

Bookmarker

# Índice de contenido

1.1 Introducción del proyecto.....	5
1.2 Propósito.....	5
1.3 Objetivos del Proyecto.....	5
1.4 Coste del proyecto.....	6
1.4.1-Costes de Desarrollo .....	6
1.4.2 Costes de Implantación.....	7
Capítulo 2: Análisis del sistema.....	7
2.1.-Introducción.....	8
2.2.-Análisis de requisitos.....	8
2.2.1 Requerimientos funcionales.....	8
2.3.-Casos de uso: Diagramas y Narrativas de Casos de Uso.....	9
Capítulo 3: Diseño del sistema.....	14
3.1.-Introducción.....	15
3.3 Diseño de la Base de Datos.....	15
3.4.1.- Diseño Lógico.....	16
3.4.2.- Diseño Relacional.....	16
3.4 Diseño de la Interfaz.....	16
Capítulo 4: Implementación.....	23
4.1 Introducción.....	24
4.2 Arquitectura cliente/servidor.....	24
4.3 Lenguajes de Programación.....	24
4.4 Herramientas de Desarrollo.....	24
4.5 Codificación.....	25
Capítulo 5: Pruebas de software.....	26

<u>5.1.-Introducción.....</u>	<u>27</u>
<u>5.2.- Técnicas de Prueba.....</u>	<u>27</u>
<u>5.2.1.- Pruebas de caja blanca o enfoque estructural.....</u>	<u>27</u>
<u>5.2.2.- Pruebas de caja negra o enfoque funcional.....</u>	<u>27</u>
<u>Capítulo 6: Conclusiones.....</u>	<u>28</u>
<u>6.1.- Conclusiones.....</u>	<u>29</u>
<u>6.2.- Propuestas Futuras (opcional).....</u>	<u>29</u>
<u>Capítulo 7: Bibliografía y referencias.....</u>	<u>30</u>
<u>7.1.-Referencias bibliográficas.....</u>	<u>31</u>
<u>Anexo 1: Manual de Instalación.....</u>	<u>32</u>
<u>Anexo 2: Manual de Usuario.....</u>	<u>33</u>

## CAPÍTULO 1: INTRODUCCIÓN

---

## **1.1 INTRODUCCIÓN DEL PROYECTO**

Bookmarker es un proyecto con la finalidad de crear una herramienta práctica que facilite la búsqueda de libros a través de Internet y permita la organización y gestión de sus lecturas, pudiendo crear marcapáginas para mantener registro de por dónde va leyendo o de momentos importantes en las novelas.

La utilización de esta aplicación puede ayudar a disfrutar más de leer al tener una manera rápida de marcar capítulos que sean de gran importancias para la trama, las páginas en las que se describen la apariencia o forma de pensar de los personajes o tener a mano cuando ocurrieron tus momentos favoritos.

## **1.2 PROPÓSITO**

El proyecto tiene como propósito permitir la búsqueda de libros por Internet, registrarlos de manera local en las categorías que quiera tener el usuario y poder crear marcadores en ellos. También se permitirá crear los libros de manera manual sin necesidad de acceder a Internet.

## **1.3 OBJETIVOS DEL PROYECTO**

Para realizar el proyecto se seguirá el siguiente esquema:.

- 1.- Investigar sobre las tecnologías requeridas para la programación móvil, la obtención de datos de libros y el guardado local en Android.
- 2.- Desarrollar la funcionalidad de búsqueda de libros, que permita acceder una base de datos en línea y recuperar información sobre los libros solicitados.
- 3.- Implementar un sistema de almacenamiento local para guardar la información de los libros en sus dispositivos y acceder a ella sin conexión a Internet.
- 4.- Diseñar y desarrollar una interfaz de usuario que permita a los usuarios navegar, organizar y gestionar su biblioteca personal de libros.

5.- Integrar la función de creación de secciones personalizadas, para que los usuarios puedan organizar sus libros en categorías o temas específicos según sus preferencias.

6.- Implementar la creación de marcadores para los libros que tenga registrado el usuario.

7.- Redactar un manual para el usuario.

8.- Redactar la memoria del proyecto.

## **1.4 COSTE DEL PROYECTO**

Para llevar a cabo el proyecto se tendrán que tener en cuenta los siguientes costes:

### **1.4.1-COSTES DE DESARROLLO**

En el apartado de hardware y software, se necesitará un ordenador en el que se pueda correr Android Studio, para ello, el ordenador deberá tener una arquitectura de CPU de x86\_64 con un procesador Intel Core de segunda generación o un CPU AMD compatible con un hipervisor de Windows y tener como sistema operativo Windows 8, 10 u 11 de 64 bits. También será necesario tener 8 GB de RAM y 8 GB de espacio disponible en disco.

Si se quisiera realizar el proyecto en un sistema operativo distinto a Windows deberá ser un MacOS 10.14 o superior con un chips basado en ARM, o Intel Core de segunda generación (o posterior) compatibles con el Hypervisor.Framework o una distribución de Linux de 64 bits que sea compatible con Gnome, KDE o Unity DE; GNU C Library (glibc) 2.31 o posteriores con un procesador igual que en Windows excepto si se usa un procesador AMD, que deberá ser compatible con AMD Virtualization (AMD-V) y SSSE3.

