

# Fortress Of Comics

**Marcos Blanco Blanco**

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

**Francesc D'Assís Giralt Queralt**

06/06/2018

© Marcos Blanco Blanco

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	Fortress Of Comics
<b>Nombre del autor:</b>	Marcos Blanco Blanco
<b>Nombre del consultor:</b>	Francesc D'Assís Giralt Queralt
<b>Fecha de entrega (mm/aaaa):</b>	06/2018
<b>Titulación:</b>	<i>Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles</i>
<b>Resumen del Trabajo (máximo 250 palabras):</b>	
<p>Este proyecto consiste en el diseño y desarrollo de Fortress Of Comics, una aplicación móvil para smartphones Android.</p> <p>Esta aplicación surge a raíz del gran crecimiento que se produce cada año en el mundo de los cómics. El número de nuevos cómics que son creados, editados y puestos a la venta va en aumento. Este aumento, y el hecho de que los temas cada vez son más variados para conseguir atraer un mayor público, procura que con el tiempo se encuentren más lectores. En multitud de ocasiones pasamos de hablar de lectores a coleccionistas cuyo hobby es el de aumentar su colección añadiendo nuevos cómics que obtienen un gran valor para ellos. Al mismo tiempo la pasión que tienen por estos produce dos factores a tener en cuenta. Por un lado, al conectar con la historia tienen la posibilidad de comentar y discutir aspectos de la historia con otros lectores dando una valoración tanto de forma verbal como cualitativa. Por otro lado, al empezar a comprarlos se interesan por cuales son las mejores tiendas, no tanto por precio sino por stock de productos, distribución y ambiente.</p> <p>Con estos datos en mente se intenta generar una aplicación para lectores de cómics, ya sean recién iniciados o expertos en este mundo, que permita gestionar sus colecciones de cómics, compartir opiniones y revelar nuevas tiendas para conocer de primera mano los establecimientos más cercanos o con mejor valoración.</p>	

**Abstract (in English, 250 words or less):**

This project consists on the design and develops of Fortress Of Comics, a mobile application for Android smartphones.

This application comes from the big increase that occurs every year in the world of comics. The number of new comics that are created, edited and sold is increasing. With this, and the fact that topics are more varied to attract more audience, get a bigger number of readers. On many occasions' simple readers become collectors whose hobby is to increase their collection by adding new comics that provides great importance for them. The passion that supposes for them brings two factors to consider. On the one hand, when a reader connects with the story that is reading gets the possibility of comment and talk over aspects of the story with other readers by giving personal appreciation. On the other hand, when a reader begins to buy comics one of the main interests is to know which are the better comic stores, mainly for its stock and distribution.

With these aspects in mind, the idea is to generate an application for comic readers, both beginners and experts, that allows to manage their collection, share opinions and reveal new comic shops to know personally what establishments are near or has better rating.

**Palabras clave (entre 4 y 8):**

Aplicación, cómic, compartir

## Índice

1. Introducción.....	1
1.1 Contexto y justificación del trabajo .....	1
1.2 Objetivos del trabajo .....	4
1.3 Enfoque y método elegido .....	5
1.4 Planificación del trabajo .....	7
1.5 Breve resumen de productos obtenidos.....	8
2. Análisis.....	10
2.1 Definición de usuarios.....	10
2.2 Definición de requisitos.....	11
2.3 Definición de casos de uso .....	13
3. Diseño .....	20
3.1 Árbol de navegación .....	20
3.2 Diseño de arquitectura.....	21
3.3 Diseño de la estructuración de datos.....	22
3.4 Diseño de las pantallas: wireframes y prototipo.....	25
4. Implementación .....	33
4.1 Estructuración del proyecto .....	33
4.2 Decisiones de implementación .....	33
5. Pruebas .....	38
5.1 Pruebas de la pantalla de inicio de sesión.....	38
5.2 Pruebas de la pantalla de registro .....	39
5.3 Pruebas de la pantalla de editar el perfil.....	40
5.4 Pruebas de la pantalla de novedades.....	40
5.5 Pruebas de la pantalla de búsquedas.....	40
5.6 Pruebas de la pantalla de nueva tienda.....	41
5.7 Prueba de la pantalla de nuevo comentario.....	42
6. Conclusiones.....	44
6.1 Aprendizaje.....	44
6.2 Planificación.....	44
6.3 Resultado final .....	44
6.4 Mejoras de futuro .....	44
Bibliografía .....	46

## **Lista de figuras**

- Figura 1: Capturas de pantalla de la aplicación CLZ
- Figura 2: Capturas de pantalla de la aplicación Comic Books Collector
- Figura 3: Capturas de pantalla de la aplicación Manga List
- Figura 4: Diagrama de Gantt de las tareas de diseño
- Figura 5: Diagrama de Gantt de las tareas de implementación
- Figura 6: Diagrama de Gantt de las tareas de la entrega final
- Figura 7: Ficha de usuario 1
- Figura 8: Ficha de usuario 2
- Figura 9: Caso de uso del registro del usuario en la aplicación
- Figura 10: Caso de uso del inicio de sesión del usuario en la aplicación
- Figura 11: Caso de uso de la búsqueda de un cómic
- Figura 12: Caso de uso de la visualización de la colección del usuario
- Figura 13: Caso de uso de la visualización de los detalles de un cómic
- Figura 14: Caso de uso de la visualización de los detalles de una tienda
- Figura 15: Caso de uso del registro de una nueva tienda
- Figura 16: Caso de uso de la visualización de los comentarios de un cómic
- Figura 17: Caso de uso del registro de un nuevo comentario
- Figura 18: Caso de uso de la visualización de la valoración de un cómic
- Figura 19: Caso de uso del registro de una nueva valoración
- Figura 20: Árbol de navegación de la aplicación
- Figura 21: Tipos de arquitectura
- Figura 22: Estructura del Model View ViewModel (MVVM)
- Figura 23: Estructura de los usuarios en Firebase
- Figura 24: Estructura de las novedades en Firebase
- Figura 25: Estructura de los cómics en Firebase
- Figura 26: Estructura de las tiendas en Firebase
- Figura 27: Estructura de los comentarios en Firebase
- Figura 28: Estructura de las valoraciones en Firebase
- Figura 29: Diagrama de las entidades de Realm

Figura 30: Wireframe y prototipo del diseño de la pantalla de *login*

Figura 31: Wireframe y prototipo del diseño de la pantalla de registro

Figura 32: Wireframe y prototipo del diseño del menú lateral de navegación

Figura 33: Wireframe y prototipo del diseño de la pantalla de inicio

Figura 34: Wireframe y prototipo del diseño de la pantalla de Mi colección

Figura 35: Wireframe y prototipo del diseño de la pantalla de búsqueda

Figura 36: Wireframe y prototipo del diseño de la pantalla de detalles del cómic

Figura 37: Wireframe y prototipo del diseño de la pantalla de comentarios

Figura 38: Wireframe y prototipo del diseño de la pantalla de nuevo comentario

Figura 39: Wireframe y prototipo del diseño de la pantalla de valorar

Figura 40: Wireframe y prototipo del diseño de la pantalla de tiendas

Figura 41: Wireframe y prototipo del diseño de la pantalla de nueva tienda

Figura 42: Wireframe y prototipo del diseño de la pantalla de detalles de tienda

Figura 43: Capturas de la pantalla de inicio de sesión y de registro

Figura 44: Capturas de la pantalla de novedades y del menú lateral

Figura 45: Capturas de la pantalla de editar el perfil y de búsqueda

Figura 46: Capturas de la pantalla de mi colección y de detalles del cómic

Figura 47: Capturas de la pantalla de comentarios del cómic y de nuevo comentario

Figura 48: Capturas de la pantalla de valoraciones del cómic y de tiendas de cómics

Figura 49: Capturas de la pantalla de nueva tienda y localización de la tienda

Figura 50: Capturas de la pantalla de comentarios de y de valoraciones de la tienda

## **Lista de tablas**

Tabla 1: Requisito funcional 1  
Tabla 2: Requisito funcional 2  
Tabla 3: Requisito funcional 3  
Tabla 4: Requisito funcional 4  
Tabla 5: Requisito funcional 5  
Tabla 6: Requisito funcional 6  
Tabla 7: Requisito funcional 7  
Tabla 8: Requisito funcional 8  
Tabla 9: Requisito funcional 9  
Tabla 10: Requisito no funcional 1  
Tabla 11: Requisito no funcional 2  
Tabla 12: Requisito no funcional 3  
Tabla 13: Requisito no funcional 4  
Tabla 14: Prueba 1 de la pantalla de inicio de sesión  
Tabla 15: Prueba 2 de la pantalla de inicio de sesión  
Tabla 16: Prueba 3 de la pantalla de inicio de sesión  
Tabla 17: Prueba 1 de la pantalla de registro  
Tabla 18: Prueba 2 de la pantalla de registro  
Tabla 19: Prueba 3 de la pantalla de registro  
Tabla 20: Prueba 4 de la pantalla de registro  
Tabla 21: Prueba 1 de la pantalla de editar el perfil  
Tabla 22: Prueba 2 de la pantalla de editar el perfil  
Tabla 23: Prueba 1 de la pantalla de novedades  
Tabla 24: Prueba 1 de la pantalla de búsqueda  
Tabla 25: Prueba 2 de la pantalla de búsqueda  
Tabla 26: Prueba 3 de la pantalla de búsqueda  
Tabla 27: Prueba 4 de la pantalla de búsqueda  
Tabla 28: Prueba 1 de la pantalla de nueva tienda  
Tabla 29: Prueba 2 de la pantalla de nueva tienda  
Tabla 30: Prueba 3 de la pantalla de nueva tienda



Tabla 31: Prueba 4 de la pantalla de nueva tienda

Tabla 32: Prueba 1 de la pantalla de nuevo comentario

Tabla 33: Prueba 2 de la pantalla de nuevo comentario

# 1. Introducción

## 1.1 Contexto y justificación del trabajo

El mundo de los comics crece y aumenta constantemente con el paso de los años. La cantidad de comics que se realizan y posteriormente comercializan aumenta a un ritmo constante proporcionalmente al número de personas que cada vez más se interesan en introducirse en este mundo.

Actualmente existe una gran mayoría de lectores de comics, muchos de ellos coleccionistas no solo interesados en la lectura de estos sino también en su obtención como un objeto destacado e importante. En muchas ocasiones, al empezar a interesarse, leer y coleccionar una serie determinada de cómics se produce en el lector la necesidad de informarse más sobre esa serie, donde poder conseguir nuevas entregas y, si hay un gran interés, compartir opiniones con otros lectores sobre la historia.

Precisamente por este incremento de lectores, que es un movimiento que cada vez va a más, cada vez más es un buen planteamiento realizar una aplicación no solo para lectores habituales de cómics sino para aquellos que empiezan a leerlos y requieren informarse de lo que pueden encontrar, a que tiendas pueden acceder...

Eso es lo que se pretende conseguir con la aplicación ***Fortress Of Comics***.

### Otras aplicaciones similares

Se han localizado varias aplicaciones similares a la que se pretende realizar, pero se han destacado *CLZ*, *Comic Books collector* o *Manga List*. A continuación se describirán brevemente estas aplicaciones.

La primera aplicación es *CLZ*:

<https://play.google.com/store/apps/details?id=com.collectorz.javamobile.android.comics>

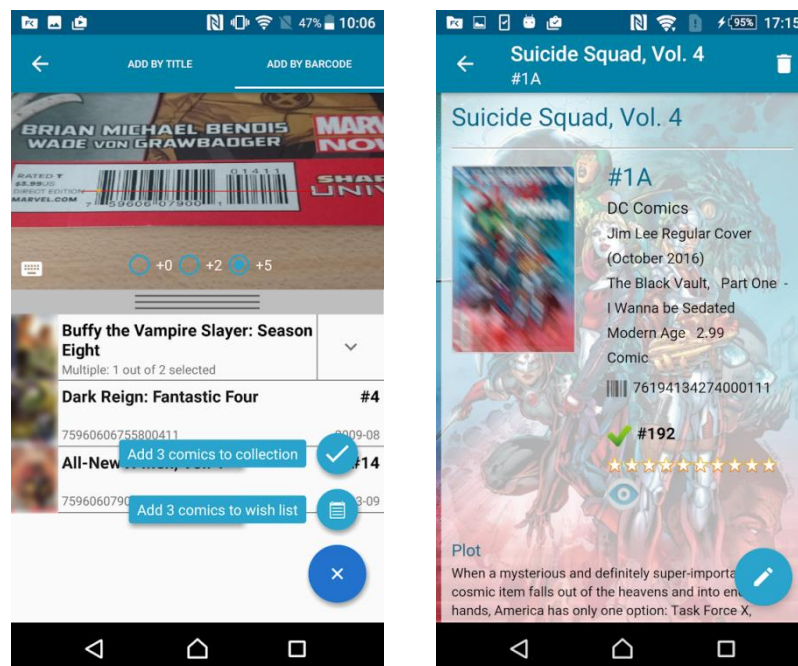


Fig. 1 – Capturas de pantalla de la aplicación CLZ

Sin duda esta app es la más completa de las tres destacadas. Permite la búsqueda de cómics tanto escaneando el código *isbn* como de forma manual, mostrando los detalles del cómic en cuestión. Además, cumple con la base que se pretende seguir que es la de gestionar la colección de cómics del usuario. No obstante, solo una parte de sus funcionalidades son gratuitas.

La segunda aplicación es *Comic Books Collector*.

<https://play.google.com/store/apps/details?id=com.sweetlime.cbcollector>

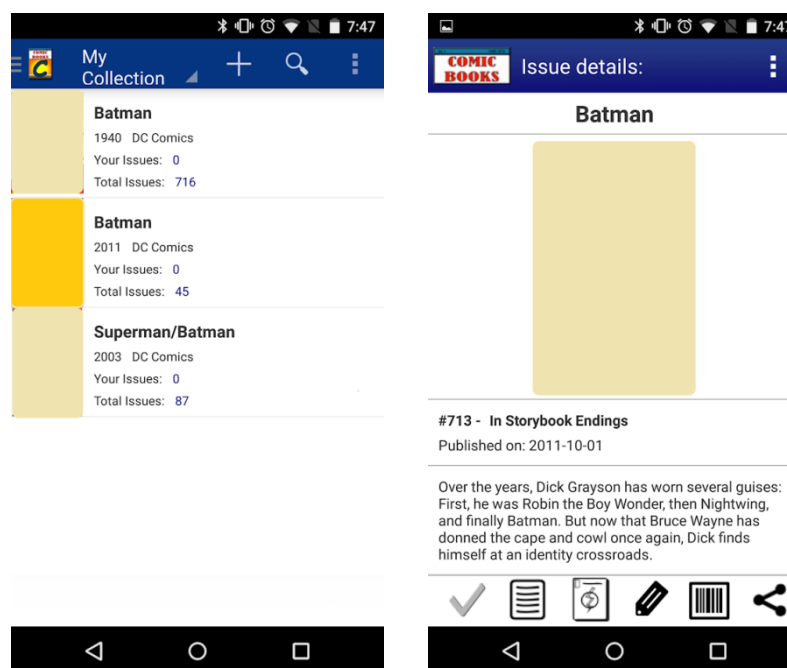


Fig. 2 – Capturas de pantalla de la aplicación Comic Books Collector

Como la anterior posee la característica de centrarse en la gestión de la colección de cómics del usuario con similitudes pero con diferencias como la búsqueda, que únicamente se puede realizar de forma manual. La principal propiedad a tener en cuenta de esta aplicación es que el usuario ha de tener una cuenta registrada en el servicio web de *ComicVine* para poder acceder a la api que proporciona los datos de cada cómic.

