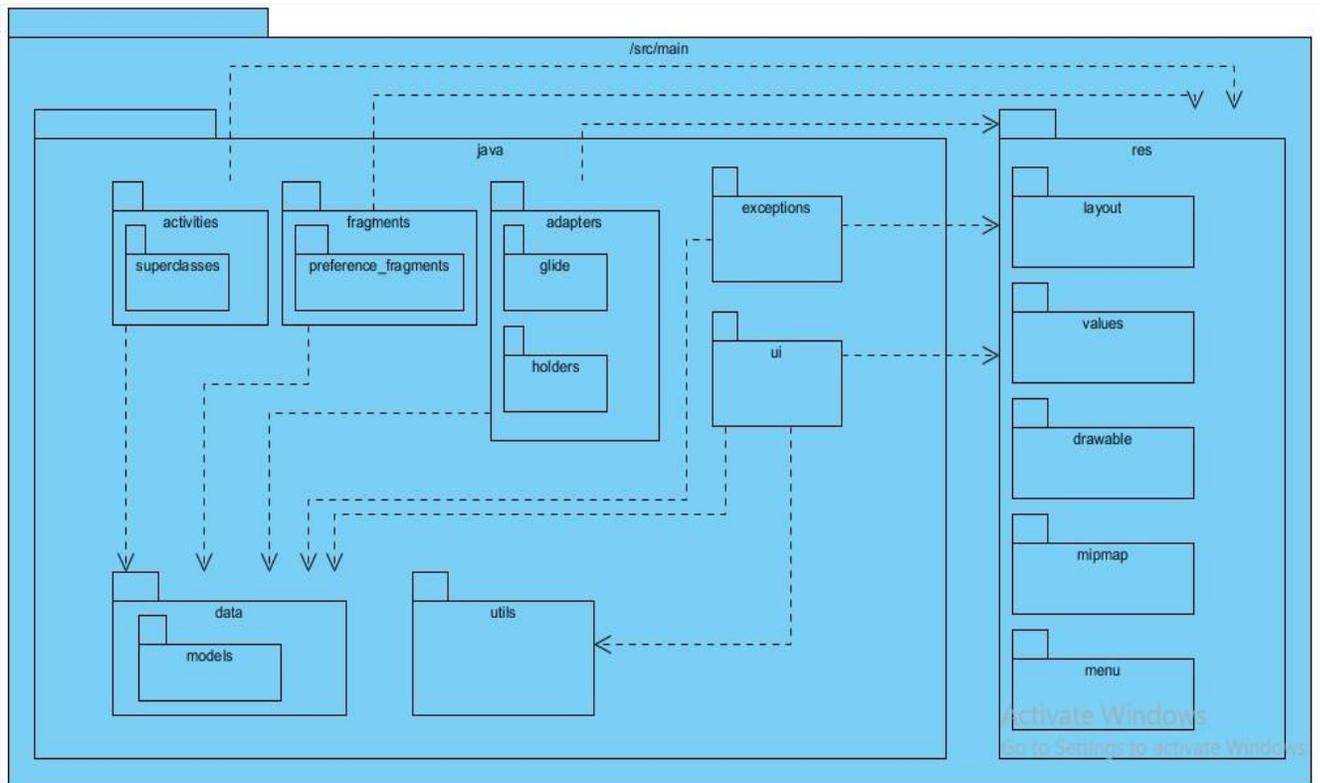


Figura 2: Diagrama de paquetes de la solución

El código de la solución estará contenido en dos paquetes: res y el java. En el primero se almacena el código con lenguaje de marcado extensible o como bien es conocido formato XML, que define las vistas, las imágenes y las cadenas de texto que se utilizan en la aplicación en distintos idiomas para su

configuración. En el segundo estarán contenidas todas las clases Java de la solución, agrupadas en dependencia de su funcionalidad.

El paquete java contiene los paquetes database y utils, constituyendo el modelo de la aplicación, encargados de almacenar las clases para el acceso de datos, la persistencia de datos y funcionalidades útiles respectivamente. Tiene el paquete activities que concentra todas las clases lógicas asociadas a las actividades o pantallas de la aplicación. Se encuentran fragments que tiene todo el código lógico correspondiente a los fragmentos que son porciones de la interfaz de usuario que puede añadirse o eliminarse de la interfaz de forma independiente al resto de elementos de la actividad. Adapters reúne el código asociado a los adaptadores que son un mecanismo de Android que hace de puente entre los datos y las vistas contenidas en un ListView, GridView o cualquier otro elemento. Exceptions tiene las clases para gestionar los posibles errores de la aplicación a partir del código implementado en estas. El paquete ui integra los archivos necesarios para gestionar las vistas de la aplicación.

El presentador aloja la lógica relacionada con cada evento ocurrido en la vista y accede a los datos necesitados en el paquete database o las funcionalidades en utils, y la vista correspondiente maneja el trabajo de la interfaz de usuario (IU) de Android. Esta última casi no contiene lógica; convierte los comandos del presentador a acciones de IU y escucha las acciones del usuario, que luego se pasan a este.

### **2.5.2 Patrones de diseño**

Los patrones de diseño están relacionados con el diseño de los objetos y frameworks de pequeña y mediana escala. Aplicables al diseño de una solución para conectar los elementos de gran escala que se definen mediante los patrones de arquitectura, y durante el trabajo de diseño detallado para cualquier aspecto del diseño local (Larman 2003).

Los patrones de diseño refinan los componentes basados en la experiencia obtenida. Son una solución estándar para un problema común de programación, ya que ayudan a mantener un código reutilizable y tener mayor control sobre los problemas recurrentes, por eso, son muy utilizados en el desarrollo de en múltiples aplicaciones (Larman 2003).

**Estos patrones se dividen en dos grupos:**

#### **Patrones GRASP**

Los patrones GRASP constituyen un apoyo para la enseñanza que ayuda a uno a entender el diseño de objetos esencial, y aplica el razonamiento para el diseño de una forma sistemática, racional y

explicable. Este enfoque para la comprensión y utilización de los principios de diseño se basa en los patrones de asignación de responsabilidades (Larman y Applying 2004).

**Experto:** se usa más que cualquier otro al asignar responsabilidades, es un principio básico que suele utilizarse en el diseño orientado a objetos (Larman y Applying 2004). Consiste en la asignación de una responsabilidad a la clase que cuenta con la información necesaria para llevarla a cabo. El uso de este patrón da pie a un bajo acoplamiento y una alta cohesión, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento. El cumplimiento de una responsabilidad requiere a menudo información distribuida en varias clases de objetos. Este patrón se evidencia en la clase ZipService.java, ExtractService; estas clases son expertas en comprimir y extraer archivos respectivamente.

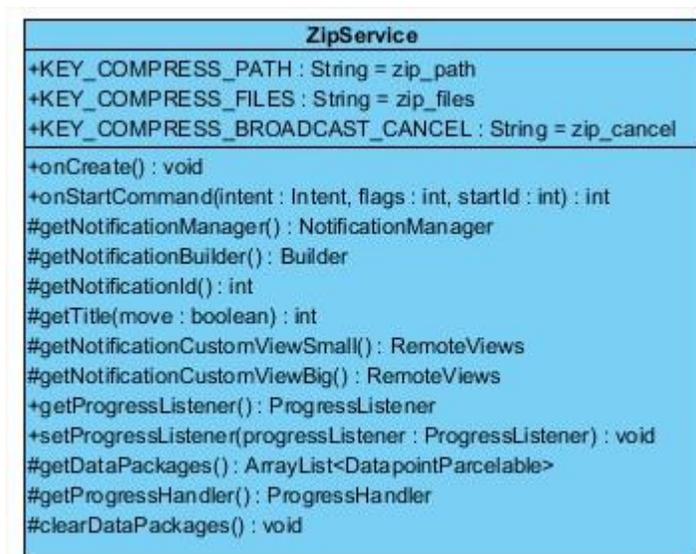


Figura 3. Aplicación del patrón experto en la clase ZipService

**Bajo Acoplamiento:** el acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras (Larman y Applying 2004). El bajo acoplamiento soporta el diseño de clases más independientes y reutilizables, lo cual reduce el impacto de los cambios y acrecienta la oportunidad de una mayor productividad. Ejemplo del uso de este patrón en la solución propuesta se muestra en las clases: AnimUtils, RootUtils, ScreenUtils, BookSorter, ComputerParcelable y DataPointParcelable donde se minimizan las relaciones de estas con el resto de las clases por lo que permite que sean reutilizadas con facilidad.