La tercera aplicación es *Manga List*:

<https://play.google.com/store/apps/details?id=be.benoit.sneyers.mangalist>

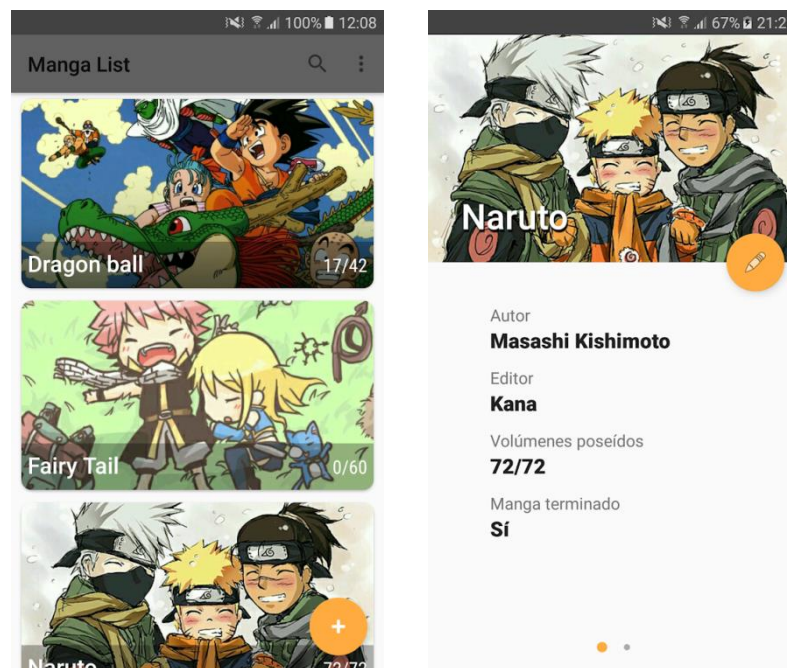


Fig. 3 – Capturas de pantalla de la aplicación Manga List

Esta aplicación se centra exclusivamente en cómics manga pero tiene una principal debilidad frente a las dos anteriores, no dispone de un servicio de datos de donde obtener los detalles de los cómics. En esta aplicación el usuario ha de introducir manualmente datos de una serie de cómics concreta para añadirla a su colección. Da la opción de especificar cuantos volúmenes de la serie posee el usuario.

Las tres son aplicaciones que representan una buena fuente de información de cómics permitiendo realizar búsquedas y recibir listados de cómics con información sobre ellos. No obstante, solo aportan hasta ese punto. No todas implementan un acercamiento a la experiencia del usuario de compartir y valorar obteniendo no solo la propia información del autor o la editorial, que puede obtener en cualquier otra parte, sino también una información más cercana de un lector con el que te puedes identificar. Estas aplicaciones no fundamentan el conocimiento de lo más importante que interesa para obtener un cómic que es conocer los

posibles establecimientos donde conseguirlo fomentando que lectores menos habituales puedan conocer nuevas tiendas donde empezar a comprar y motivarse a leer. La gran mayoría de veces, los propios usuarios son la mayor fuente información para este tipo de datos que se puede encontrar dado que son los que viven la experiencia en directo. Así que un apartado donde los usuarios registran los establecimientos en los que han estado para informar a los demás sería de gran ayuda para el resto de lectores.

Algo interesante es que *CLZ* implementa el escaneo de código de barras para obtener el iban del cómic y buscarlo en la base de datos. Es un punto interesante pensando en la situación de una persona que quiere comprar algo nuevo pero no sabe que esperar de ello. Escaneando el código en la propia tienda y obteniendo información extra de ese producto puede obtener una mejor perspectiva para seleccionar. No obstante, esta aplicación requiere de un pago para obtener su completa funcionalidad.

## 1.2 Objetivos del trabajo

El proyecto a desarrollar consistirá en la realización de una aplicación para dispositivos móviles que permitirá a los usuarios lectores de cómics gestionar su colección personal de cómics.

La aplicación dará al usuario la opción de buscar y seleccionar un cómic escaneando el isbn en el código de barras o escribiendo directamente para añadirlo a su colección personal generando un registro que recopilará su colección personal. A medida que el usuario obtiene su lista de cómics también podrá compartirla con otros usuarios. A parte, otra ventaja que se obtiene con esta funcionalidad es la de buscar esta información en una tienda escaneando el cómic directamente sin necesidad de buscarlo por internet en una página web concreta.

Al entrar a la página de detalles de un cómic concreto se podrán escribir opiniones y dar valoraciones para que otros usuarios puedan interesarse en su compra o dar una réplica a otra opinión.

Por último, la aplicación dispondrá de una sección donde se puedan introducir, compartir y visualizar datos de tiendas de cómics para que otros usuarios puedan conocer su ubicación y algún dato de la tienda. Está sección dispondrá de una página para introducir los datos de la tienda, una página para ver el listado de tiendas y una página para ver los detalles con una mapa de localización de la tienda.

Los objetivos principales que conllevará este proyecto son:

- Permitir al usuario identificarse a través de un sistema de *login*.
- Incorporar la funcionalidad de escaneo y gestionarla para recibir el dato escaneado.

- Confeccionar un perfil personal donde el usuario pueda disponer de su colección.
- Permitir a los usuarios compartir sus colecciones, opiniones y valoraciones de los cómics.
- Disponer de un apartado que permita el conocimiento de las tiendas habituales visitadas por los usuarios.

### 1.3 Enfoque y método elegido

#### Elección de la plataforma

Después de revisar algunas aplicaciones enfocadas al mismo tema, el plan es realizar una aplicación nativa de *Android* a la que se pueda acceder inmediatamente desde el dispositivo, que permita alguna funcionalidad como la de escaneo, almacenamiento en de datos en el dispositivo...

La utilización de desarrollo nativo permite el uso de funciones avanzadas de cada plataforma, sacan provecho de los procesadores gráficos y experimentan cada funcionalidad a un nivel al que un *plugin* de una aplicación híbrida no puede acercarse. De hecho, al no necesitar *plugins* pueden utilizar las últimas características de la plataforma.

Las plataformas nativas aprovechan funciones de componentes de la UI permitiendo a algunos elementos realizar transiciones complejas o animaciones consumiendo mucha menos memoria. Se pueden crear tantos hilos de ejecución como deseemos y podemos utilizar servicios para realizar trabajo en segundo plano. Proporcionan una mayor seguridad y mejoran la duración de la batería. El código está optimizado para la arquitectura, hace un uso más adecuado de los distintos núcleos de CPU/GPU y necesita menos capas de software para ejecutarse.

A parte, las aplicaciones nativas se desarrollan en entornos de desarrollo que disponen de herramientas de diseño gráfico que permiten más facilidad en la creación de la interfaz de usuario.

Finalmente, se ha decidido desarrollar la aplicación con *Android* debido a la falta de recursos y conocimientos necesarios para el desarrollo de una aplicación con *iOS*.

#### Servicio web de datos

En estos momentos no se dispone de una api completa (existen *apis* centradas en cómics de *Marvel* o manga en general, pero no todos en general) y gratuita donde se pueda obtener la información de cada uno de los cómics. De manera que se utilizará *Firebase* como servicio de base de datos externa generando un archivo *json* que guardará todos los datos generados por la aplicación así como los datos de los cómics. Obviamente, para la realización de este proyecto se pretende generar y comprobar la funcionalidad de la aplicación móvil así que no se generará

ni dispondrá de una base de datos de cómics completa, serán añadidos algunos cómics para las pruebas pertinentes y en el futuro se intentará ir completando.

### Estrategia

Después de revisar las aplicaciones ya existentes comentadas anteriormente se ha llegado a la conclusión de desarrollar un producto nuevo pero tomando como ejemplo el enfoque que se emplea en ellas. alguna de las aplicaciones ya utiliza un sistema de escaneo para buscar los cómics, que es un elemento que se desea incorporar en la aplicación a desarrollar.

Las diferentes metodologías de desarrollo contempladas han sido el método en cascada, desarrollo rápido de aplicaciones, desarrollo ágil y mobile-d.

El método en cascada se aplica normalmente en proyectos donde los requisitos están fijados y no van a cambiar durante el ciclo de vida del desarrollo. Como el propio nombre indica esta metodología se guía por realizar las tareas a realizar secuencialmente, dedicarle un periodo de tiempo a una tarea y no empezar la siguiente hasta tener está finalizada. Se pone especial atención a la planificación, el tiempo y las fechas límite.

El desarrollo rápido de aplicaciones pretende la obtención de un prototipo funcional en un corto periodo de tiempo para, en el futuro, ir mejorándola incluyendo más funcionalidades. Prioriza la implementación sobre la planificación.

La metodología de desarrollo ágil tiene en cuenta la alta frecuencia de cambios en el entorno de desarrollo tales como los cambios en las distintas plataformas. Se rige por realizar las aplicaciones en periodos cortos de desarrollo.

El objetivo de la metodología mobile-d es el de conseguir ciclos de desarrollo muy rápidos y se compone de distintas fases: exploración, inicialización, fase de producto, fase de estabilización y fase de pruebas.

La exploración se centra en la planificación, se definen el alcance del proyecto y las funcionalidades que se quieren realizar. La inicialización consiste en la preparación e identificación de los recursos necesarios. En la fase de producto se repiten iterativamente los pasos de planificación, iniciación y entrega con cada funcionalidad a realizar. Durante la fase de estabilización se unifican todas las partes realizadas durante la fase de producto para obtener un producto completo y comprobar su funcionamiento conjunto. Por último, la fase de pruebas tiene como objetivo disponer de una versión estable y funcional de la aplicación según los requisitos establecidos.

La metodología escogida ha sido la del método en cascada que consiste en realizar el desarrollo paso a paso, dividiéndolo en distintas fases, pero con una pequeña énfasis en las fases de la metodología mobile-d. Por lo tanto, la forma de actuar será dedicarse a fondo en una tarea hasta finalizarla y pasar a la siguiente, asegurándose que si la tarea previa se ha resuelto correctamente la siguiente se podrá realizar en óptimas condiciones. Por otra parte, se procederá a realizar cada tarea o funcionalidad planificándola y realizándole las pruebas pertinentes para finalmente realizar las pruebas finales de una versión lo más completa posible de todas las funcionalidades realizadas en conjunto.

#### 1.4 Planificación del trabajo

##### Recursos de hardware

- **Ordenador portátil ASUS** como dispositivo para realizar el desarrollo de la aplicación
- **Smartphone Samsung Galaxy Grand Prime** para la realización de pruebas

##### Recursos de software

- **Android Studio 3.0** como entorno de desarrollo.
- **Firebase** como servidor con base de datos externa a través de archivo json.
- **Notepad++** como editor de código para poder realizar modificaciones en archivos json.
- **JustInMind** como herramienta de prototipado.
- **Microsoft Office 2010** para la realización de la documentación y la presentación.

##### Tareas a realizar

- Diseño:

En esta tarea se realizará un análisis de la aplicación donde se confeccionará el número de pantallas y la navegación entre pantallas. Posteriormente se diseñarán unos *wireframes* de las pantallas y, finalmente, se realizará un prototipo inicial de la aplicación con una herramienta de prototipado.

- Implementación:

A lo largo de esta tarea se desarrollará con el entorno de desarrollo elegido el código necesario para el correcto funcionamiento de la aplicación, obteniendo las vistas y la lógica de está. También se realizarán algunas pruebas.

- Entrega final:



Finalmente se acabará de redactar la documentación de la realización del proyecto así como la presentación de este y de un manual de usuario para la aplicación.

## Planificación temporal

El tiempo que se dedicará a la realización diaria de este proyecto será de 2 horas en días laborales y de 5 horas en días no laborales, sumando un total de 20 horas semanales.

La planificación de las subtareas de cada entrega será la siguiente:

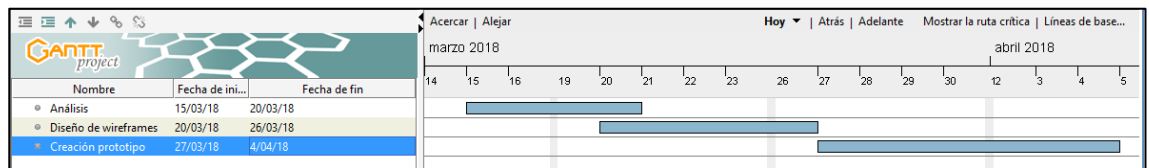


Fig. 4 – Diagrama de Gantt de las tareas de diseño

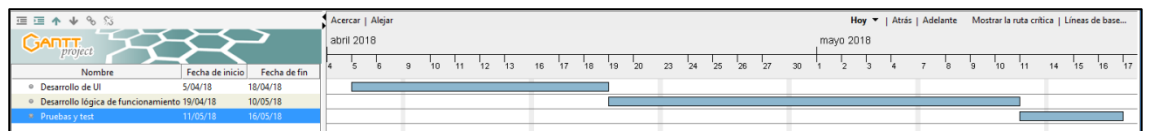


Fig. 5 – Diagrama de Gantt de las tareas de implementación

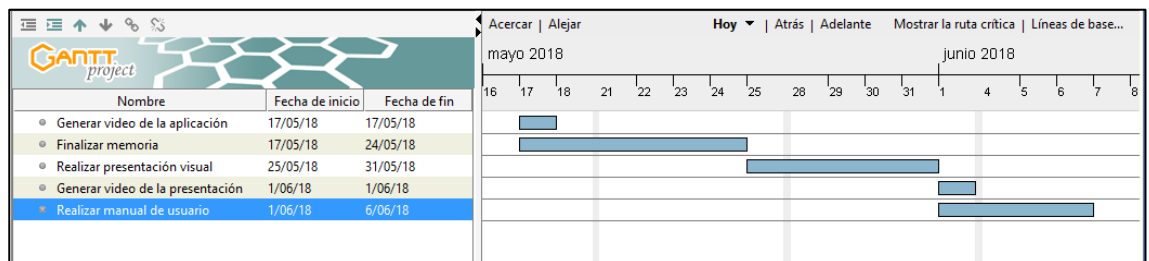


Fig. 6 – Diagrama de Gantt de las tareas de la entrega final

Esta planificación es una previsión inicial del tiempo a invertir en cada tarea, pero puede llegar a variar.

## 1.5 Breve sumario de productos obtenidos

Al finalizar el periodo de realización de este TFM se obtendrán los siguientes productos:

- Memoria:
  - Documento que recogerá la planificación del trabajo, descripción detallada de los procesos de diseño e implementación de la aplicaciones y unas conclusiones
- Presentación:

- Documento que contendrá la ayuda visual para realizar la presentación del proyecto.
- Video de la presentación:
  - Video que mostrará el documento de la presentación acompañado de audio con los comentarios y explicaciones del proyecto.
- Código fuente de la aplicación:
  - Archivos generados en el proyecto que contienen todo el código de la aplicación
- Archivo ejecutable de la aplicación:
  - Archivo que se ejecuta en el dispositivo móvil para instalar la aplicación.
- Manual de usuario de la aplicación:
  - Documento que recopilará una descripción de las principales funcionalidades de la aplicación para su correcto uso.
- Video del funcionamiento de la aplicación:
  - Video donde se demostrará el funcionamiento de la aplicación al completo.

## 2. Análisis

### 2.1 Definición de usuarios

Tal y como se ha especificado en el capítulo anterior los principales usuarios a los que va dirigida la aplicación que se pretende realizar en este proyecto son lectores de cómics. No necesariamente han de ser coleccionistas expertos en cómics, puede ser una aplicación para esa persona que está interesada en introducirse por primera vez en este mundo o que no ha leído demasiados y aprovecha el uso de la aplicación para buscar en la base de datos los cómics que encuentra en la tienda e informarse de que trata y que valoración le dan otros lectores. A parte, si no conoce una tienda cercana a la que ir podrá ver que le ofrece la aplicación a través de las tiendas que hayan compartido otros usuarios.

A continuación se definirá el perfil de dos posibles usuarios describiendo sus características y necesidades referidas a la temática de la aplicación.

#### Usuario 1



Fig. 7: Ficha de usuario 1

## Usuario 2



Fig. 8: Ficha de usuario 2

## 2.2 Definición de requisitos

En este apartado se definirán los distintos requisitos que presenta la realización de esta aplicación. Los casos se dividirán en requisitos funcionales y no funcionales.

### Requisitos funcionales

Requisito funcional 1	
Nombre	Registrarse
Descripción	El usuario debe poder registrarse para acceder al contenido de la aplicación.

Tabla 1: Requisito funcional 1

Requisito funcional 2	
Nombre	Iniciar/cerrar sesión
Descripción	El usuario debe poder iniciar sesión a través de un <i>login</i> y cerrar la sesión cuando decida finalizar su uso.

Tabla 2: Requisito funcional 2

<b>Requisito funcional 3</b>	
Nombre	Menú de navegación lateral
Descripción	La aplicación debe disponer de un menú lateral desplegable que disponga de las principales opciones de navegación.

*Tabla 3: Requisito funcional 3*

<b>Requisito funcional 4</b>	
Nombre	Escanear/Buscar un cómic
Descripción	La aplicación debe disponer de un sistema de escaneo y búsqueda para encontrar un cómic concreto.

*Tabla 4: Requisito funcional 4*

<b>Requisito funcional 5</b>	
Nombre	Visualizar colección
Descripción	La aplicación debe disponer de una vista donde visualizar la colección del usuario.

*Tabla 5: Requisito funcional 5*

<b>Requisito funcional 6</b>	
Nombre	Visualizar información de un cómic
Descripción	La aplicación debe disponer de una vista donde visualizar los distintos datos de un cómic concreto.

*Tabla 6: Requisito funcional 6*

<b>Requisito funcional 7</b>	
Nombre	Visualizar información de tiendas
Descripción	La aplicación debe disponer de una vista donde visualizar los distintos datos de una tienda de cómics concreto.

*Tabla 7: Requisito funcional 7*

<b>Requisito funcional 8</b>	
Nombre	Obtención de datos
Descripción	La aplicación debe ser capaz de realizar operaciones de recuperación en una base de datos.

*Tabla 8: Requisito funcional 8*

<b>Requisito funcional 9</b>	
Nombre	Inserción de datos
Descripción	La aplicación debe ser capaz de realizar operaciones de inserción en una base de datos.

*Tabla 9: Requisito funcional 9*

### Requisitos no funcionales

<b>Requisito no funcional 1</b>	
Nombre	Sistema operativo (SO)
Descripción	La aplicación debe utilizarse en dispositivos con sistema operativo <i>Android</i> .

*Tabla 10: Requisito no funcional 1*

Requisito no funcional 2	
Nombre	Versión del SO
Descripción	La aplicación debe utilizarse en dispositivos con la versión de <i>Lollipop</i> o superior ( $\geq$ <i>Android</i> 5.0)

Tabla 11: Requisito no funcional 2

Requisito no funcional 3	
Nombre	Interfaz de usuario intuitiva
Descripción	La aplicación debe disponer de una interfaz de usuario sencilla e intuitiva donde el usuario sea capaz de desenvolverse con facilidad.

Tabla 12: Requisito no funcional 3

Requisito no funcional 4	
Nombre	Desarrollo “ <i>responsive</i> ”
Descripción	La aplicación debe ser capaz de adaptarse a distintas pantalla de Smartphone con distinta resolución.

Tabla 13: Requisito no funcional 4

### 2.3 Definición de casos de uso

Una vez nombrados los requisitos funcionales se han definido los casos de uso derivados de estos. A continuación se detallarán los casos de uso donde se destacará la implicación del usuario con la funcionalidad correspondiente.

#### Casos de uso

En el momento de registrarse el usuario se realizará una petición al servidor con los datos especificados para el registro. El primer paso que se realizará es comprobar con los datos ubicados en el servidor si existe un usuario con el mismo nombre introducido por el actual usuario. Si no existe ningún usuario con el mismo nombre se registrará el usuario en los datos del servidor. En caso contrario no se permitirá realizar la acción.

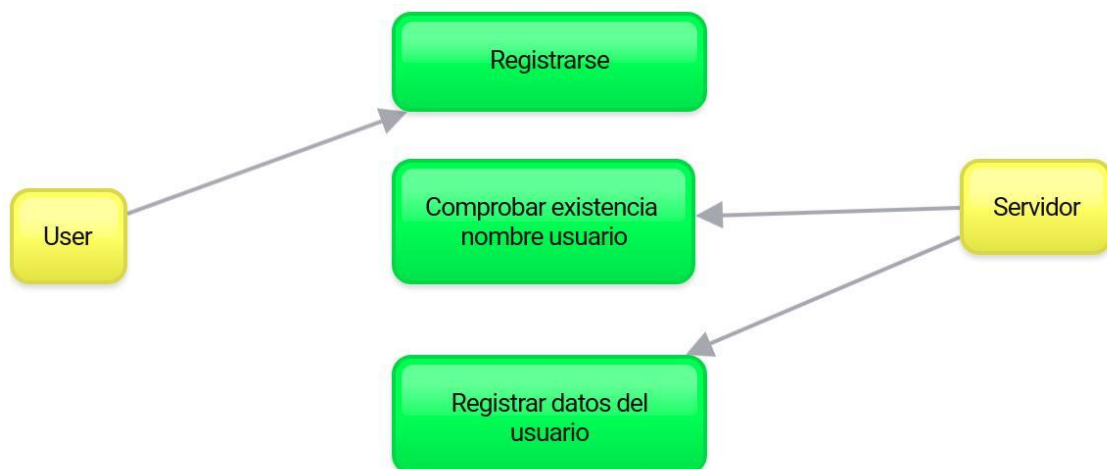


Fig. 9: Caso de uso del registro del usuario en la aplicación

El proceso de inicio de sesión realizará el mismo paso de comprobación que al registrarse con la diferencia que no requerirá la acción extra de registrar datos en el servidor.

Una vez comprobada la existencia del nombre del usuario en los datos del servidor se verificará que las credenciales introducidas (nombre y contraseña) coincidan con los datos registrados del usuario. En caso de obtener acceso se visualizará la pantalla inicial y se realizará una petición para visualizar las noticias/novedades recientes relacionadas con cómics.

El proceso será el de realizar una petición de recuperación de datos a la base de datos interna que contendrá un listado de novedades que habrá registrado previamente del servidor para visualizarlos en pantalla.

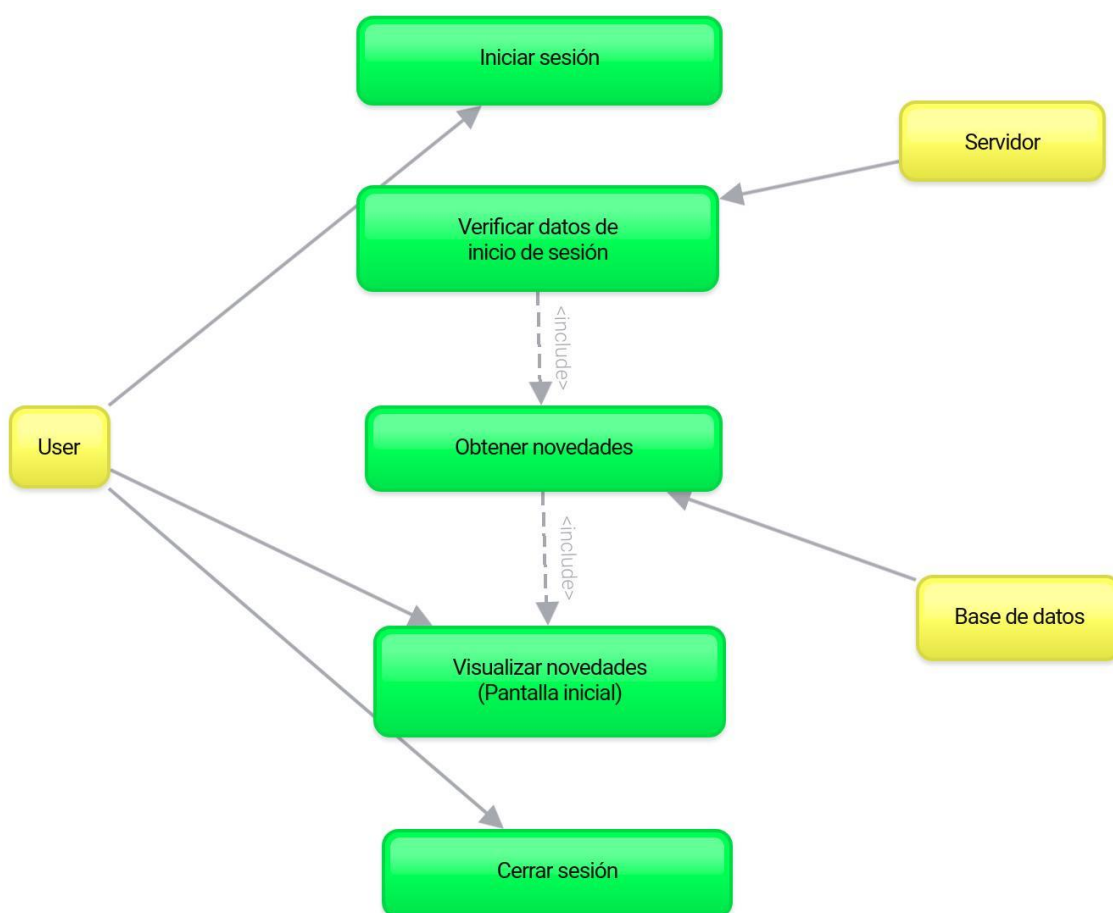


Fig. 10: Caso de uso del inicio de sesión del usuario en la aplicación

En el caso de la búsqueda el inicio sería similar al de los dos casos anteriores. Al realizar la petición de búsqueda se comprobará la existencia de un elemento que coincida con el dato que se ha introducido. En caso de no encontrar ninguna coincidencia simplemente se finalizará el proceso sin resultados impidiendo completar la acción.

En caso de encontrar una coincidencia se completará el proceso realizando una petición al servidor para recuperar los datos del cómic buscado para ser mostrados por pantalla.

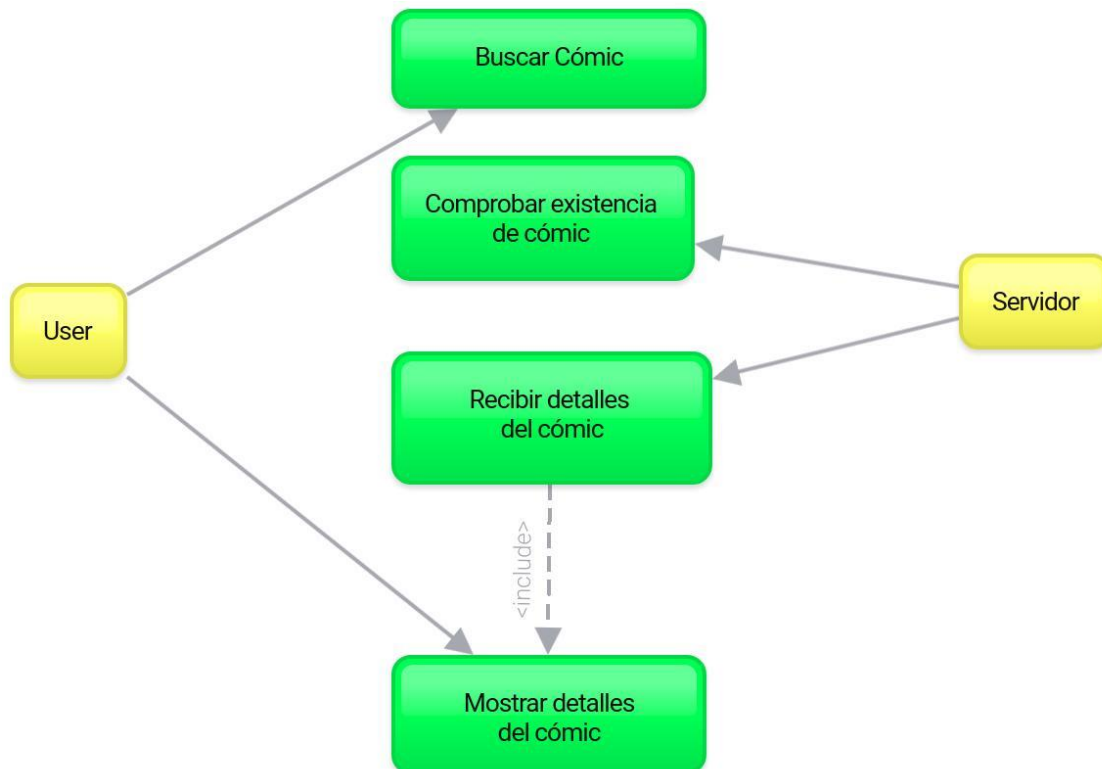


Fig. 11: Caso de uso de la búsqueda de un cómic

En el caso de que el usuario quiera acceder a los datos de su colección se realizará una petición a la base de datos para recibir los datos de cada uno de los cómics que se encuentran registrados en la colección del usuario para mostrarlos en forma de listado.

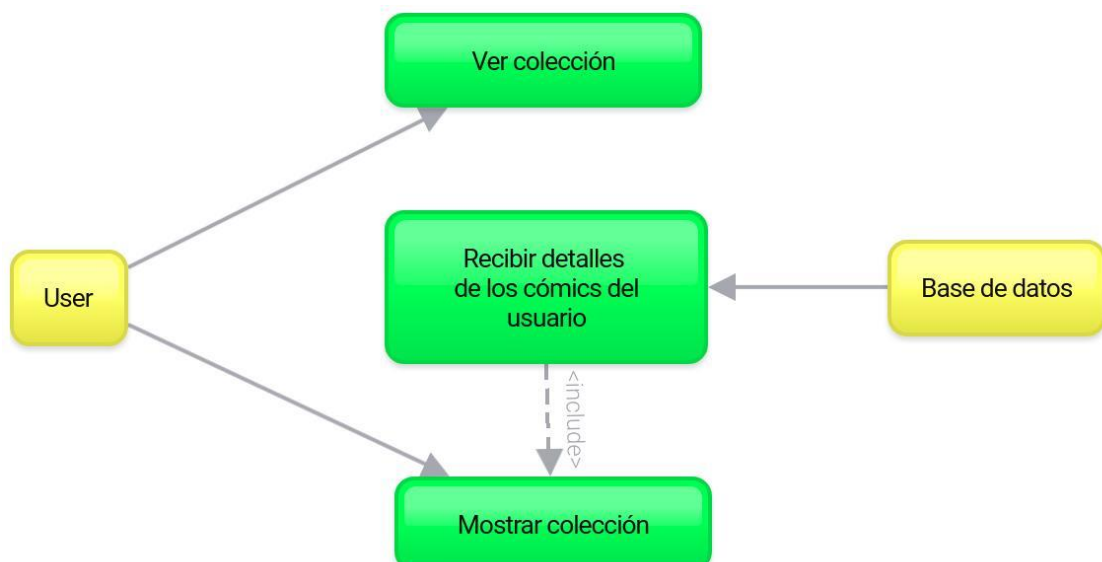


Fig. 12: Caso de uso de la visualización de la colección del usuario

El usuario podrá acceder de dos maneras distintas a los detalles al completo de un cómic: desde su colección y a través de una búsqueda. En este proceso se realizará una petición a la base de datos interna para recuperar los datos del cómic en cuestión para poder mostrarlos por pantalla.



Una vez mostrados los detalles por pantalla, en caso de haber accedido desde una búsqueda, y no haber añadido aún dicho cómic en la colección, el usuario tendrá la opción de añadirlo realizando una petición de registro al servidor.