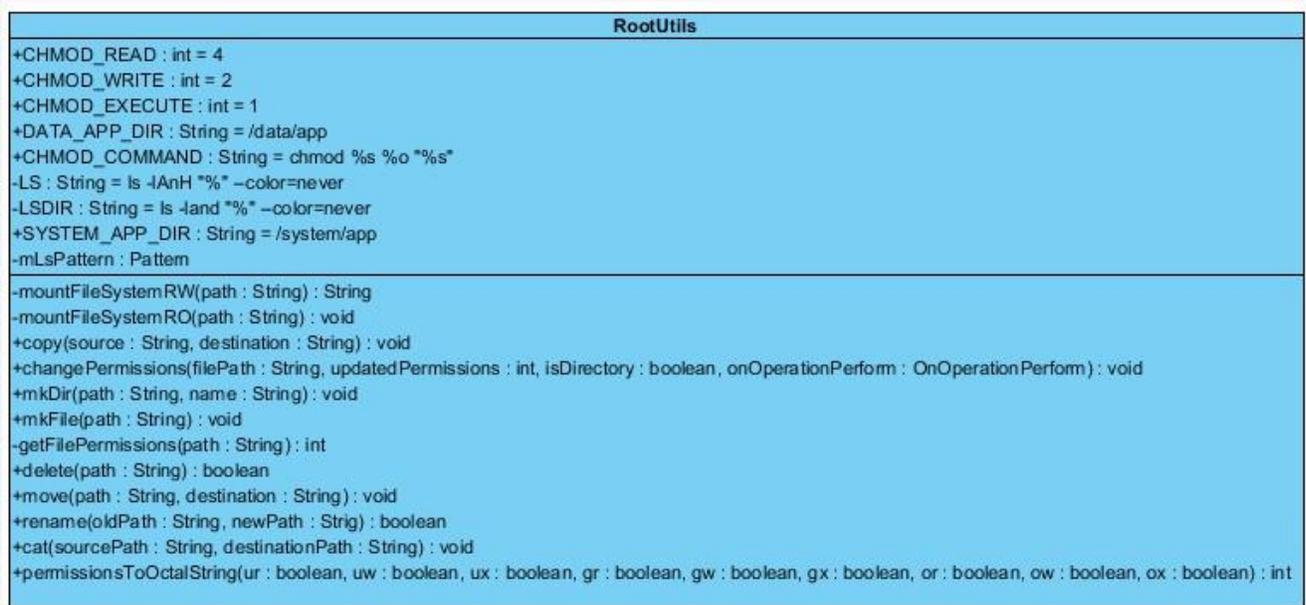


Figura 4. Aplicación del patrón bajo acoplamiento en la clase RootUtils

**Alta cohesión:** cada elemento del diseño debe realizar una labor única dentro del sistema, lo cual expresa que la información que almacena una clase, debe ser coherente y estar lo más relacionada con ella posible. Este patrón es el encargado de asignar responsabilidades, de manera que la información que se almacena en una clase, sea la necesaria y esté bien delimitada (Larman y Applying 2004). En el diseño de la solución que se propone, se evidencia este patrón en la clase AppThemeManager, clase encargada de administrar los temas de la aplicación, y para realizar esta tarea se relaciona directamente con la clase AppTheme.

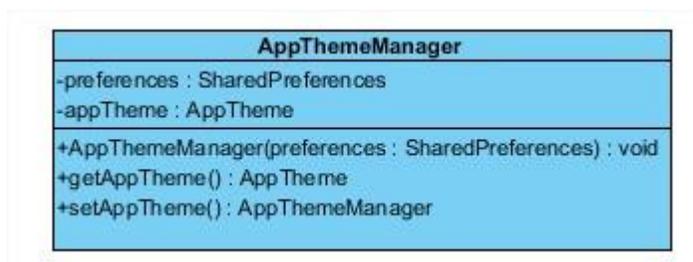


Figura 5. Aplicación del patrón alta cohesión en la clase AppThemeManager

### Patrones GOF

Los patrones GOF son alternativas de solución a problemas conocidos, pero son mucho más específicas las situaciones en las que se aplican. Los cuales se clasifican en tres grupos fundamentales: los patrones Creacionales que es donde se inicializan y configuran los objetos, los patrones

Estructurales donde se separa la interfaz de la implementación y los patrones de Comportamiento los cuales describen el comportamiento entre las clases y los objetos (Larman y Applying 2004).

**Singleton:** Está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso global a ella (Larman y Applying 2004). Se empleó en la clase AppConfig.java y MainActivityHelper debido a la necesidad de que exista una sola instancia de cada una con objetivo de minimizar el consumo de recursos de memoria.

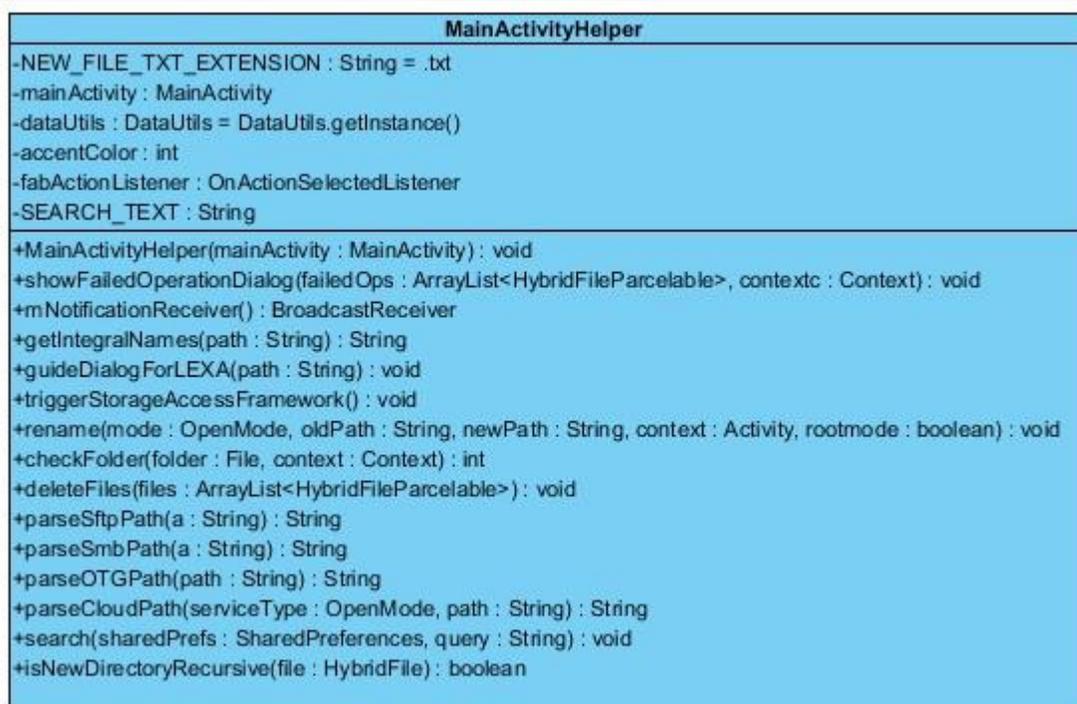


Figura 6. Aplicación del patrón Singleton en la clase MainActivityHelper

### 2.5.3 Diagrama de clases del diseño

Un Diagrama de Clases de Diseño (DCD) muestra la especificación para las clases software de una aplicación. Incluye la siguiente información:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Navegabilidad.

- Dependencias.