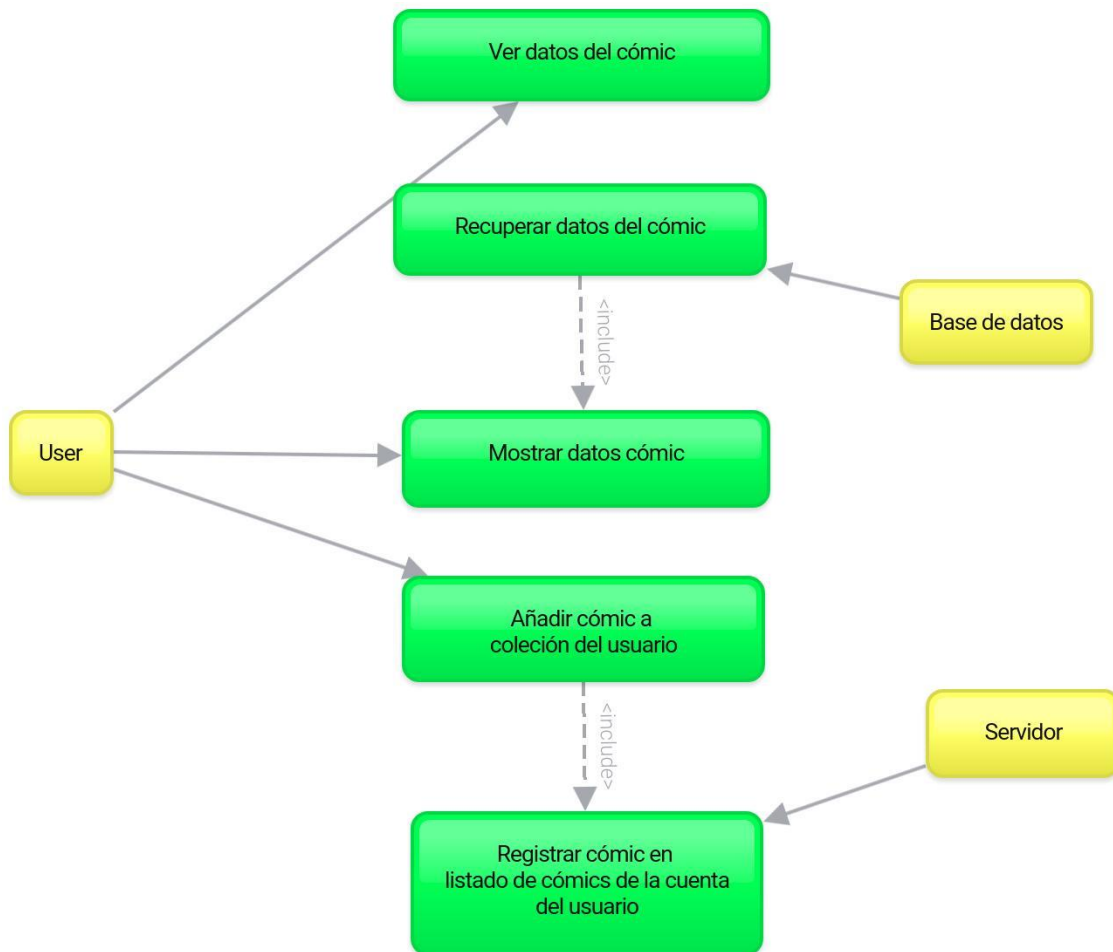


Fig. 13: Caso de uso de la visualización de los detalles de un cómic

El usuario podrá acceder a los detalles de una tienda desde el propio listado de tiendas. En este proceso se realizará una petición a la base de datos interna para recuperar los datos de la tienda en cuestión para poder mostrarlos por pantalla.

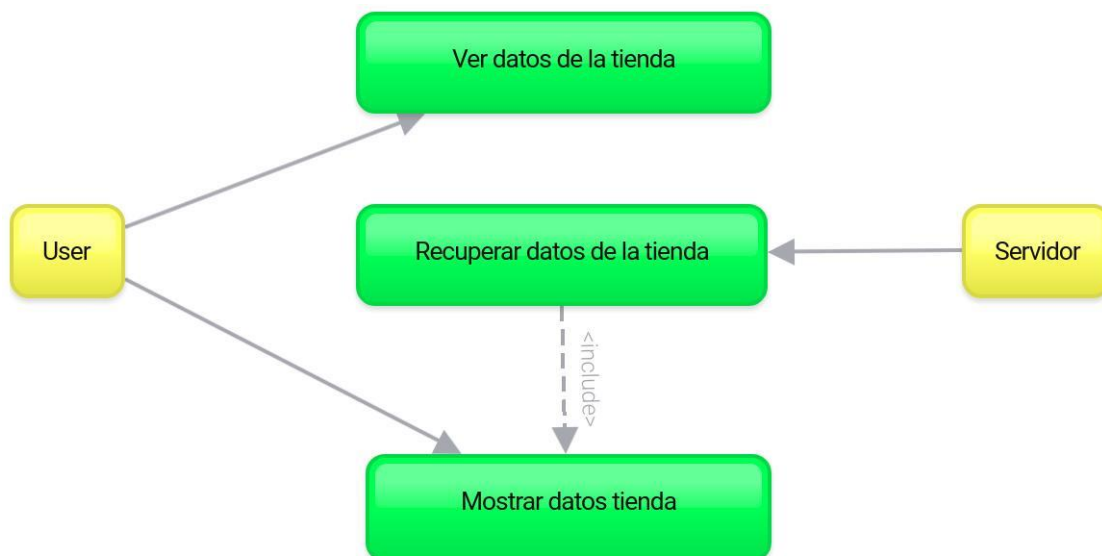


Fig. 14: Caso de uso de la visualización de los detalles de una tienda

El propósito del apartado de las tiendas es que sea gestionado por los propios usuarios, es decir, que sean ellos los que introduzcan la información de las tiendas para que otros con menos experiencia en la compra de cómics puedan estar más informados.

El usuario podrá enviar una petición de registro enviando datos de una tienda para ser registrado por el servidor para que cuando se realice un volcado de los datos de las tiendas disponible en la cache de la aplicación en el dispositivo del usuario.



Fig. 15: Caso de uso del registro de una nueva tienda

Si el usuario quiere ver los comentarios realizados sobre un cómic o una tienda se realizará una petición a base de datos que devuelva todos los comentarios relacionados directamente con dicho elemento cacheados del servidor.

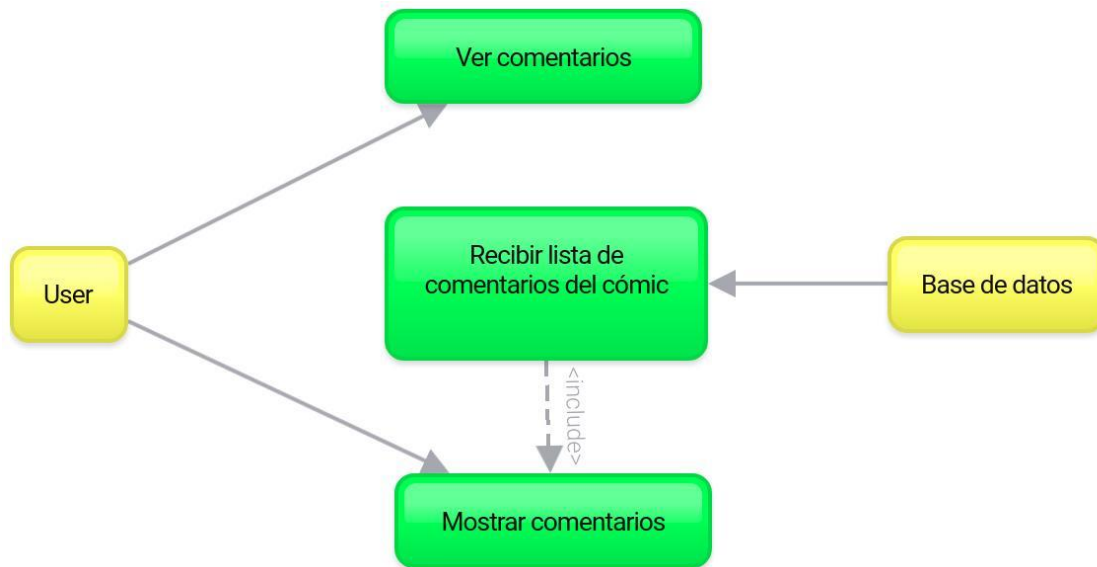


Fig. 16: Caso de uso de la visualización de los comentarios de un cómic

En el caso de añadir un nuevo comentario sobre un cómic o tienda se enviará una petición con los datos necesarios, como el nombre del usuario o el mensaje, para insertarlo en el listado de comentarios del servidor. De esta manera cuando se vuelva cachear la lista de comentarios se dispondrá del nuevo comentario.



Fig. 17: Caso de uso del registro de un nuevo comentario

Cuando el usuario quiera conocer la valoración de un cómic o una tienda se realizará una petición a base de datos que devuelva todas las valoraciones cacheadas del servidor para ese elemento en concreto.

En el momento de mostrar la valoración lo que verá el usuario será la media de todas las valoraciones registradas y la suya propia.

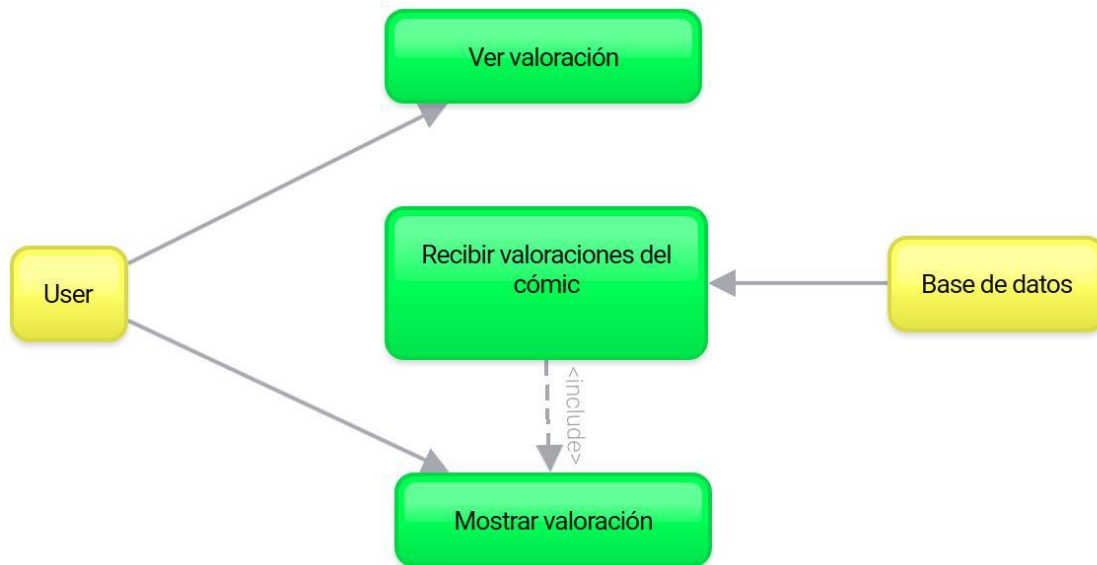


Fig. 18: Caso de uso de la visualización de la valoración de un cómic

En el caso de añadir una valoración sobre un cómic o una tienda se puede encontrar dos casos, el primero que el usuario nunca antes haya valorado ese mismo cómic o tienda y el segundo que si lo haya hecho con anterioridad.

Por ello se realizará una comprobación para saber si ya existe valoración del usuario en cuestión. En caso que ya exista se modificará el valor actual en el servidor y, en caso contrario, se registrará una nueva valoración con el nombre del usuario y su valoración en el servidor.

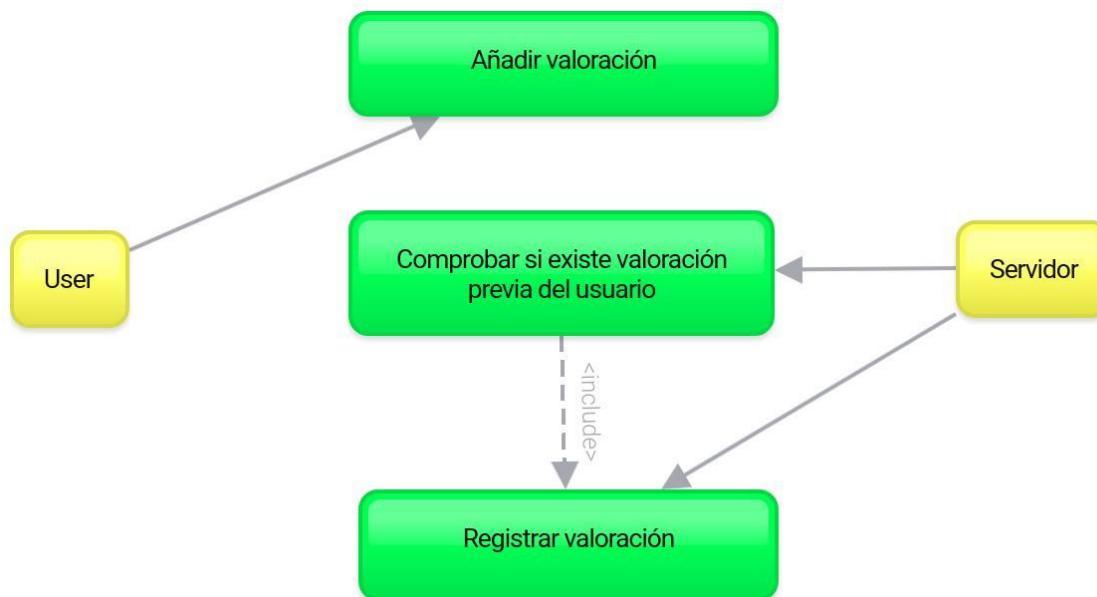


Fig. 19: Caso de uso del registro de una nueva valoración

## 3. Diseño

### 3.1 Árbol de navegación

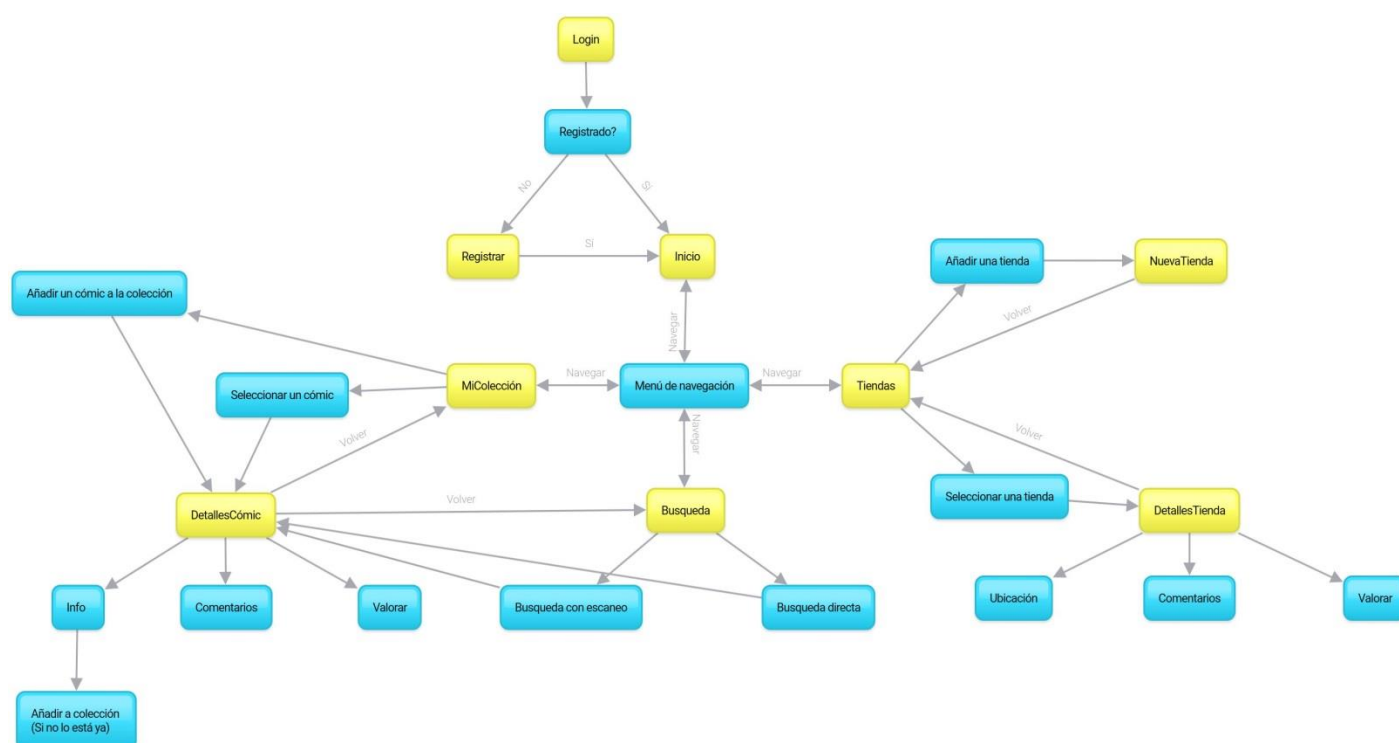


Fig. 20: Árbol de navegación de la aplicación (amarillo -> pantallas, azul -> detalles)

La disposición de la navegación de la aplicación es la representada en la imagen de arriba (figura 9). Al iniciar la aplicación se mostrará la pantalla de *login* donde el usuario podrá iniciar sesión y acceder al contenido de la aplicación o acceder a la pantalla de registro, en caso de no estarlo ya, donde se registrará para conseguir acceso.

Una vez iniciada la sesión el usuario será dirigido a la pantalla de inicio donde se mostrarán las últimas novedades del mundo del cómic y, principalmente, se dispondrá del menú de navegación que permitirá acceder al resto de secciones: *MiColección*, *Búsqueda* y *Tiendas*.

En la pantalla *MiColección* se visualizará la colección de cómics del usuario donde el usuario podrá realizar dos acciones: seleccionar un cómic y acceder a la pantalla *DetallesCómico*, donde visualizará la información del cómic, y añadir un cómic accediendo a la pantalla de *Búsqueda*.

La pantalla de búsqueda permitirá al usuario buscar un cómic concreto a través de su código *ISBN*. Existirán dos métodos para referenciar ese código: escribiéndolo manualmente con el teclado o escaneándolo con el dispositivo. Si se encuentra un resultado para el código introducido se accederá a la pantalla *DetallesCómico* y en caso de no estar vinculado a la colección se visualizará un botón para dar la posibilidad de añadirlo.

Por otro lado, la pantalla *Tiendas* donde se visualizarán las tiendas disponibles en base de datos y desde donde se podrá acceder a la pantalla *DetallesTienda* para visualizar la información de la tienda o a la pantalla *NuevaTienda* para añadir una nueva tienda a la base de datos.

### 3.2 Diseño de arquitectura

Actualmente, en *Android* se dispone de tres tipos de arquitectura principales que suelen ser los más utilizados. Estas tres arquitecturas son el *modelo vista controlador* (MVC), *modelo vista presentador* (MVP) y el *modelo vista vista modelo* (MVVM).

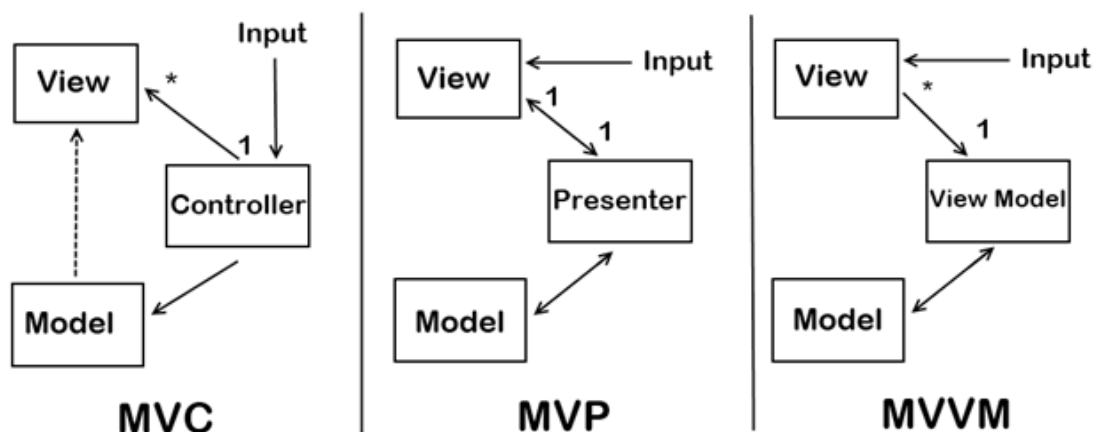


Fig. 21 – Tipos de arquitectura en Android

En el *modelo vista controlador* el usuario envía una orden al *controlador* que es el que comunica directamente con ambos para indicar tanto a *vista* como a *modelo* lo que deben hacer. No obstante, la comunicación no se produce de manera inversa. El *controlador* se encarga de decir a la vista como ha de actualizarse e informa al *modelo* los datos que ha de entregar a la *vista*.

En el caso del *modelo vista presentador*, el *modelo* y la *vista* no tienen relación directa. El usuario interactúa con la *vista* que le indica la acción a realizar o los datos que necesita y el presentador se encarga de realizar la acción o comunicarse con el *modelo* para pedir los datos y entregárselos a la *vista* para mostrarlos, actualizando así la *vista*.

El *modelo vista vista modelo* es quizá la más novedosa de las tres arquitecturas. Esta arquitectura es parecida al modelo vista presentador con la diferencia que la comunicación entre la *vista* y la *vista modelo* no es bidireccional. En este caso la vista es la que indica a la vista modelo que acción realizar y esta se comunica con el modelo para obtener los datos.

Lo mejor que aporta esta estructura es el uso del componente *LiveData*. Este componente es un elemento observable que permite ser observado desde otra capa de la estructura como la *vista* sin necesidad de crear dependencias extra.

Gracias al *LiveData* la *UI* solo ha de comunicarse con la *vista modelo* a través de un evento para procesar la acción y el componente *LiveData* creado en la *vista modelo* se encarga de actualizar los datos entregados por el *modelo* que serán observados desde la *vista*.

Además, la *vista modelo* permite conservar y actualizar los cambios de la *vista* aunque se produzcan cambios en la configuración de la pantalla, como un cambio de orientación.

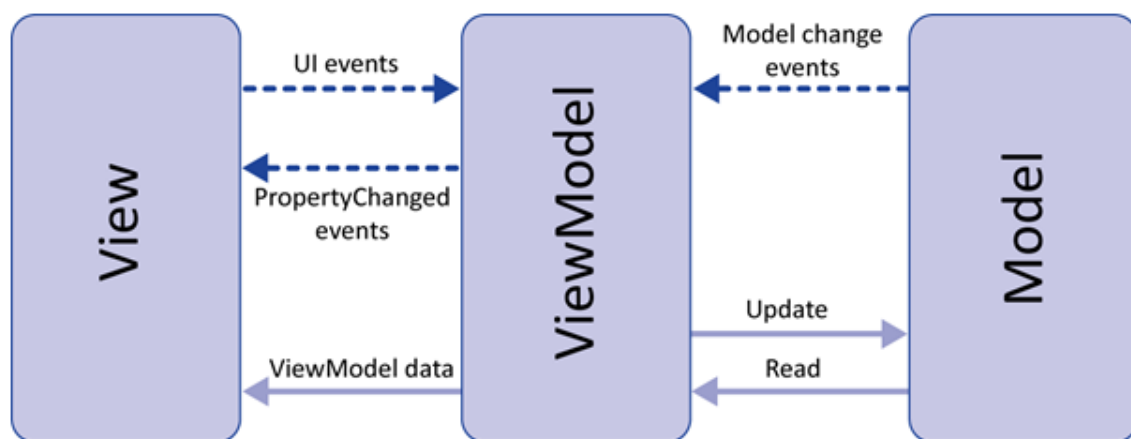


Fig. 22 – Estructura del Model View ViewModel (MVVM)

Teniendo en cuenta estas ventajas, se ha decidido utilizar la arquitectura *modelo vista vista modelo* para el desarrollo de la aplicación.

### 3.3 Diseño de la estructuración de datos

Los datos de la aplicación serán gestionados desde *Firebase* en la parte servidor y desde *Realm* de forma interna en la aplicación.

En el lado del servidor, en *Firebase*, habrá cuatro tipos principales de elementos: los usuarios, las novedades, los cómics y las tiendas.

Cada elemento del tipo usuario contendrá el nombre de usuario, la contraseña encriptada, la ruta de su imagen y la colección o lista de cómics, que en realidad será una lista de los códigos *isbn* de los cómics.

Los elementos del tipo novedad contendrá el título de la noticia, la url de la página que informa de la noticia y la *url* de una imagen que representa la noticia.

En el caso del cómic contendrá el nombre del autor, el listado de comentarios, el nombre de la editorial, el género, el código *isbn*, la sinopsis, el título, la url de la imagen de la portada y el listado de valoraciones.

Los elementos de tipo tienda contendrán el listado de comentarios, la dirección de la tienda, el nombre de la tienda, la provincia donde está situada y la lista de valoraciones.

Los comentarios y valoraciones contendrán el nombre del usuario que ha comentado/valorado, el mensaje/valoración enviado y, en el caso de los comentarios, la fecha y hora en la que realizó el comentario.

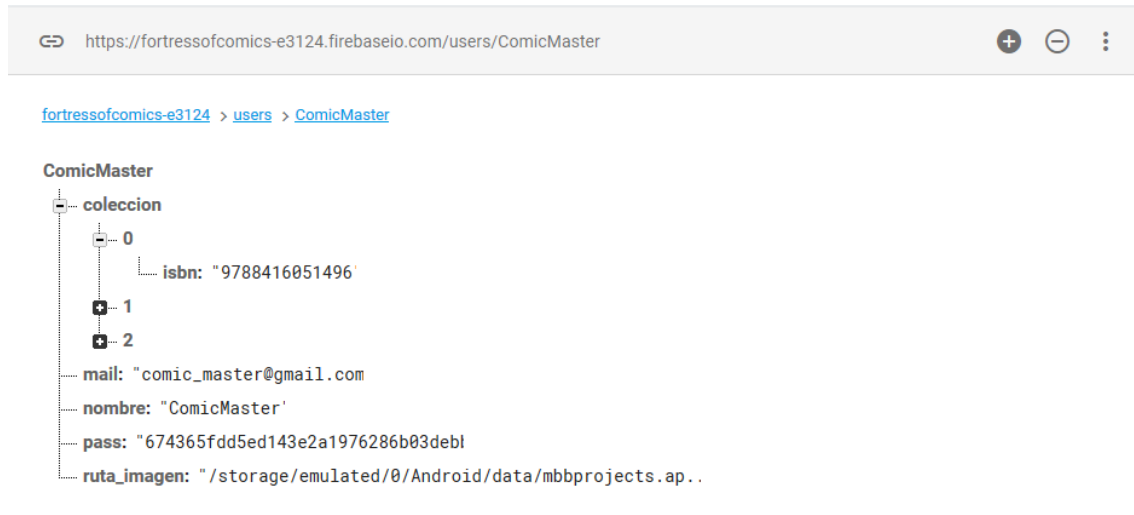


Fig. 23 – Estructura de los usuarios en Firebase



Fig. 24 – Estructura de las novedades en Firebase





Fig. 25 – Estructura de los cómics en Firebase

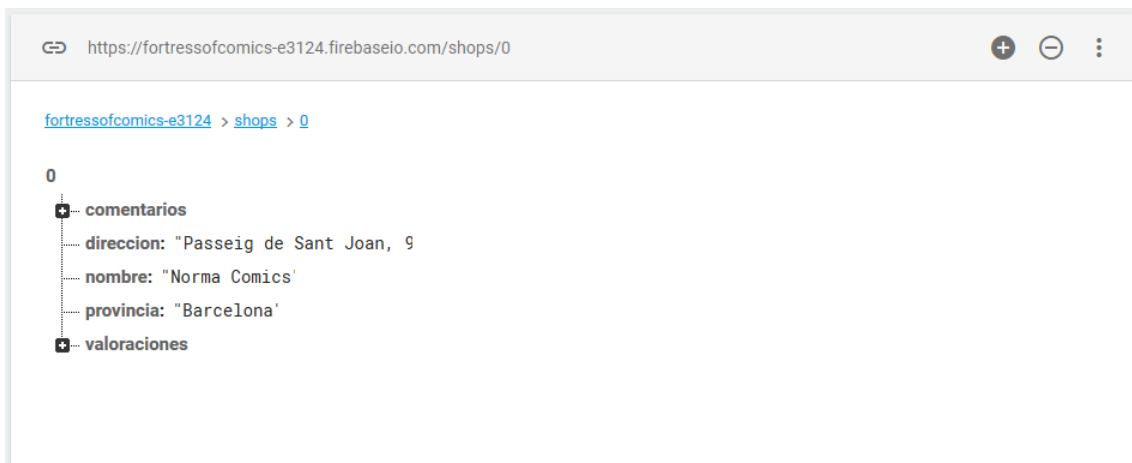


Fig. 26 – Estructura de las tiendas en Firebase

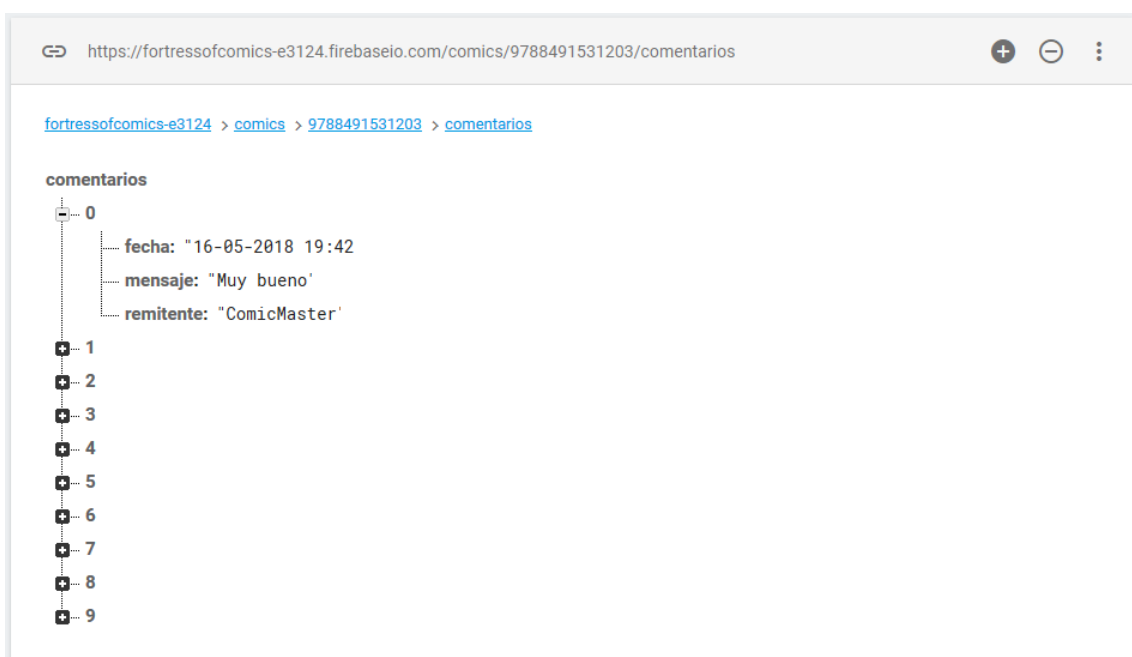


Fig. 27 – Estructura de los comentarios en Firebase

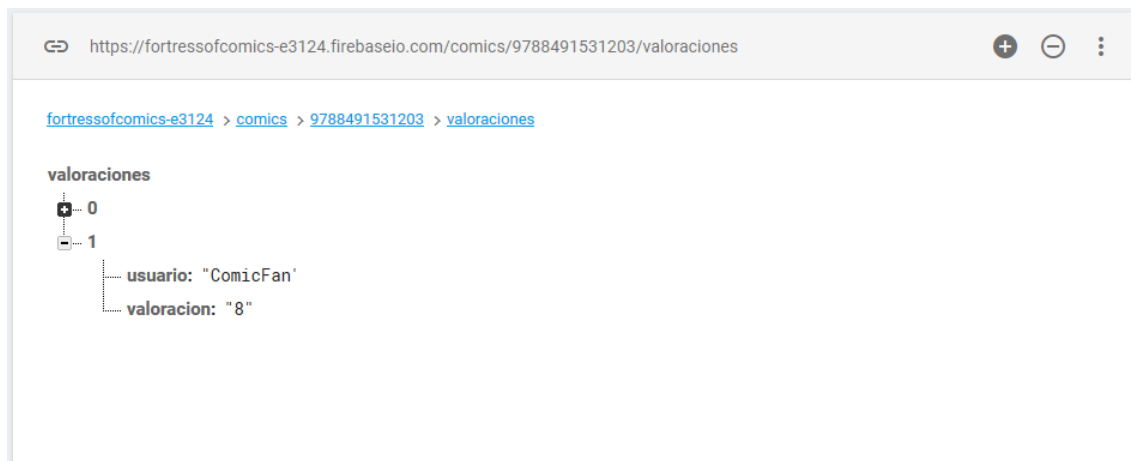


Fig. 28 – Estructura de las valoraciones en Firebase

En el lado de la aplicación se trabajará con un sistema de datos no relacional que almacenará los datos por separado. Incluso en algunos casos se almacenarán por separado datos que en el lado del servidor están juntos para tener una más fácil gestión de estos dentro de la aplicación. Este es el caso del usuario y su colección y del cómic o la tienda y sus comentarios y valoraciones.