A diferencia de las clases conceptuales del Modelo del Dominio, las clases de diseño de los DCD muestran las definiciones de las clases software en lugar de los conceptos del mundo real (Diagrama de Clases de Diseño | LENGUAJE DE MODELADO UNIFICADO 2015). A continuación se muestra un fragmento del diagrama de clases del diseño relativo a la lógica de acceso a datos y a la vista de la propuesta de solución:

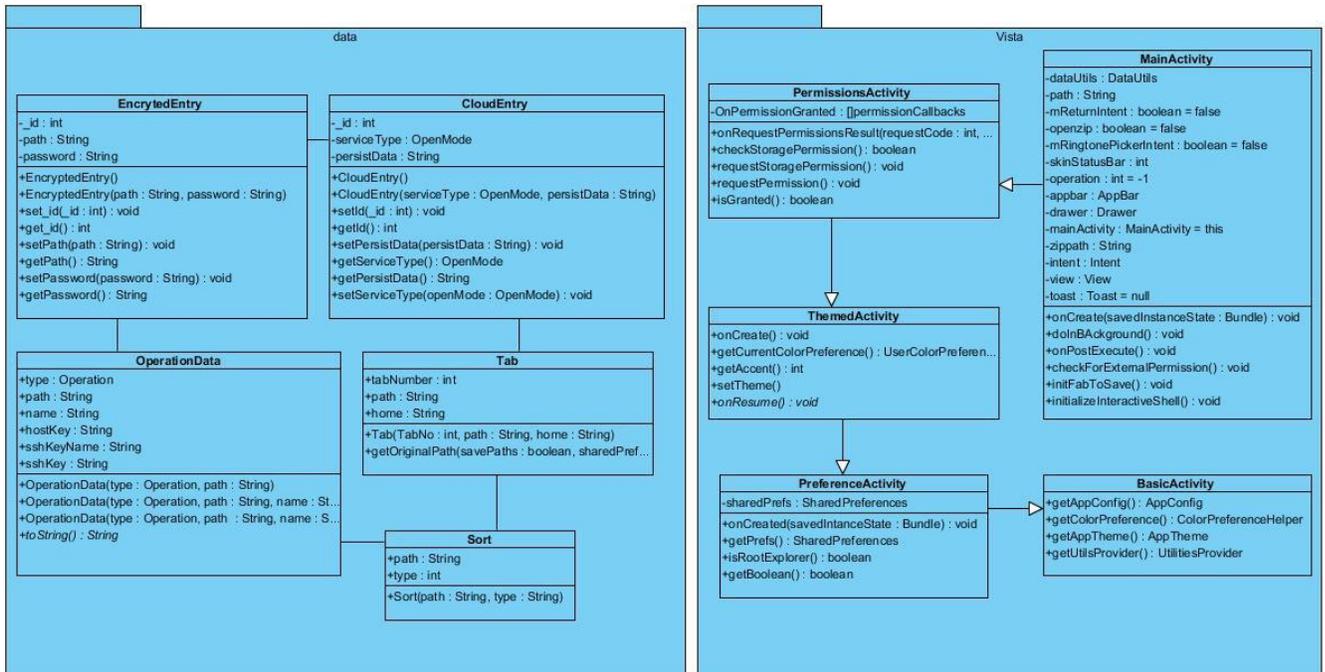


Figura 7. Fragmento del diagrama de clases del diseño relativo a la lógica de acceso a datos y vista de la propuesta solución.

## 2.6. Conclusiones parciales

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

- La captura de los requisitos y la elaboración de las historias de usuario correspondientes permite comprender mejor las funcionalidades de la aplicación que se desea desarrollar y el comportamiento de la misma.
- La utilización de la arquitectura MVP y los patrones de diseño contribuyen al diseño de la aplicación, proporcionando una estructura para la misma y posibilitando el empleo de buenas prácticas de programación y la reutilización de código.

- Como resultado principal de este capítulo queda plasmada la propuesta de solución para la problemática planteada, detallada a través de los requisitos funcionales y no funcionales, las historias de usuario, la arquitectura y el uso de patrones de diseño.

## **CAPÍTULO 3. Implementación y prueba del navegador de archivo para NovaDroid**

### **3.1. Introducción**

La fase de implementación en el desarrollo de un producto de software, es el mecanismo donde se ponen en práctica todas las descripciones y arquitecturas propuestas en las fases de análisis y diseño, es el complemento del trabajo de las fases que lo preceden dentro del proceso de desarrollo de software. Para un despliegue exitoso del navegador primero este debe pasar por un conjunto de pruebas las cuales permiten validar el correcto funcionamiento. En el presente capítulo se exponen las especificaciones asociadas a la implementación de la aplicación, se describe el estándar de codificación empleado, como también se describe el diseño de las pruebas realizadas a la aplicación y los resultados obtenidos durante cada iteración. Además, se explican de manera breve los resultados arrojados por las pruebas realizadas a las diferentes funcionalidades.

### **3.2. Código fuente**

Para obtener una versión funcional de la aplicación se deben implementar los componentes que se han definido, como resultado se obtienen archivos que contienen el código fuente de la aplicación. El código fuente de un software es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto, en el código fuente de un programa está escrito su funcionamiento. Estas instrucciones son escritas en un lenguaje de programación que consiste en un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones (Lazo Y 2008).

#### **3.2.1 Estándares de codificación**

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez (Arevalo Lizardo 2012).

Para facilitar el entendimiento del código y fijar un modelo a seguir, se establece el estándar de codificación. A continuación, se muestran algunos de los aspectos de este estándar para el lenguaje Java utilizados en el entorno integrado de desarrollo Android Studio.

## Nombres de clases y métodos

Para la definición de las clases y métodos en el código de la aplicación se utilizó el estándar CamelCase. Este es un estilo de escritura que se aplica a frases o palabras compuestas. Existen dos tipos de estándares de CamelCase (Rogers 2014).

- **UpperCamelCase:** cuando la primera letra de cada una de las palabras es mayúscula.  
Ejemplo: EjemploDeUpperCamelCase.
- **lowerCamelCase:** igual que la anterior con la excepción de que la primera letra es minúscula.  
Ejemplo: ejemploDeLowerCamelCase.

En la definición del nombre de las clases fue utilizado UpperCamelCase. Ver figura 8.

En la nomenclatura de los métodos fue utilizado lowerCamelCase. Ver figura 9.

```
public class MainActivity extends PermissionsActivity implements SmbConnectionListener,
    DataChangeListener, BookmarkCallback, SearchWorkerFragment.HelperCallbacks,
    CloudConnectionCallbacks, LoaderManager.LoaderCallbacks<Cursor> {

    public static final Pattern DIR_SEPARATOR = Pattern.compile("/");
    public static final String TAG_ASYNC_HELPER = "async_helper";

    private DataUtils dataUtils;

    public String path = "";
    public boolean mReturnIntent = false;
    public boolean openzip = false;
    public boolean mRingtonePickerIntent = false;
    public int skinStatusBar;

    public SpeedDialView floatingActionButton;

    public MainActivityHelper mainActivityHelper;

    public int operation = -1;
    public ArrayList<HybridFileParcelable> oparraylist;
    public ArrayList<ArrayList<HybridFileParcelable>> oparraylistList;

    // oppathe - the path at which certain operation needs to be performed
    // oppathel - the new path which user wants to create/modify
    // oppathlist - the paths at which certain operation needs to be performed (pairs with
    public String oppathe, oppathel;
    public ArrayList<String> oppathelist;
```

Figura 8. Ejemplo del estilo UpperCamelCase en la declaración de la clase principal de la aplicación.

```

public Void doInBackground() {
    dataUtils.setHiddenFiles(utilsHandler.getHiddenFilesConcurrentRadixTree());
    dataUtils.setHistory(utilsHandler.getHistoryLinkedList());
    dataUtils.setGridfiles(utilsHandler.getGridViewList());
    dataUtils.setListfiles(utilsHandler.getListViewList());
    dataUtils.setBooks(utilsHandler.getBookmarksList());
    ArrayList<String[]> servers = new ArrayList<String[]>();
    servers.addAll(utilsHandler.getSmbList());
    servers.addAll(utilsHandler.getSftpList());
    dataUtils.setServers(servers);

    return null;
}

@Override
public void onPostExecute(Void result) {
    drawer.refreshDrawer();

    if (savedInstanceState == null) {
        if (openProcesses) {
            FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();
            transaction.replace(R.id.content_frame, new ProcessViewerFragment(), KEY_INTENT_PROCESS_VIEWER);
            //transaction.addToBackStack(null);
            drawer.setSomethingSelected(true);
            openProcesses = false;
        }
    }
}

```

Figura 9. Ejemplo del estilo lowerCamelCase en la declaración de métodos de la clase principal.

## Estructura

- Los nombres de las variables responderán al estilo de capitalización lowerCamelCase. Ver figura 8.
- No se usarán nombres de variables que coincidan con palabras reservadas.
- Los nombres de variables globales deberán escribirse todo en mayúsculas con las palabras separadas por un guion bajo ("\_"). Todas serán declaradas como public static. Ver figura 8.
- Se inicializarán las variables locales donde se declaran. La única razón para no hacerlo será si su valor inicial depende de cálculos posteriores. Ver figura 8.
- El código será tabulado y espaciado a través del formato que aplica la combinación de teclas ALT+Shift+F definida en la configuración del IDE Android Studio, en la sección referente a los atajos de teclado (File/Settings/Keymap).
- El idioma inglés se empleará para la codificación de la solución.