La estructura en el caso de *Realm* es la siguiente:

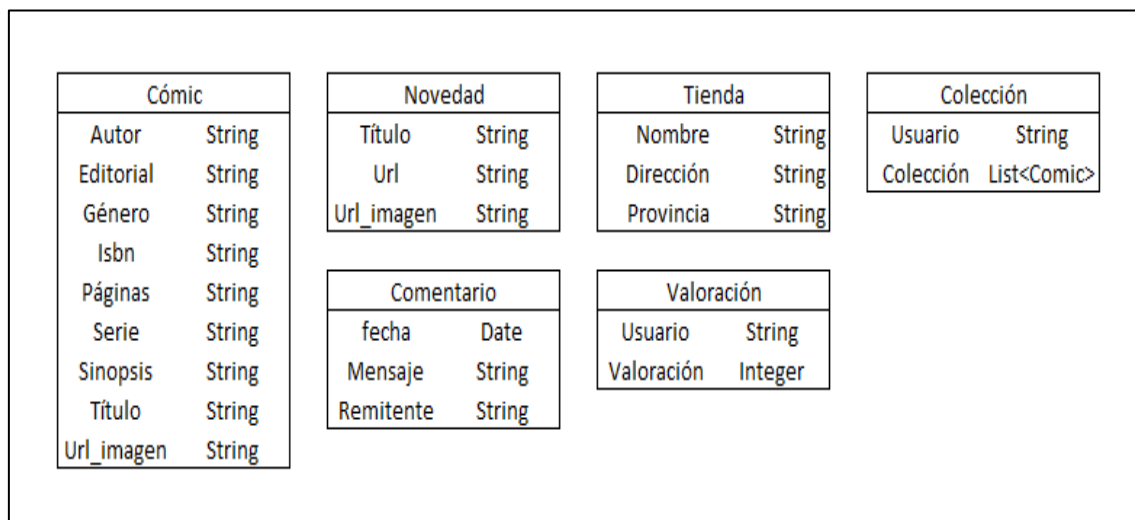


Fig. 29 – Diagramas de las entidades de Realm

### 3.4 Diseño de las pantallas: wireframes y prototipo

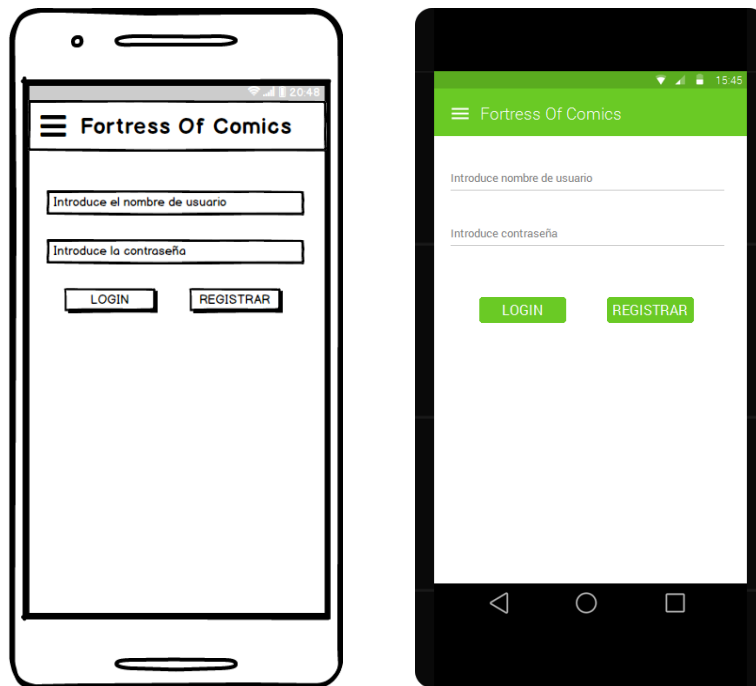


Fig.30 – Wireframe y prototipo del diseño de la pantalla de login

La *figura 10* representa el diseño de la pantalla de *login*. En esta pantalla el usuario dispondrá de dos campos de entrada, donde deberá introducir el nombre de usuario y la contraseña respectivamente, y dos botones para realizar el *login* y acceder al contenido de la aplicación o para acceder a la pantalla de registro en caso de no tener acceso.

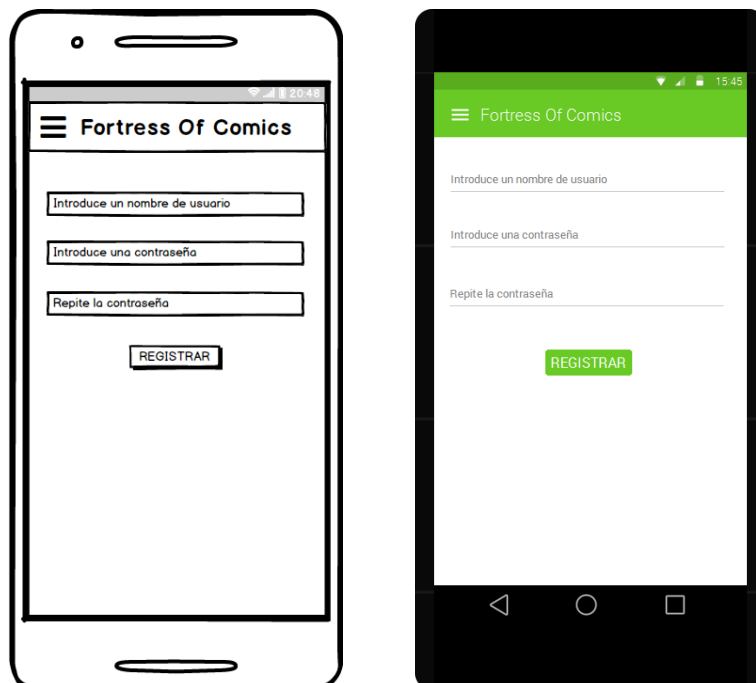


Fig. 31 – Wireframe y prototipo del diseño de la pantalla de registro

La *figura 11* representa el diseño de la pantalla de registro. En esta pantalla el usuario dispondrá de tres campos de entrada, donde deberá introducir el nombre de usuario y la contraseña dos veces, y un botón para formalizar el registro.

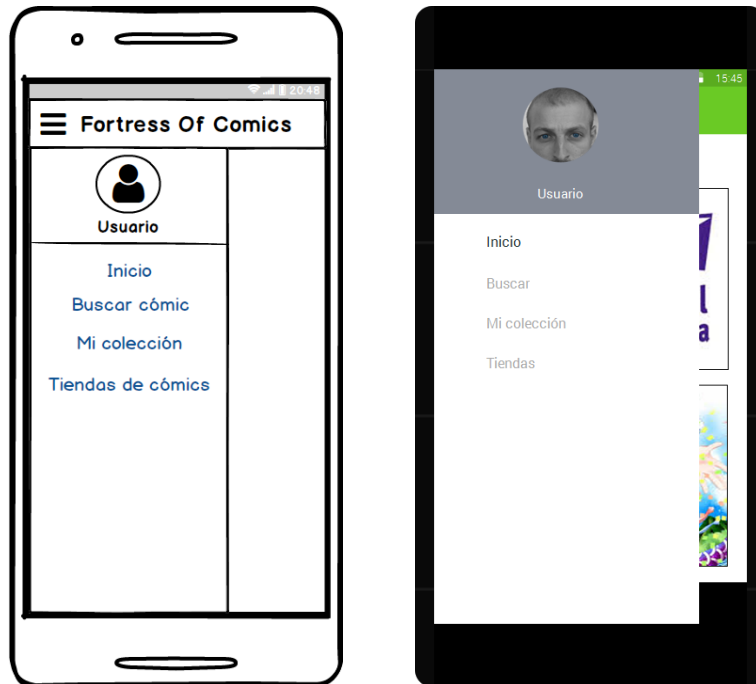


Fig. 32 – Wireframe y prototipo del diseño del menú lateral de navegación

La *figura 12* representa el diseño del menú lateral de navegación. Este menú dispondrá de un encabezado con el nombre de usuario y un espacio donde el usuario podrá colocar una imagen. Las opciones de navegación disponibles serán las cuatro pantallas principales: *Inicio*, *Buscar*, *Mi colección* y *Tiendas*.



Fig. 33 – Wireframe y prototipo del diseño de la pantalla de inicio

La *figura 13* representa el diseño de la pantalla de inicio. En esta pantalla se mostrarán anuncios y novedades del mundo del cómic como, por ejemplo, las promociones para el salón del cómic o del manga.



Fig. 34 – Wireframe y prototipo del diseño de la pantalla de Mi colección

La figura 14 representa el diseño de la pantalla de Mi colección. En esta pantalla se mostrarán las portadas de los cómics que el usuario tiene asignado a su colección. Al realizar *click* en una imagen se accede a los detalles del cómic (figura 16) y al hacerlo sobre el botón flotante se permite buscar un cómic para añadir a la colección (figura 15).



Fig. 35 – Wireframe y prototipo del diseño de la pantalla de búsqueda

La figura 15 representa el diseño de la pantalla de búsqueda. En esta pantalla el usuario dispondrá de un campo de entrada, donde introducir el código *ISBN* del cómic, y de dos botones, el primero accederá al sistema de escaneo con la

cámara del dispositivo para obtener automáticamente el código y el segundo para realizar la búsqueda en la base de datos.



Fig. 36 – Wireframe y prototipo del diseño de la pantalla de detalles del cómic

La figura 16 representa el diseño de la pantalla de detalles del cómic. En esta pantalla se mostrarán los principales datos de cada cómic, así como la oportunidad de añadirlo a la colección si fuera necesario. Esta pantalla contendrá otras dos secciones: comentarios y valorar (figuras 17 y 19).

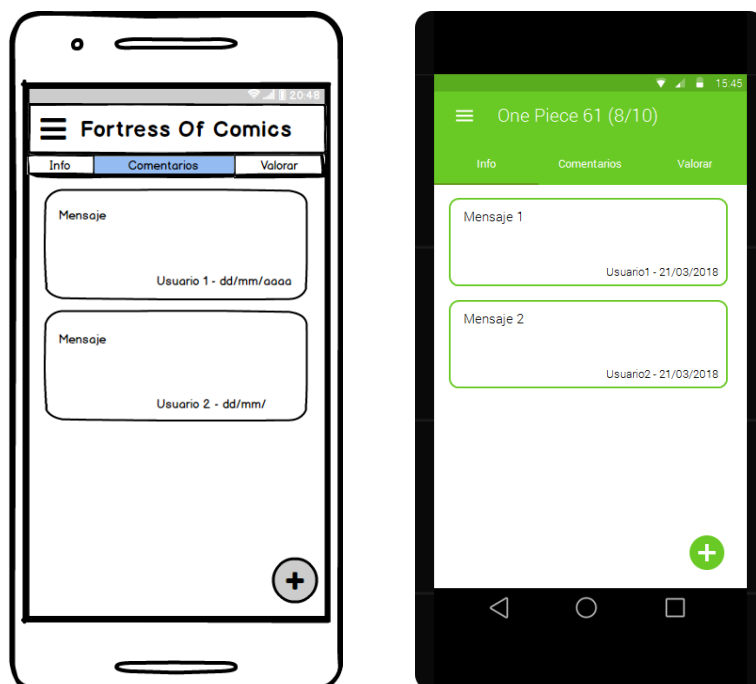


Fig. 37 – Wireframe y prototipo del diseño de la pantalla de comentarios

La figura 17 representa el diseño de la pantalla de comentarios. En esta pantalla se mostrarán los comentarios que vayan enviando los usuarios de un

cómic. Cada comentario contendrá el mensaje, el nombre del usuario y la fecha de envío. El botón flotante permite añadir un nuevo comentario (*figura 18*).



Fig. 38 – Wireframe y prototipo del diseño de la pantalla de nuevo comentario

La *figura 18* representa el diseño de la pantalla de nuevo comentario. En esta pantalla el usuario dispondrá de un campo de entrada donde introducir el mensaje y un botón para confirmar el envío del comentario.

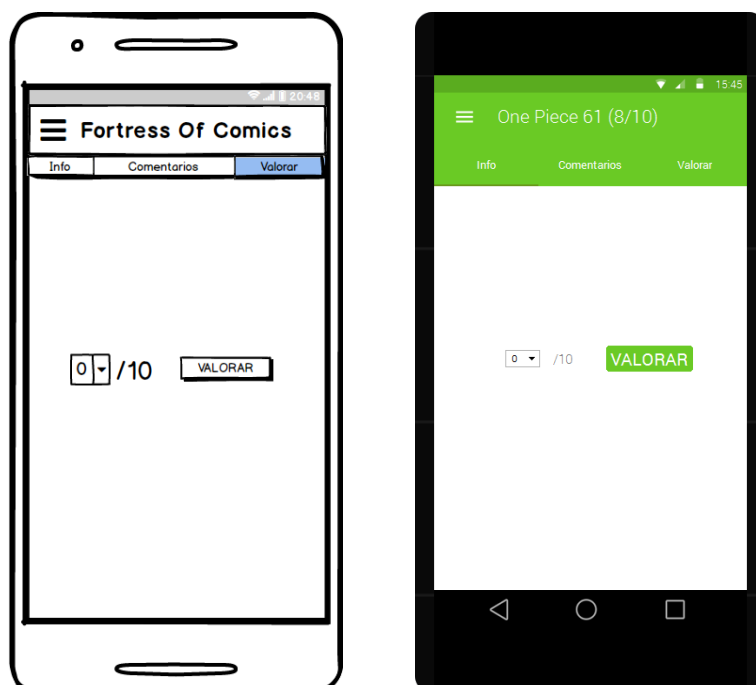


Fig. 39 – Wireframe y prototipo del diseño de la pantalla de valorar

La *figura 19* representa el diseño de la pantalla de valorar. En esta pantalla se mostrarán anuncios y novedades del mundo del cómic como, por ejemplo, las promociones para el salón del cómic o del manga.



Fig. 40 – Wireframe y prototipo del diseño de la pantalla de tiendas

La figura 20 representa el diseño de la pantalla de tiendas. En esta pantalla se mostrará una lista de las tiendas que se han añadido en la base de datos de la aplicación. Cada elemento de la lista mostrará el nombre, valoración y provincia. Al realizar *click* en cualquier elemento se accede a los detalles de la tienda (figura 22) y al hacerlo en el botón flotante se permite añadir una nueva tienda (figura 21).