### 3.3. Diagrama de despliegue

El modelo de despliegue consiste en un modelo de objetos que tiene como objetivo describir la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre las estaciones de trabajo (denominados nodos). La vista de despliegue representa la disposición de las instancias de

componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Los nodos representan objetos físicos existentes en tiempo de ejecución, sirven para modelar recursos que tiene memoria y capacidad de proceso y puede ser tanto ordenadores como dispositivos, memoria o personas (Sommerville 2011).



Figura 10. Diagrama de despliegue

### 3.4. Validación del sistema

Terminada la implementación del producto, es necesario realizarle pruebas con el objetivo de detectar errores en la aplicación y en la documentación; este proceso resulta de gran importancia ya que da una medida de la calidad del mismo siempre que se aplique una estrategia de prueba de forma correcta. Una estrategia de prueba de software proporciona una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos se requerirán. Por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de la prueba y la recolección y evaluación de los resultados (Pressman 2010). A continuación, se muestran las pruebas realizadas al sistema y los resultados obtenidos por cada una.

#### 3.4.1 Pruebas de unidad

Las pruebas de unidad o unitarias son el proceso de probar componentes del programa tales como métodos o clases de objetos. Las funciones o los métodos individuales son el tipo más simple de componente. Las pruebas deben llamarse para dichas rutinas con diferentes parámetros de entrada (Sommerville 2011).

El método aplicado para esta prueba fue el de caja blanca, donde las pruebas se enfocan en la estructura de control del programa. 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.

Con el fin de automatizar este tipo de pruebas sobre la solución se decidió emplear la herramienta JUnit, la cual está integrada con el IDE Android Studio que se utilizó para desarrollar la solución. Se implementaron dos casos de prueba a través de la clase ShadowCryptUtilTest, AppThemeTest, donde se comprueba el correcto funcionamiento de las clases ShadowCryptUtil y AppTheme, siendo esta última la clase encargada de gestionar los temas de la aplicación.

### 3.4.2 Pruebas de validación

La prueba de validación proporciona un aseguramiento final de que el software cumple con todos los requisitos funcionales, de comportamiento y desempeño. La prueba se concentra en las acciones visibles para el usuario y en la salida que este puede reconocer. La validación se alcanza cuando el software funciona de tal manera que satisface las expectativas razonables (especificación de requisitos de software) del cliente. Se logra mediante una serie de pruebas que demuestran que se cumple con los requisitos (Pressman 2010).

Para realizar estas pruebas se emplearon pruebas funcionales, ya que aseguran el apropiado trabajo de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados. Las metas de estas pruebas son verificar la apropiada aceptación de datos y verificar el procesamiento, recuperación e implementación adecuada de las reglas del negocio. Para llevarlas a cabo, el método a emplear fue el de caja negra, el cual se centra en los requisitos funcionales del software. Es decir, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa (Pressman 2010).

Los casos de prueba se diseñaron según las funcionalidades 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 plantilla de casos de pruebas recoge la especificación de una historia de usuario, dividida en secciones y escenarios, detallando las funcionalidades descritas en ella y describiendo cada variable.

A continuación, se muestran los casos de prueba correspondientes a las historias de usuario “Crear Carpeta”, “Copiar carpetas o archivos” y “Eliminar carpetas o archivos” debido a que estas responden a funcionalidades importantes dentro de la propuesta de solución.

Tabla 7: Caso de prueba de la Historia de Usuario No1 “Crear Carpeta”.

Escenario	Descripción	Respuesta de la aplicación	Flujo central
EC1 Crear Carpeta	La aplicación al ser ejecutada debe permitir al usuario	Crea una carpeta nueva dentro de la estructura de archivos y	El usuario debe navegar hasta la ubicación donde desea crear la carpeta.

	crear una carpeta.	carpetas del sistema operativo.	<p>El sistema muestra botón de menú de opciones, luego del usuario hacer clic en el botón lista las opciones de menú.</p> <p>El usuario selecciona la opción “Crear carpeta”.</p> <p>El sistema permite al usuario introducir un nombre para la nueva carpeta.</p> <p>El usuario selecciona la opción crear.</p> <p>El sistema debe listar la carpeta en la ubicación deseada.</p>
--	--------------------	---------------------------------	--

Tabla 8: Caso de prueba de la Historia de Usuario No2 “Copiar carpetas o archivos”.

Escenario	Descripción	Respuesta de la aplicación	Flujo central
EC1 Copiar carpetas o archivos	La aplicación al ser ejecutada debe permitir al usuario copiar un archivo o una carpeta.	Copia un archivo o una carpeta desde una ubicación a otra dentro de la estructura de archivos y carpetas del sistema operativo.	<p>El usuario debe navegar hasta la ubicación donde desea copiar la carpeta.</p> <p>El sistema permite al usuario seleccionar la carpeta que desea copiar.</p> <p>El sistema muestra botón de menú de opciones, luego del usuario hacer clic en el botón lista las opciones de menú.</p> <p>El usuario selecciona la opción “Copiar”.</p> <p>El usuario debe navegar hasta la ubicación donde desea pegar la carpeta que fue copiada.</p> <p>El sistema permite al usuario seleccionar dentro del menú de opciones la opción “Pegar”.</p> <p>El sistema debe listar la carpeta en la ubicación deseada.</p>

Tabla 9: Caso de prueba de la Historia de Usuario No3 “Eliminar carpetas o archivos”.

Escenario	Descripción	Respuesta de la aplicación	Flujo central
EC1 Eliminar carpetas o archivos	La aplicación al ser ejecutada debe permitir al usuario	Elimina un archivo o una carpeta dentro de la estructura de	<p>El usuario debe navegar hasta la ubicación donde desea eliminar la carpeta.</p> <p>El sistema permite al usuario seleccionar la carpeta que desea eliminar.</p>

	eliminar un archivo o una carpeta.	archivos y carpetas del sistema operativo.	<p>El sistema muestra botón de menú de opciones, luego del usuario hacer clic en el botón lista las opciones de menú.</p> <p>El usuario selecciona la opción "Eliminar".</p> <p>El sistema debe mostrar una caja de diálogo que permita cancelar o confirmar la opción deseada.</p> <p>El sistema debe eliminar la carpeta en la ubicación deseada.</p>
--	------------------------------------	--	---

### 3.4.3 Pruebas de aceptación

Esta es la etapa final en el proceso de pruebas, antes de que el sistema se acepte para uso operacional. El sistema se pone a prueba con datos suministrados por el cliente del sistema, en vez de datos de prueba simulados. Las pruebas de aceptación revelan los errores y las omisiones en la definición de requerimientos del sistema, ya que los datos reales ejercitan el sistema en diferentes formas a partir de los datos de prueba. Asimismo, las pruebas de aceptación revelan problemas de requerimientos, donde las instalaciones del sistema en realidad no cumplan las necesidades del usuario o cuando sea inaceptable el rendimiento del sistema (Sommerville 2011).

Para llevar a cabo las pruebas de aceptación se hizo necesario de a través del tipo de prueba alfa incorporar un cliente o usuario final directamente al proceso de prueba de la aplicación. Entiéndase por prueba alfa cuando esta se lleva a cabo en el sitio del desarrollador por un grupo representativo de usuarios finales, es decir, en un ambiente controlado, propiciando que el desarrollador pueda registrar errores y problemas de uso.

### 3.4.4 Aplicación y resultados de las pruebas

Para la ejecución de las pruebas de validación y de aceptación de tipo alfa se utilizaron los siguientes dispositivos:

#### Laptop:

- ✓ Modelo: HP Pavilion
- ✓ Procesador: Intel(R) Core(TM) i5-8250U
- ✓ CPU: @ 1.60Hz 1.80Gz
- ✓ RAM: 8.00 GB

#### Emulador NOX:

- ✓ Versión: 6.6.0.1
- ✓ Sistema Operativo: Android 5
- ✓ SDK: 28
- ✓ Procesador: Intel(R) Core(TM) i5-8250
- ✓ RAM: 1.5 GB

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). A continuación se describen los aspectos que se tuvieron en cuenta en cada clasificación.

NCS: son las no conformidades referentes a las funcionalidades de la aplicación: validaciones incorrectas o respuestas de la aplicación diferente a lo descrito previamente en las historias de usuario.