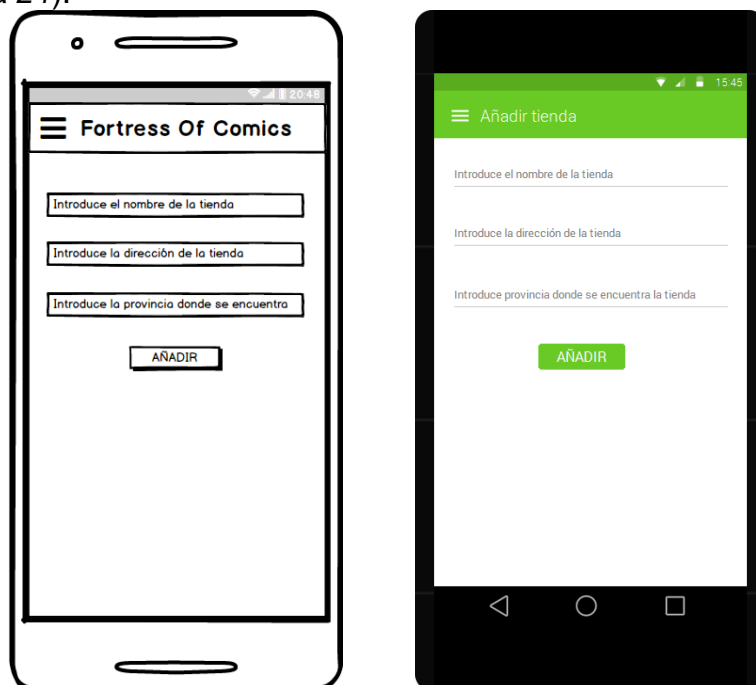


Fig. 41 – Wireframe y prototipo del diseño de la pantalla de nueva tienda

La figura 21 representa el diseño de la pantalla de nueva tienda. En esta pantalla el usuario dispondrá de tres campos de entrada donde introducir el



nombre de la nueva tienda, la dirección y la provincia donde se encuentra. Por último, el botón que confirma el proceso de añadir la tienda a la base de datos.

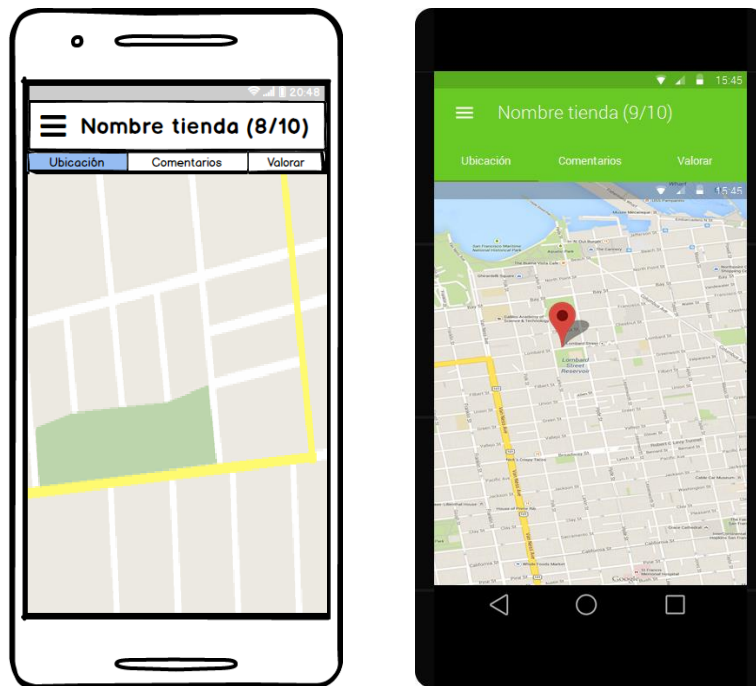


Fig. 42 – Wireframe y prototipo del diseño de la pantalla de detalles de tienda

La *figura 22* representa el diseño de la pantalla de detalles de la tienda. En esta pantalla se mostrará la ubicación de la tienda en un mapa para poder localizarla. A parte, se podrá acceder a dos secciones idénticas a las de las *figuras 17* y *19* pero referentes a la tienda.

## 4. Implementación

### 4.1 Estructuración del proyecto

El proyecto está estructurado en módulos que representan elementos principales de la app y cada módulo contiene tres módulos idénticos: *view*, *domain* y *model*. El módulo *view* contiene todos los archivos relacionados con la vista (*activities*, *adapters*, *fragments*), el módulo *domain* contiene los casos de usos y los modelos de los elementos y el módulo de *model* contiene todos los archivos relacionados con la escritura y lectura de datos, tanto en servidor como en base de datos.

### 4.2 Decisiones de implementación

En el momento de llevar a cabo la implementación se ha intentado seguir todas las decisiones que se habían fijado durante la etapa de diseño. La estructura comentada anteriormente ha sido fruto de utilizar la arquitectura *MVVM* donde los tres módulos descritos que contiene cada módulo principal representan las capas de esta arquitectura. Dentro de cada módulo *view* encontramos un archivo *viewmodel* donde se implementa el funcionamiento de los elementos *livedata* y se realiza la llamada a los casos de uso a partir de las peticiones de la vista.

La parte de estructuración de datos se ha manejado con *Realm* de forma interna y para comunicarse con el servidor se ha utilizado *Retrofit* para recibir la respuesta y *Gson* para mapear automáticamente el *json* en un objeto definido en la aplicación. Todos los procesos de llamada de datos se han realizado aplicando *RxJava* consiguiendo así un flujo asíncrono fuera del hilo principal al mismo tiempo que se mapean las diferentes entidades.

Para realizar el escaneo en la página de las búsquedas se ha importado la librería *ZXing* ("*Zebra crossing*"), de fácil uso para un simple escaneo de código de barras. En la pantalla de localización de la tienda se ha utilizado la librería de mapas de *Google*.

### 4.3 Resultado final: Pantallas de la aplicación



Fig. 43 – Capturas de la pantalla de inicio de sesión y de registro

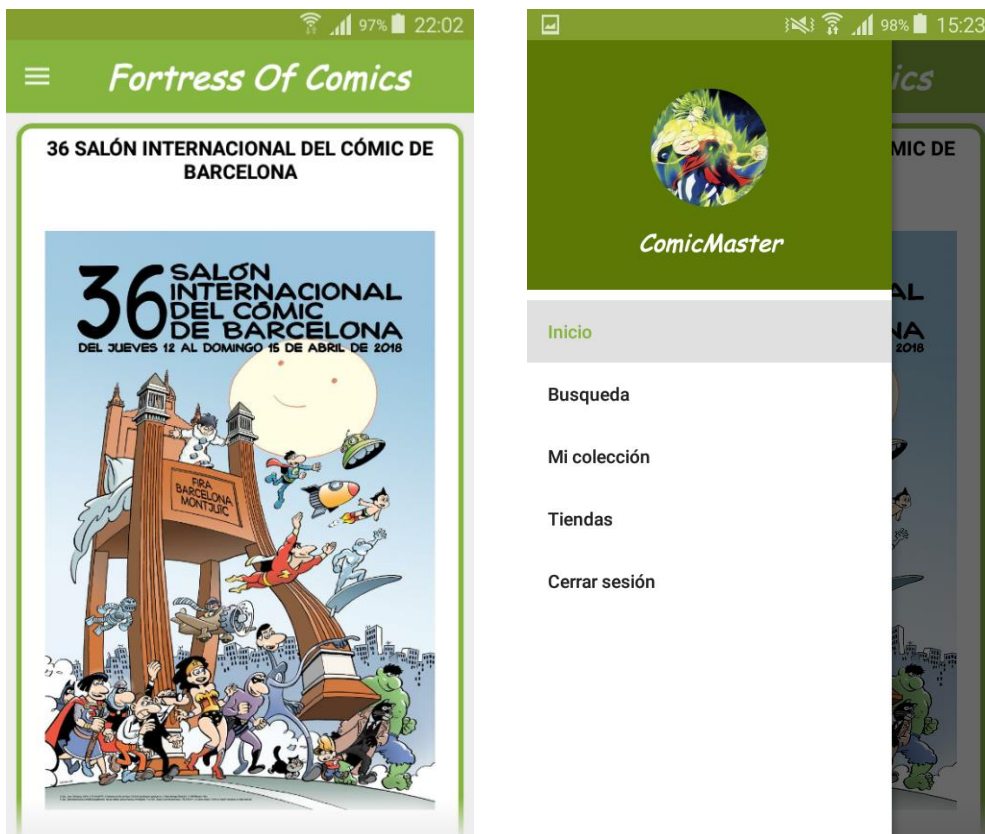


Fig. 44 – Capturas de la pantalla de novedades y del menú lateral



Fig. 45 – Capturas de la pantalla de editar el perfil y de búsqueda

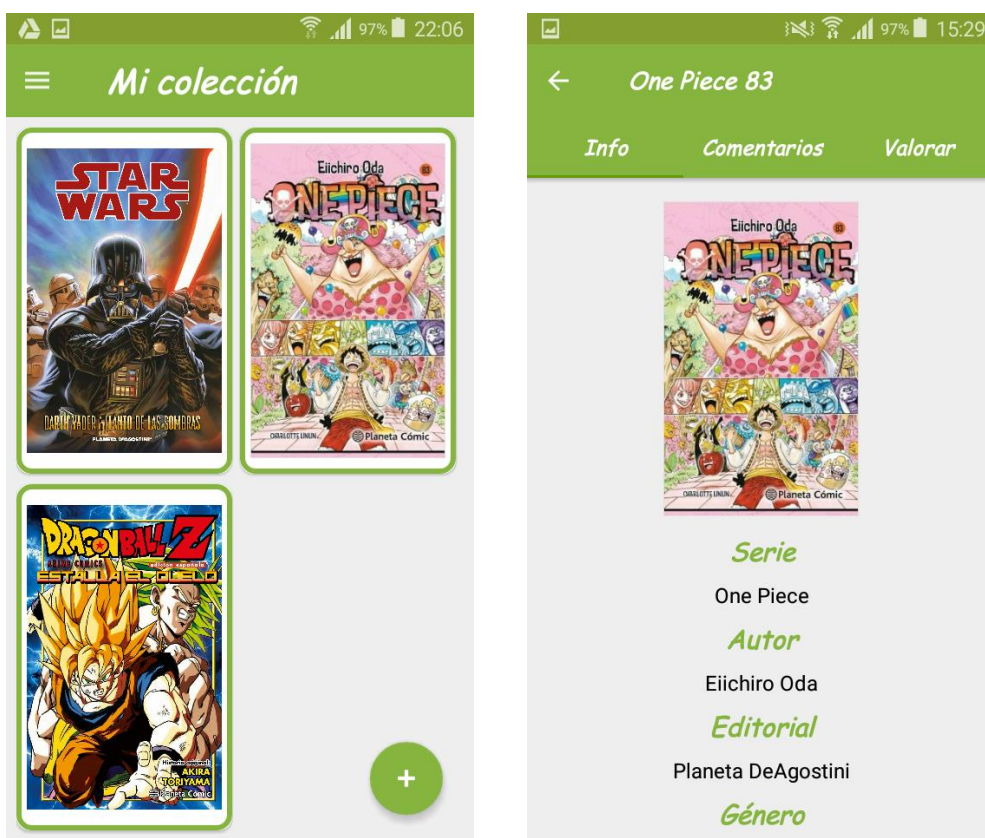


Fig. 46 – Capturas de la pantalla de mi colección y de detalles del cómic

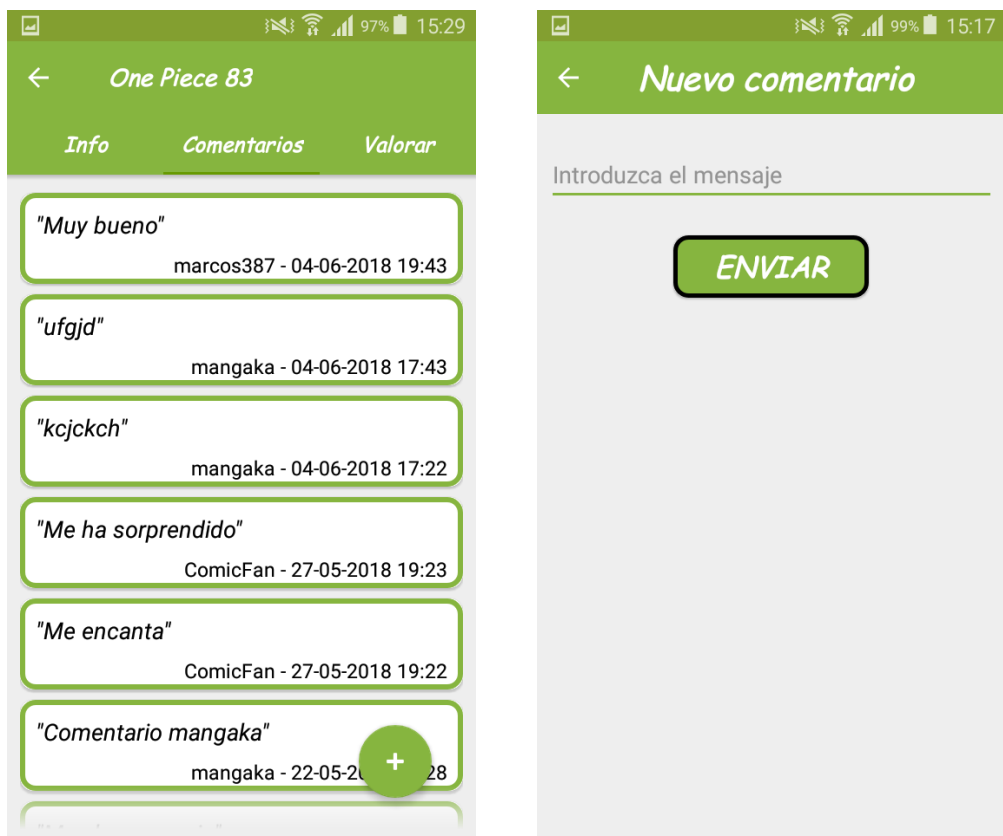


Fig. 47 – Capturas de la pantalla de comentarios del cómic y de nuevo comentario



Fig. 48 – Capturas de la pantalla de valoraciones del cómic y de tiendas de cómics