NCNS: son las no conformidades en cuanto al diseño de la propuesta de solución y errores ortográficos. Fueron realizadas 4 iteraciones de pruebas, ejecutándose al término de cada una de ellas pruebas de regresión, con el objetivo de asegurar que al resolverse las no conformidades detectadas, estas no introdujeran nuevos errores en la solución.

En la primera iteración se detectaron 4 NCS y 2 NCNS, las cuales fueron resueltas satisfactoriamente.

No Conformidades Significativas:

1. La aplicación se detuvo al cambiar la vista de los elementos de la aplicación.
2. La aplicación se detuvo al modificar el tema y el color de la misma.
3. La aplicación no validó extraer los elementos de una carpeta comprimida.
4. La aplicación no mostró el archivo comprimido luego de haberse ejecutado la funcionalidad para comprimir.

No Conformidades No Significativas:

1. La aplicación posee faltas de ortografía en algunas etiquetas de diferentes vistas.
2. La aplicación posee etiquetas sin traducir al idioma inglés.

En la segunda iteración fueron detectadas 3 NCS y 2 NCNS, estas siendo también solucionadas de igual manera que en la iteración anterior.

No Conformidades Significativas:

1. La aplicación no validó que al presionar *backspace* en el teclado se moviera hacia atrás en la ruta de archivos y carpetas.
2. Al presionar el botón flotante no lista sus opciones.
3. La aplicación se detuvo al buscar un elemento dentro de la estructura de archivos y carpetas del sistema operativo.

No Conformidades No Significativas:

1. La aplicación muestra incorrectamente los textos presentes en la vista que se muestra al seleccionar la opción "Acerca de", que se encuentra en el panel desplegable.
2. Se detectaron en la lógica de la aplicación incorrectas llamadas por el id hacia los elementos de la vista del programa.

En la tercera iteración solo fue detectada una NCS, la cual fue solucionada satisfactoriamente.

No Conformidades Significativas:

1. La aplicación no muestra el texto de diálogo con el usuario al seleccionar la opción crear carpeta que permite al mismo renombrar esta.

Por último en la cuarta iteración no fueron detectadas NCS ni NCNS, demostrándose entonces que la aplicación cumple con los requisitos funcionales establecidos y en consecuencia se considera concluida. Se presenta a continuación un gráfico con un resumen de los resultados obtenidos en las cuatro iteraciones realizadas:

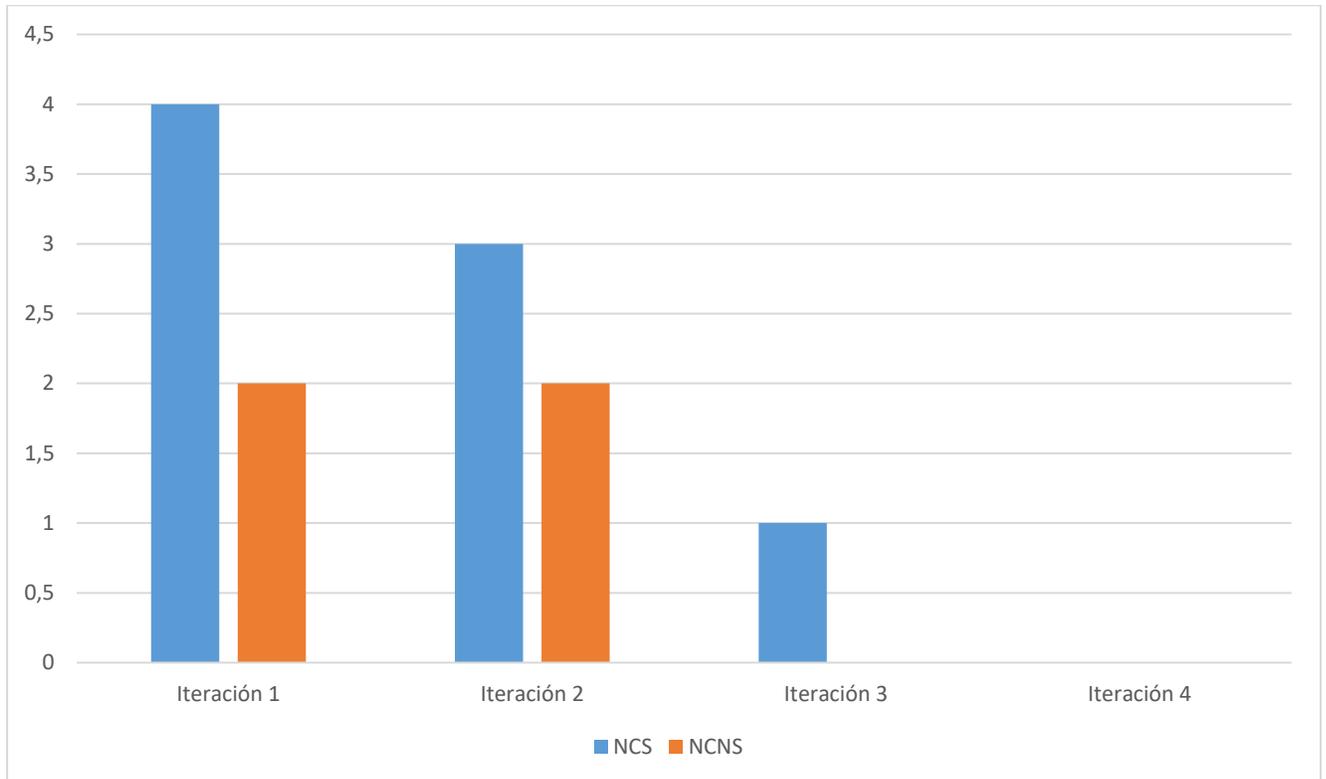


Figura 11. Gráfico de barra de No Conformidades

### 3.5. Conclusiones parciales

- Se especifica el uso del estándar de codificación para lograr un estilo claro y organizado del código durante la fase de implementación, esto permite un adecuado entendimiento del código fuente por otros programadores ajenos al equipo de desarrollo.
- Con el propósito de mostrar los componentes del sistema y sus relaciones, se realiza el diagrama de componentes.

- La aplicación de las pruebas unitarias, de validación y de aceptación permitieron comprobar el correcto funcionamiento del código de la solución, validar la completitud de los requisitos y determinar la aceptación por parte del cliente.

## CONCLUSIONES GENERALES

Una vez finalizada la investigación y el desarrollo del presente trabajo se concluye que:

- El estudio de los principales conceptos relacionados a un navegador de archivo y a la gestión de archivos y carpetas de un sistema operativo, permitió sentar las bases para el desarrollo e implementación de un navegador de archivo para NovaDroid.
- El diseño de un navegador de archivo guiado por la metodología AUP variación UCI, haciendo uso del lenguaje de modelado UML y mediante el apoyo de la herramienta CASE Visual Paradigm para UML posibilitó la obtención de una solución escalable, ofreciendo facilidad para el desarrollo del navegador.
- La implementación de la aplicación permitió navegar por la estructura de archivos y carpetas del sistema operativo NovaDroid, sirvió para darle respuesta a los requisitos definidos y de esa manera responder al problema de investigación planteado.
- La ejecución de pruebas a la aplicación siguió una estrategia de pruebas, que contempla pruebas unitarias, pruebas de validación y pruebas de aceptación arrojando como resultado un total de 12 no conformidades, corregidas en tres iteraciones, lo cual permitió la validación de todos los requisitos identificados y la aceptación por parte del cliente.

## **RECOMENDACIONES**

Una vez implementada la solución el autor recomienda:

Desarrollar versiones de la aplicación NovaDroidFile para así incorporar nuevas funcionalidades o comportamientos al software.

Desarrollar el mismo para otros sistemas operativos.

## REFERENCIAS BIBLIOGRÁFICAS

- A COLLECTION OF SAMPLES TO DISCUSS AND SHOWCASE DIFFERENT ARCHITECTURAL TOOLS AND PATTERNS FOR ANDROID APPS, android, 2018. Kotlin. S.l.: Android. [Consulta: 29 enero 2020]. Disponible en: <https://github.com/android/architecture-samples>.
- ALEGSA, L., 2016a. Definición de Administrador de archivos (aplicación). [en línea]. [Consulta: 15 noviembre 2019]. Disponible en: [http://www.alegsa.com.ar/Dic/administrador\\_de\\_archivos.php](http://www.alegsa.com.ar/Dic/administrador_de_archivos.php).
- ALEGSA, L., 2016b. Definición de Directorio (informática). [en línea]. [Consulta: 23 enero 2020]. Disponible en: <http://www.alegsa.com.ar/Dic/directorio.php>.
- ALEGSA, L., 2016c. Definición de eliminar borrar o suprimir (informática). [en línea]. [Consulta: 15 noviembre 2019]. Disponible en: <http://www.alegsa.com.ar/Dic/eliminar.php>.
- ANDROID STUDIO RELEASE NOTES | ANDROID STUDIO, 2017. Android Studio release notes | Android Developers. [en línea]. [Consulta: 27 noviembre 2019]. Disponible en: <https://developer.android.com/studio/releases>.
- AREVALO LIZARDO, M.E., 2012. Propuesta de Estándar de desarrollo o codificación (Primera Entrega) #programacion – Maria Eugenia Arevalo Lizardo. [en línea]. [Consulta: 16 junio 2020]. Disponible en: <https://arevalomaria.wordpress.com/2012/11/02/propuesta-de-estandar-de-desarrollo-o-codificacion-primera-entrega-programacion/>.
- BÁEZ, M., 2012. BÁEZ, Manuel, et al. Introducción a android. EME Madrid, España, 2012, p. 121 - Buscar con Google. [en línea]. [Consulta: 23 enero 2020]. Disponible en: [https://www.google.com/search?ei=SwAqXqvKGcO45gLfxpioBg&q=B%C3%81EZ%2C+Manuel%2C+et+al.+Introducci%C3%B3n+a+android.+EME+Madrid%2C+Espa%C3%B1a%2C+2012%2C+p.+121&oq=B%C3%81EZ%2C+Manuel%2C+et+al.+Introducci%C3%B3n+a+android.+EME+Madrid%2C+Espa%C3%B1a%2C+2012%2C+p.+121&gs\\_l=psy-ab.3...201125.201125..203141...0.0..0.0.....1....2j1..gws-wiz.O-E9wjE\\_Iq4&ved=0ahUKEwjrm\\_uXxprnAhVDnFkKHV8jBmUQ4dUDCAo&uact=5](https://www.google.com/search?ei=SwAqXqvKGcO45gLfxpioBg&q=B%C3%81EZ%2C+Manuel%2C+et+al.+Introducci%C3%B3n+a+android.+EME+Madrid%2C+Espa%C3%B1a%2C+2012%2C+p.+121&oq=B%C3%81EZ%2C+Manuel%2C+et+al.+Introducci%C3%B3n+a+android.+EME+Madrid%2C+Espa%C3%B1a%2C+2012%2C+p.+121&gs_l=psy-ab.3...201125.201125..203141...0.0..0.0.....1....2j1..gws-wiz.O-E9wjE_Iq4&ved=0ahUKEwjrm_uXxprnAhVDnFkKHV8jBmUQ4dUDCAo&uact=5).
- BELLAS, F.G., UNANUE, R.M. y FERNÁNDEZ, V.D.F., 2016. *Lenguajes de programación y procesadores*. S.l.: Editorial Centro de Estudios Ramon Areces SA.
- BUI, D., 2017. Reactive Programming and Clean Architecture in Android Development. ,
- CASTILLO, L., 2019. Características — Android OS 0.1 documentation. [en línea]. [Consulta: 23 enero 2020]. Disponible en: <https://androidos.readthedocs.io/en/latest/data/caracteristicas/>.
- COHN, 2018. Amazon.com: User Stories Applied: For Agile Software Development (Addison-Wesley Signature Series (Beck)) eBook: Mike Cohn: Kindle Store. [en línea]. [Consulta: 29 enero

- 2020]. Disponible en: [https://www.amazon.com/-/es/User-Stories-Applied-Development-Addison-Wesley-ebook-dp-B0054KOL74/dp/B0054KOL74/ref=mt\\_kindle?\\_encoding=UTF8&me=&qid=](https://www.amazon.com/-/es/User-Stories-Applied-Development-Addison-Wesley-ebook-dp-B0054KOL74/dp/B0054KOL74/ref=mt_kindle?_encoding=UTF8&me=&qid=).
- DIAGRAMA DE CLASES DE DISEÑO | LENGUAJE DE MODELADO UNIFICADO, 2015. Diagrama de Clases de Diseño | LENGUAJE DE MODELADO UNIFICADO - UML. [en línea]. [Consulta: 20 febrero 2020]. Disponible en: [http://stadium.unad.edu.co/ovas/10596\\_9836/diagrama\\_de\\_clases\\_de\\_diseo.html](http://stadium.unad.edu.co/ovas/10596_9836/diagrama_de_clases_de_diseo.html).
- ESCALONA, K., 2002. [Consulta: 28 enero 2020]. Disponible en: [https://www.google.com/search?source=hp&ei=i80wXoraCuj85gLXmrKgBA&q=Escalona%2C+Koch+2002&oq=Escalona%2C+Koch+2002&gs\\_l=psy-ab.3...7016.7016..8005...0.0..0.0.....0....2j1..gws-wiz.PrvplostfMY&ved=0ahUKEwjKy-PiwqfnAhVovlkKHVeNDEQQ4dUDCAU&uact=5](https://www.google.com/search?source=hp&ei=i80wXoraCuj85gLXmrKgBA&q=Escalona%2C+Koch+2002&oq=Escalona%2C+Koch+2002&gs_l=psy-ab.3...7016.7016..8005...0.0..0.0.....0....2j1..gws-wiz.PrvplostfMY&ved=0ahUKEwjKy-PiwqfnAhVovlkKHVeNDEQQ4dUDCAU&uact=5).
- EXES, C. de I. a J., 2015. Cursos oracle,cursos java,master oracle,master java,formación en informatica. [en línea]. [Consulta: 19 febrero 2020]. Disponible en: [https://www.mundojava.net/caracteristicas-del-lenguaje.html?Pg=java\\_inicial\\_4\\_1.html](https://www.mundojava.net/caracteristicas-del-lenguaje.html?Pg=java_inicial_4_1.html).
- FLANAGAN, D., 2005. *Java in a Nutshell*. S.l.: O'Reilly Media, Inc. ISBN 978-1-4493-6668-1.
- GRADLE: ADAPTABLE, FAST AUTOMATION FOR ALL, 2019. *Adaptable, fast automation for all. Contribute to gradle/gradle development by creating an account on GitHub* [en línea]. Groovy. S.l.: Gradle. [Consulta: 27 noviembre 2019]. Disponible en: <https://github.com/gradle/gradle>.
- HOWLING, P., 2019. Administrador de archivos - Howling Pixel. [en línea]. [Consulta: 27 noviembre 2019]. Disponible en: [https://howlingpixel.com/i-es/Administrador\\_de\\_archivos](https://howlingpixel.com/i-es/Administrador_de_archivos).
- IAN SOMERVILLE, 2005. *Ingeniería de Software*. S.l.: s.n.
- ICTEA, 2019. ¿Qué es el Sistema Operativo Android? - Base de Conocimientos - ICTEA. [en línea]. [Consulta: 27 noviembre 2019]. Disponible en: <http://www.ictea.com/cs/index.php?rp=/knowledgebase/8974/iQue-es-el-Sistema-Operativo-Android.html>.
- INFORMÁTICA, T., 2019. Los mejores administradores de archivos para Windows, Android y Mac. *Tecnología + Informática* [en línea]. [Consulta: 27 noviembre 2019]. Disponible en: <https://tecnologia-informatica.com/administradores-de-archivos/>.
- IONOS, D., 2019. Gestor de archivos Android. *IONOS Digitalguide* [en línea]. [Consulta: 14 noviembre 2019]. Disponible en: <https://www.ionos.es/digitalguide/servidores/herramientas/gestor-de-archivos-android/>.