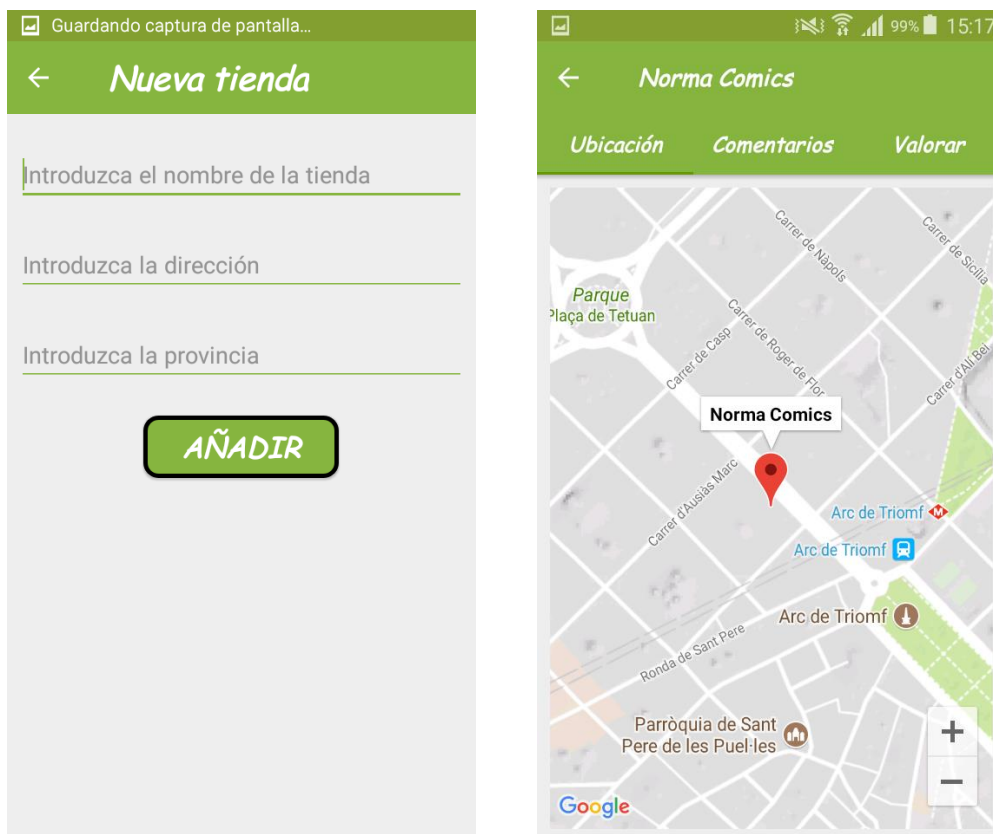


Fig. 49 – Capturas de la pantalla de nueva tienda y de localización de la tienda

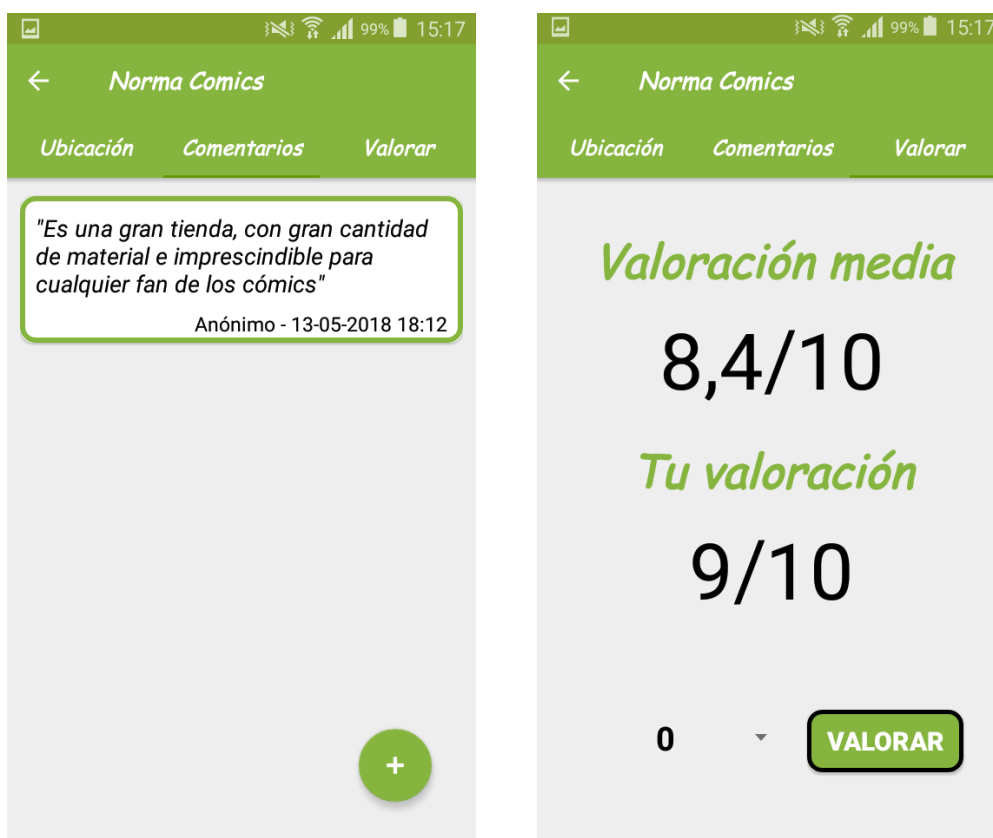


Fig. 50 – Capturas de la pantalla de comentarios y de valoraciones de la tienda

## 5. Pruebas

Una vez realizada la implementación de las funcionalidades de la aplicación se ha llevado a cabo una prueba correspondiente para verificar su correcto funcionamiento. Al realizar estas pruebas se han podido descubrir posibles fallos que han ayudado a corregir estos defectos y obtener el resultado esperado por las pruebas.

### 5.1 Pruebas de la pantalla de inicio de sesión

Prueba 1	
Nombre	Campos vacíos
Pasos	Entra en la pantalla de inicio de sesión.
	Pulsa <i>LOGIN</i> sin haber rellenado los campos.
	Se muestra mensaje avisando al usuario.
Requisitos	Ninguno
Expectativas	Se muestra mensaje informando al usuario.

Tabla 14: Prueba 1 de la pantalla de inicio de sesión

Prueba 2	
Nombre	Credenciales erróneas
Pasos	Entra en la pantalla de inicio de sesión.
	Pulsa <i>LOGIN</i> introduciendo datos incorrectos.
	Se muestra mensaje avisando al usuario.
Requisitos	Ninguno
Expectativas	Se muestra mensaje informando al usuario.

Tabla 15: Prueba 2 de la pantalla de inicio de sesión

Prueba 3	
Nombre	Credenciales correctas
Pasos	Entra en la pantalla de inicio de sesión.
	Pulsa <i>LOGIN</i> introduciendo datos correctos de un usuario existente.
	Se muestra mensaje avisando al usuario.
Requisitos	Ninguno
Expectativas	Se accede a la pantalla inicial de la aplicación.

Tabla 16: Prueba 3 de la pantalla de inicio de sesión

## 5.2 Pruebas de la pantalla de registro

Prueba 1	
Nombre	Campos vacíos
Pasos	Entra en la pantalla de registro.
	Pulsa <i>REGISTRAR</i> sin haber rellenado los campos.
	Se muestra mensaje avisando al usuario.
Requisitos	Internet
Expectativas	Se muestra mensaje informando al usuario.

Tabla 17: Prueba 1 de la pantalla de registro

Prueba 2	
Nombre	Usuario ya existe
Pasos	Entra en la pantalla de registro.
	Pulsa <i>REGISTRAR</i> introduciendo un nombre de usuario ya existente.
	Se muestra mensaje avisando al usuario.
Requisitos	Internet
Expectativas	Se muestra mensaje informando al usuario.

Tabla 18: Prueba 2 de la pantalla de registro

Prueba 3	
Nombre	Contraseñas no coinciden
Pasos	Entra en la pantalla de registro.
	Pulsa <i>REGISTRAR</i> al rellenar los campos de contraseña sin ser los valores idénticos.
	Se muestra mensaje avisando al usuario.
Requisitos	Ninguno
Expectativas	Se muestra mensaje informando al usuario.

Tabla 19: Prueba 3 de la pantalla de registro

Prueba 4	
Nombre	Credenciales correctas
Pasos	Entra en la pantalla de registro.
	Pulsa <i>REGISTRAR</i> introduciendo datos válidos.
	Se muestra mensaje avisando al usuario.
Requisitos	Internet
Expectativas	Se accede a la pantalla inicial de la aplicación.

Tabla 20: Prueba 4 de la pantalla de registro



### 5.3 Pruebas de la pantalla de editar el perfil

Prueba 1	
Nombre	Campos vacíos
Pasos	Entra en la pantalla de editar perfil.
	Pulsa <i>MODIFICAR</i> sin haber rellenado los campos de mail o contraseña (según el que se quiera modificar).
	Se muestra mensaje avisando al usuario.
Requisitos	Internet
Expectativas	Se muestra mensaje informando al usuario.

Tabla 21: Prueba 1 de la pantalla de editar el perfil

Prueba 2	
Nombre	Contraseñas no coinciden
Pasos	Entra en la pantalla de editar perfil.
	Pulsa <i>MODIFICAR</i> al rellenar los campos de contraseña sin ser los valores idénticos.
	Se muestra mensaje avisando al usuario.
Requisitos	Internet
Expectativas	Se muestra mensaje informando al usuario.

Tabla 22: Prueba 2 de la pantalla de editar el perfil

### 5.4 Pruebas de la pantalla de novedades

Prueba 1	
Nombre	Enlace funciona
Pasos	Entra en la pantalla de novedades.
	Pulsa en cualquier elemento de la lista de novedades.
	Accede a la página web que contiene la noticia.
Requisitos	Internet
Expectativas	Se abre el navegador y muestra la página web asociada.

Tabla 23: Prueba 1 de la pantalla de novedades

### 5.5 Pruebas de la pantalla de búsquedas

Prueba 1	
Nombre	Campo vacío
Pasos	Entra en la pantalla de búsqueda.
	Pulsa <i>BUSCAR</i> con el campo del código vacío.
	Se muestra mensaje avisando al usuario.
Requisitos	Ninguno
Expectativas	Se muestra mensaje informando al usuario.

Tabla 24: Prueba 1 de la pantalla de búsquedas

Prueba 2	
Nombre	Código incorrecto
Pasos	Entra en la pantalla de búsqueda.
	Pulsa <i>BUSCAR</i> con un código no existente.
	Se muestra mensaje avisando al usuario.
Requisitos	Ninguno
Expectativas	Se muestra mensaje informando al usuario.

Tabla 25: Prueba 2 de la pantalla de búsquedas

Prueba 3	
Nombre	Escáner interrumpido
Pasos	Entra en la pantalla de búsqueda.
	Pulsa <i>Escanear</i> y vuelve a la pantalla anterior con el botón de volver del dispositivo..
	Se muestra mensaje avisando al usuario.
Requisitos	Ninguno
Expectativas	Se muestra mensaje informando al usuario.

Tabla 26: Prueba 3 de la pantalla de búsquedas

Prueba 4	
Nombre	Código existente
Pasos	Entra en la pantalla de búsqueda.
	Pulsa <i>BUSCAR</i> con un código válido.
	Accede a la pantalla de detalles del cómic.
Requisitos	Ninguno
Expectativas	Accede a los detalles del cómic.

Tabla 27: Prueba 4 de la pantalla de búsquedas

## 5.6 Pruebas de la pantalla de nueva tienda

Prueba 1	
Nombre	Campo vacío
Pasos	Entra en la pantalla de nueva tienda.
	Pulsa <i>AÑADIR</i> con los campo de los datos de la tienda vacíos.
	Se muestra mensaje avisando al usuario.
Requisitos	Internet
Expectativas	Se muestra mensaje informando al usuario.

Tabla 28: Prueba 1 de la pantalla de nueva tienda

<b>Prueba 2</b>	
Nombre	Localización no existente
Pasos	Entra en la pantalla de nueva tienda.
	Pulsa <i>AÑADIR</i> con los datos de una localización no existente.
	Se muestra mensaje avisando al usuario.
Requisitos	Internet
Expectativas	Se muestra mensaje informando al usuario.

Tabla 29: Prueba 2 de la pantalla de nueva tienda

<b>Prueba 3</b>	
Nombre	Nombre de tienda existe
Pasos	Entra en la pantalla de nueva tienda.
	Pulsa <i>AÑADIR</i> con un nombre de tienda ya registrado.
	Se muestra mensaje avisando al usuario.
Requisitos	Ninguno
Expectativas	Se muestra mensaje informando al usuario.

Tabla 30: Prueba 3 de la pantalla de nueva tienda

<b>Prueba 4</b>	
Nombre	Nombre y localización válidos
Pasos	Entra en la pantalla de nueva tienda.
	Pulsa <i>AÑADIR</i> con datos válidos.
	Vuelve a la pantalla de tiendas.
Requisitos	Internet
Expectativas	Se muestra la nueva tienda en la pantalla de tiendas.

Tabla 31: Prueba 4 de la pantalla de nueva tienda

## 5.7 Prueba de la pantalla de nuevo comentario

<b>Prueba 1</b>	
Nombre	Campo vacío
Pasos	Entra en la pantalla de nuevo comentario.
	Pulsa <i>ENVIAR</i> con el campo del mensaje vacío.
	Se muestra mensaje avisando al usuario.
Requisitos	Internet
Expectativas	Se muestra mensaje informando al usuario.

Tabla 32: Prueba 1 de la pantalla de nuevo comentario

<b>Prueba 2</b>	
Nombre	Envío con éxito
Pasos	Entra en la pantalla de nuevo comentario.
	Pulsa <i>ENVIAR</i> con un mensaje introducido.
	Vuelve a la pantalla de comentarios.
Requisitos	Internet
Expectativas	Se muestra el comentario al principio de la lista de comentarios.

*Tabla 33: Prueba 2 de la pantalla de nuevo comentario*

## 6. Conclusiones

### 6.1 Aprendizaje

Durante la realización de este trabajo final de máster se han podido poner en práctica muchos de los conocimientos que se han ido obteniendo durante el transcurso del máster. No obstante, la mejor parte ha sido poner en práctica nuevos conocimientos adquiridos durante el desarrollo del trabajo. El uso de arquitecturas como el MVVM no es un tema que se profundice en el máster pero en este periodo de tiempo se ha aprendido a ir haciendo uso de esta arquitectura. Al mismo tiempo se han utilizado herramientas como RxJava o Retrofit en combinación la arquitectura que han permitido diseñar y crear una estructura más ordenada y clara, así como realizar operaciones de forma asíncrona con una mayor eficiencia.

### 6.2 Planificación

Cumplir con la planificación establecida es posiblemente el mayor problema encontrado durante el desarrollo de este proyecto. Durante el período de la PAC 1 se tardó bastante tiempo en tomar una decisión sobre la temática de la aplicación y esto hizo que se perdiera tiempo de otras tareas, aunque se pudo cumplir con lo establecido. No obstante, en los periodos de tiempo de las PACs 2 y 3, debido a motivos laborales y, en ocasiones, personales se ha restado el tiempo fijado en el día a día para el desarrollo de las tareas. Se intentó aumentar las horas establecidas para los días festivos pero no fue suficiente. La PAC2 se pudo entregar más o menos en mínimas condiciones, pero no paso igual con la PAC3.

### 6.3 Resultado final

El resultado de la aplicación, en general, es bastante satisfactorio dentro de lo que cabe, dados los problemas de planificación sufridos. Se han podido generar todas las pantallas que se habían diseñado en la etapa de análisis y diseño y las funcionalidades cumplen mínimamente los requisitos establecidos. No obstante, su rendimiento se podría mejorar bastante. Hay que tener en cuenta, tal y como se ha comentado, que se han utilizado herramientas aprendidas recientemente cuyo uso se puede mejorar claramente para obtener un mejor resultado.

### 6.4 Mejoras de futuro

A medida que se iba desarrollando la aplicación han ido surgiendo bastantes mejoras que implementar. La primera mejora que ha surgido es, sin duda, la de incluir filtros en todas las pantallas que incluyen una lista permitiendo al usuario ver los detalles según sus interés. Por ejemplo, sería interesante poder filtrar

las tiendas por provincia para visualizar solo aquellas que sean cercanas. Otra mejora es buscar los cómics por otras propiedades dado que en estos momentos solo se pueden buscar a través de su código isbn. Sería una buena mejora si se pudiera buscar no solo por nombre sino por autor o género. Por último sería interesante que la app conectará más a los usuarios pudiendo compartir no solo sus opiniones, sino también que pudieran visualizar también las colecciones de otros usuarios, por ejemplo.

# Bibliografía

Android Developers

<https://developer.android.com/>

“Ribas Lequerica, Joan (2014). Desarrollo de aplicaciones para Android (Edición 2015). Madrid: ANAYA MULTIMEDIA S.A.”

“Tornés Gironés, Jesús (2014). El gran libro de Android (4ª Edición). Barcelona: MARCOMBO S.A.”

CLZ Comics - Comic Database

<https://play.google.com/store/apps/details?id=com.collectorz.javamobile.android.comics>

Comic Books Collector

<https://play.google.com/store/apps/details?id=com.sweetlime.cbcollector>

Manga List

<https://play.google.com/store/apps/details?id=be.benoit.sneyers.mangalist>

Métodos aplicables para el desarrollo de aplicaciones móviles

<https://www.genbetadev.com/desarrollo-aplicaciones-moviles/metodos-aplicables-para-el-desarrollo-de-aplicaciones-moviles>

MVVM Compared To MVC and MVP

<http://geekswithblogs.net/dlussier/archive/2009/11/21/136454.aspx>

MVVM architecture, ViewModel and LiveData

<https://proandroiddev.com/mvvm-architecture-viewmodel-and-livedata-part-1-604f50cda1>

Realm Java 5.2.0

<https://realm.io/docs/java/latest>

Using RxJava 2 - Tutorial

<http://www.vogella.com/tutorials/RxJava/article.html>

“Dupree, K. Matt (2015). RxJava for Android App Development. O’Reilly.”

Retrofit – A simple Android Tutorial

[https://medium.com/@prakash\\_pun/retrofit-a-simple-android-tutorial-48437e4e5a23](https://medium.com/@prakash_pun/retrofit-a-simple-android-tutorial-48437e4e5a23)

ZXing ("Zebra Crossing") barcode scanning library for Java, Android  
<https://github.com/zxing/zxing>

Google Maps para cada plataforma  
<https://developers.google.com/maps/?hl=es-419>