- LARMAN, C., 2003. *UML y PATRONES. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. (Begoña, M., Trad.). S.l.: Madrid: Pearson Educación, SA (Original en inglés publicado en 2002).
- LARMAN, C. y APPLYING, U.M.L., 2004. *Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development , Chapter 6*. S.l.: Prentice Hall. October.
- LAZO Y, L., 2008. Lazo (lenguaje de programación) - Lasso (programming language) - qwe.wiki. [en línea]. Disponible en: [https://es.qwe.wiki/wiki/Lasso\\_\(programming\\_language\)](https://es.qwe.wiki/wiki/Lasso_(programming_language)).
- LICHOWSKI, M.C.B., Josefina Sastre Y. Gabriela, 2017. Interfaz hombre-máquina. ¿Manipulación o simplificación?: La metáfora de escritorio. *Interfaz hombre-máquina. ¿Manipulación o simplificación?* [en línea]. [Consulta: 23 enero 2020]. Disponible en: <http://interfaceshm.blogspot.com/2007/07/la-metfora-de-escritorio.html>.
- LUCA CRISAN, A., 2019. *A study of Kotlin 's: conciseness, safety and interoperability*. B.S. thesis. S.l.: Universitat Politècnica de Catalunya.
- OSNAYA, E., 2012. Administracion de Archivos. *Administracion de Archivos -eosnaya* [en línea]. [Consulta: 15 noviembre 2019]. Disponible en: <https://eosnaya.wordpress.com/1-conceptos-basicos-de-archivos/>.
- PRESSMAN, R.S., 2005a. *Software engineering: a practitioner's approach*. S.l.: Palgrave macmillan.
- PRESSMAN, R.S., 2010b. *Software engineering: a practitioner's approach*. S.l.: Palgrave macmillan.
- RAFFINO, M.E., 2019. Archivo en Informática: Concepto, Características y Formato. [en línea]. [Consulta: 15 noviembre 2019]. Disponible en: <https://concepto.de/archivo-informatico/>.
- RAMÍREZ, F.J., 2018. Metodologías para el desarrollo de software. [en línea], [Consulta: 27 noviembre 2019]. Disponible en: [https://www.academia.edu/9953322/Metodologias\\_para\\_el\\_desarrollo\\_de\\_software](https://www.academia.edu/9953322/Metodologias_para_el_desarrollo_de_software).
- ROGERS, J., 2014. File Naming Standards for Digital Collections. *B Sides* [en línea], Disponible en: <https://ir.uiowa.edu/bsides/35>.
- SOMMERVILLE, I., 2005. *Ingeniería del software*. S.l.: Pearson educación.
- SOMMERVILLE, I., 2011. SOMMERVILLE, I., 2011. Ingeniería del Software. 9na. Mexico: Addison Wesley. Pearson Education, Inc. ISBN 978-607-32-0603-7 - Buscar con Google. [en línea]. [Consulta: 28 enero 2020]. Disponible en: <https://www.google.com/search?source=hp&ei=IdIwXo70N82E5wK58rCQDw&q=SOMMERVILLE%2C+I.%2C+2011.+Ingenier%C3%ADa+del+Software.+9na.+Mexico%3A+Addison+Wesley.+Pearson+Education%2C+Inc.+ISBN+978-607-32-0603-7&oq=SOMMERVILLE%2C+I.%2C+2011.+Ingenier%C3%ADa+del+Software.+9na.+Mexico%3A+Addison+Wesley.+Pearson+Education%2C+Inc.+ISBN+978-607-32-0603-7>

7&gs\_l=psy-ab.3...9870.9870..10741...0.0..0.0.0.....0....2j1..gws-wiz.divSIJBALvo&ved=0ahUKEwjOiPiSx6fnAhVNwlkKHTk5DPiQ4dUDCAU&uact=5.

TECNOLOGICON, 2015. Definición De Directorio (Informática). *Tecnologicon* [en línea]. [Consulta: 23 enero 2020]. Disponible en: <https://tecnologicon.com/definicion-de-directorio-informatica/>.

TOMÁS, J., 2013. El Gran Libro de Android avanzado: 1: Amazon.es: JESUS TOMÁS GIRONÉS: Libros. [en línea]. [Consulta: 23 enero 2020]. Disponible en: <https://www.amazon.es/El-Gran-Libro-Android-avanzado/dp/8426720781>.

UCI, 2019. Nova Droid | Universidad de las Ciencias Informáticas. [en línea]. [Consulta: 27 noviembre 2019]. Disponible en: <https://www.uci.cu/investigacion-y-desarrollo/productos/nova/nova-droid>.

UNIVERSIDAD OBERTA DE CATALUNYA, 2017. Design Toolkit | Metáfora. [en línea]. [Consulta: 23 enero 2020]. Disponible en: <http://design-toolkit.reursos.uoc.edu/es/metafora/>.

UPTODOWN TECHNOLOGIES, S., 2018a. Amaze File Manager 3.3.2 para Android - Descargar. [en línea]. [Consulta: 23 enero 2020]. Disponible en: <https://amaze-file-manager.uptodown.com/android>.

UPTODOWN TECHNOLOGIES, S., 2018b. Simple File Manager 5.0.4 para Android - Descargar. [en línea]. [Consulta: 23 enero 2020]. Disponible en: <https://simple-mobile-tools-file-manager.uptodown.com/android>.

VANEGAS, C.A., 2012. Desarrollo de aplicaciones sobre Android - PDF Free Download. [en línea]. [Consulta: 23 enero 2020]. Disponible en: <https://docplayer.es/amp/616914-Desarrollo-de-aplicaciones-sobre-android.html>.

VERSIONES DE ANDROID, 2018. Todas las versiones del sistema operativo Android. *NorfiPC* [en línea]. [Consulta: 23 enero 2020]. Disponible en: <https://norfipc.com/celulares/todas-versiones-sistema-operativo-android.php>.

VISUAL PARADIGM FREQUENTLY ASKED QUESTIONS, 2017. Visual Paradigm Frequently Asked Questions. [en línea]. [Consulta: 27 noviembre 2019]. Disponible en: <https://www.visual-paradigm.com/support/faq.jsp>.

WHAT IS UNIFIED MODELING LANGUAGE (UML)?, 2017. What is Unified Modeling Language (UML)? [en línea]. [Consulta: 27 noviembre 2019]. Disponible en: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>.

## Anexos

### Anexo 1. Descripción de las HU

Tabla 10: Historia de usuario "Crear archivo".

Historia de Usuario	
<b>Número:</b> HU – 2	<b>Nombre del requisito:</b> Crear Archivo.
<b>Programador:</b> Alvaro Castro Pérez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 3
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 2
<b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario crear archivos. Se listan a continuación las funcionalidades requeridas: <ul style="list-style-type: none"><li>• Permitir al usuario navegar hasta la ubicación donde desea crear el archivo.</li><li>• Mostrar el botón de menú de opciones.</li><li>• Listar opciones del menú luego de hacer clic en el botón.</li><li>• Permitir al usuario elegir en el menú la opción "Crear Archivo".</li><li>• Permitir al usuario introducir el nombre del archivo.</li><li>• Mostrar la opción cancelar y crear.</li><li>• Listar nuevo archivo.</li></ul>	

Tabla 11: Historia de usuario "Cambiar el nombre de una carpeta o archivo."

Historia de Usuario	
<b>Número:</b> HU – 3	<b>Nombre del requisito:</b> Cambiar el nombre de una carpeta o archivo.
<b>Programador:</b> Alvaro Castro Pérez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 2
<b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario renombrar archivos o	

carpetas.

Se listan a continuación las funcionalidades requeridas:

- Permitir al usuario navegar hasta la ubicación donde desea renombrar el archivo o la carpeta.
- Permitir al usuario seleccionar una carpeta o un archivo.
- Mostrar el botón de menú de opciones.
- Listar opciones del menú luego de hacer clic en el botón.
- Permitir al usuario elegir en el menú la opción "Renombrar Archivo".
- Permitir al usuario introducir el nombre del archivo.
- Mostrar la opción cancelar y crear.
- Listar el archivo con el nuevo nombre.

Tabla 12: Historia de usuario "Seleccionar carpetas."

Historia de Usuario	
<b>Número:</b> HU – 4	<b>Nombre del requisito:</b> Seleccionar Carpetas.
<b>Programador:</b> Alvaro Castro Pérez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 1
<b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario seleccionar carpetas. Se listan a continuación las funcionalidades requeridas: <ul style="list-style-type: none"><li>• Permitir al usuario navegar hasta la ubicación donde desea seleccionar la carpeta.</li><li>• Permitir al usuario seleccionar una carpeta o un archivo.</li></ul>	

Tabla 13: Historia de usuario "Mover carpetas o archivos."

<b>Número:</b> HU – 5	<b>Nombre del requisito:</b> Mover carpetas o archivos.
<b>Programador:</b> Alvaro Castro Pérez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 4
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 2
<p><b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario mover carpetas o archivos. Se listan a continuación las funcionalidades requeridas:</p> <ul style="list-style-type: none"> <li>• Permitir al usuario navegar hasta la ubicación donde desea mover la carpeta o el archivo.</li> <li>• Permitir al usuario seleccionar una carpeta o un archivo.</li> <li>• Mostrar el botón de menú de opciones.</li> <li>• Listar opciones del menú luego de hacer clic en el botón.</li> <li>• Permitir al usuario elegir en el menú la opción “Mover Archivo”.</li> <li>• Permitir al usuario navegar hasta la ubicación donde desea pegar el archivo o la carpeta que va a ser movido.</li> <li>• Permitir al usuario seleccionar dentro de la lista de opciones del menú la opción “Pegar”</li> <li>• Listar el archivo en la nueva ubicación.</li> </ul>	

<b>Número:</b> HU – 8	<b>Nombre del requisito:</b> Buscar carpeta o archivo.
<b>Programador:</b> Alvaro Castro Pérez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 1
<p><b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario buscar un archivo o una carpeta.</p> <p>Se listan a continuación las funcionalidades requeridas:</p> <ul style="list-style-type: none"> <li>• Permitir al usuario navegar hasta la ubicación donde desea buscar el archivo o la carpeta.</li> <li>• Permitir al usuario seleccionar en la barra superior el botón de “Buscar” que se identifica por una lupa.</li> <li>• Permitir al usuario introducir el nombre del archivo o carpeta que desea buscar.</li> <li>• Mostrar al usuario los elementos asociados al nombre introducido en la búsqueda.</li> </ul>	

Tabla 15: Historia de usuario “Comprimir carpetas o archivos.”

<b>Número:</b> HU – 9	<b>Nombre del requisito:</b> Comprimir carpetas o archivos.
<b>Programador:</b> Alvaro Castro Pérez	<b>Iteración Asignada:</b> 2
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 3
<p><b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario renombrar archivos o carpetas.</p> <p>Se listan a continuación las funcionalidades requeridas:</p> <ul style="list-style-type: none"> <li>• Permitir al usuario navegar hasta la ubicación donde desea comprimir el archivo o la carpeta.</li> <li>• Permitir al usuario seleccionar una carpeta o un archivo.</li> <li>• Mostrar el botón de menú de opciones.</li> <li>• Listar opciones del menú luego de hacer clic en el botón.</li> <li>• Permitir al usuario elegir en el menú la opción “Comprimir Archivo”.</li> <li>• Listar el archivo.zip en la misma ubicación del archivo original.</li> </ul>	

Tabla 16: Historia de usuario "Extraer carpetas o archivos."

<b>Número:</b> HU – 10	<b>Nombre del requisito:</b> Extraer carpetas o archivos.
<b>Programador:</b> Alvaro Castro Pérez	<b>Iteración Asignada:</b> 2
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 4
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 4
<p><b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario extraer elementos comprimidos.</p> <p>Se listan a continuación las funcionalidades requeridas:</p> <ul style="list-style-type: none"> <li>• Permitir al usuario navegar hasta la ubicación donde desea descomprimir el elemento.</li> <li>• Permitir al usuario seleccionar el elemento comprimido.</li> <li>• Mostrar el botón de menú de opciones.</li> <li>• Listar opciones del menú luego de hacer clic en el botón.</li> <li>• Permitir al usuario elegir en el menú la opción "Descomprimir".</li> <li>• Listar en la misma ubicación el elemento ya descomprimido.</li> </ul>	

Tabla 17: Historia de usuario "Compartir carpetas o archivos."

<b>Número:</b> HU – 12	<b>Nombre del requisito:</b> Compartir carpetas o archivos.
<b>Programador:</b> Alvaro Castro Pérez	<b>Iteración Asignada:</b> 2
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 4
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 3
<p><b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario compartir archivos o carpetas.</p> <p>Se listan a continuación las funcionalidades requeridas:</p> <ul style="list-style-type: none"> <li>• Permitir al usuario navegar hasta la ubicación donde desea compartir una carpeta o un</li> </ul>	

archivo.

- Permitir al usuario seleccionar una carpeta o un archivo.
- Mostrar el botón de menú de opciones.
- Listar opciones del menú luego de hacer clic en el botón.
- Permitir al usuario elegir en el menú la opción “Compartir”.
- Mostrar al usuario las distintas aplicaciones por la que puede compartir el archivo.
- Abrir la aplicación deseada por la que se va a compartir el archivo.
- Mostrar botón para enviar el elemento.
- Enviar el elemento seleccionado.

Tabla 18: Historia de usuario “Ordenar carpetas o archivos.”

Historia de Usuario	
<b>Número:</b> HU – 13	<b>Nombre del requisito:</b> Ordenar carpetas y archivos.
<b>Programador:</b> Alvaro Castro Pérez	<b>Iteración Asignada:</b> 2
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 2
<b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario ordenar archivos y carpeta. Se listan a continuación las funcionalidades requeridas: <ul style="list-style-type: none"><li>• Permitir al usuario navegar hasta la ubicación donde desea ordenar los archivos y carpetas.</li><li>• Mostrar el botón de menú de opciones.</li><li>• Listar opciones del menú luego de hacer clic en el botón.</li><li>• Permitir al usuario elegir en el menú la opción “Ordenar”.</li><li>• Mostrar al usuario los métodos de ordenación.</li><li>• Permitir al usuario seleccionar un método por el cual va a organizar los archivos y carpetas.</li><li>• Listar los archivos y carpetas ordenados por el método de ordenación seleccionado.</li></ul>	

Tabla 19: Historia de usuario "Mostrar propiedades de una carpeta o archivo."

<b>Número:</b> HU – 14	<b>Nombre del requisito:</b> Mostrar propiedades de una carpeta o archivo.
<b>Programador:</b> Alvaro Castro Pérez	<b>Iteración Asignada:</b> 2
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 1
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 1
<p><b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario ver las propiedades de los archivos o las carpetas.</p> <p>Se listan a continuación las funcionalidades requeridas:</p> <ul style="list-style-type: none"> <li>• Permitir al usuario navegar hasta la ubicación donde desea ver las propiedades del archivo o la carpeta.</li> <li>• Permitir al usuario seleccionar una carpeta o un archivo.</li> <li>• Mostrar el botón de menú de opciones.</li> <li>• Listar opciones del menú luego de hacer clic en el botón.</li> <li>• Permitir al usuario elegir en el menú la opción "Detalles"</li> <li>• Mostrar detalles del archivo o carpeta seleccionada.</li> </ul>	

Tabla 20: Historia de usuario "Cambiar vista."

<b>Número:</b> HU – 15	<b>Nombre del requisito:</b> Cambiar vista.
<b>Programador:</b> Alvaro Castro Pérez	<b>Iteración Asignada:</b> 2
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 1
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 2
<p><b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario cambiar la vista de la aplicación.</p> <p>Se listan a continuación las funcionalidades requeridas:</p>	

- Permitir al usuario seleccionar el botón para cambiar la vista de la aplicación.
- Mostrar las opciones de vistas de la aplicación.
- Permitir al usuario seleccionar una de las opciones.
- Mostrar la vista de aplicación seleccionada.

Tabla 21: Historia de usuario "Abrir carpetas o archivos."

<b>Número:</b> HU – 17	<b>Nombre del requisito:</b> Abrir carpetas o archivos.
<b>Programador:</b> Alvaro Castro Pérez	<b>Iteración Asignada:</b> 2
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 1
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 1
<p><b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario abrir archivos o carpetas. Se listan a continuación las funcionalidades requeridas:</p> <ul style="list-style-type: none"> <li>• Permitir al usuario navegar hasta la ubicación donde desea abrir la carpeta o un archivo.</li> <li>• Permitir al usuario hacer doble clic sobre la carpeta o archivo que desea abrir.</li> <li>• Abrir el archivo o la carpeta seleccionada.</li> </ul>	

Tabla 22: Historia de usuario "Listar opciones del menú."

<b>Número:</b> HU – 18	<b>Nombre del requisito:</b> Listar opciones del menú.
<b>Programador:</b> Alvaro Castro Pérez	<b>Iteración Asignada:</b> 2
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 1
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 1
<p><b>Descripción:</b> Luego de ejecutar la aplicación esta permite al usuario ordenar archivos y carpeta. Se listan a continuación las funcionalidades requeridas:</p> <ul style="list-style-type: none"> <li>• Mostrar el botón de menú de opciones.</li> <li>• Permitir al usuario hacer clic sobre el botón de menú de opciones.</li> </ul>	

<ul style="list-style-type: none"> <li>• Mostrar opciones del menú.</li> </ul>
Empty content area

Tabla 14:  
Historia  
de  
usuario  
"Buscar

carpeta o archivo."