Aunque Android Studio permita simular un dispositivo móvil, poder comprobar las funcionalidades en un dispositivo real podrá ser de gran ayuda.

En el apartado de costes de personal, al ser un proyecto pequeño no será necesario más de una persona pero deberá tener conocimientos previos de programación en Java o Kotlin.

### **1.4.2 COSTES DE IMPLANTACIÓN**

Para poner en funcionamiento el proyecto solo será necesario un dispositivo Android donde se ejecute la aplicación, ya que se podrán hacer las búsquedas por Internet a bases de datos libres y no hará falta pagar por ninguna licencia, permiso o servicio.

## **CAPÍTULO 2: ANÁLISIS DEL SISTEMA**

---

## **2.1.- INTRODUCCIÓN**

En este capítulo se tratará el análisis de los requisitos, el diseño de la aplicación, la implementación y las pruebas que se han realizado.

## **2.2.- ANÁLISIS DE REQUISITOS**

Bookmarker debe permitir al usuario buscar libros en un repositorio externo dada una consulta para, posteriormente, permitir al usuario escoger el resultado más relevante, guardarla de forma local según su clasificación y crear anotaciones a modo de marcapáginas. También debe permitir el registro de libros de forma manual.

El repositorio externo que se usará es la base de datos de libros de acceso público perteneciente a Internet Archive, Open Library.

El proyecto debe tener una interfaz práctica que siga las convenciones dictadas por Google.

### **2.2.1 REQUERIMIENTOS FUNCIONALES.**

Dado el análisis de requisitos realizado, los requerimientos funcionales del proyecto serán los siguientes:

1. Búsqueda de libros: Los usuarios deben poder realizar búsquedas de libros al repositorio de Open Library.
2. Almacenamiento local de datos: La aplicación debe permitir a los usuarios guardar información sobre libros de manera local para poder acceder a ellos sin necesidad de Internet.
3. Organización de la biblioteca: Los usuarios deben poder crear sus propias secciones y organizar sus libros en ellos.
4. Creación de anotaciones: Los usuarios deben poder crear anotaciones vinculadas a los libros que sirvan como marcapáginas.
5. Registro manual de libros: Los usuarios deben tener la opción de introducir manualmente libros en su base de datos local.

### **2.2.2 REQUERIMIENTOS NO FUNCIONALES**

Los requerimientos no funcionales de la aplicación serán los siguientes:

1. Usabilidad: La aplicación debe tener una interfaz de usuario práctica y que resulte de fácil comprensión.



2. Rendimiento: La aplicación no deberá de sufrir de ralentizaciones, minimizando las posibles esperas.

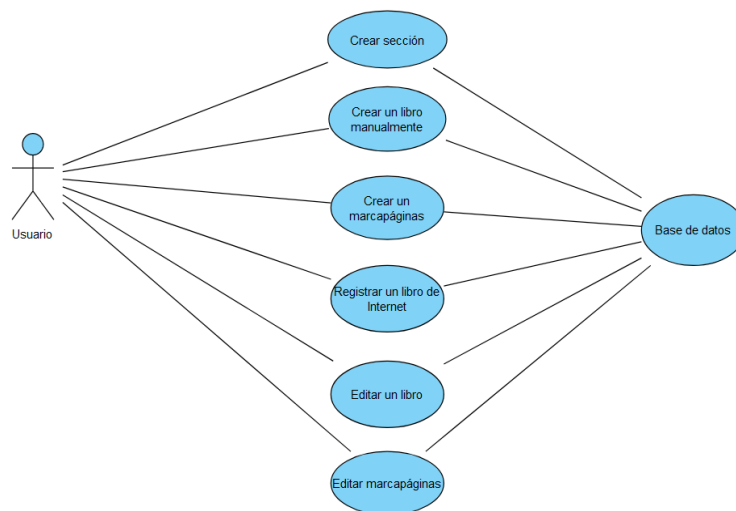
3. Compatibilidad: La aplicación deberá ser compatible con distintas versiones de Android y con distintas pantallas.

4. Estética: La aplicación debe cumplir con las directrices de Google usando las distintas librerías de Material Design.

### 2.3.- CASOS DE USO: DIAGRAMAS Y NARRATIVAS DE CASOS DE USO

Los casos de uso son descripciones de cómo un usuario interactúa con un sistema para lograr un objetivo. Consisten de una secuencia de acciones que describen cómo se realiza la acción y cuál es el resultado esperado. Los casos de uso son utilizados en el análisis y diseño de sistemas para capturar y comunicar los requisitos funcionales del sistema de manera efectiva.

#### DIAGRAMA DE CASO DE USO



#### a) Caso de uso 1: CREAR SECCION.

**Nombre del caso de uso:** Crear sección.

**Actor Principal:** Usuario.

**Condiciones de entrada:** El usuario se encuentra en una vista que permita la creación de secciones (biblioteca, editando libro o creando libro).

**Flujo de eventos:**

1. El usuario selecciona la opción de crear una nueva sección en la barra superior.
2. El sistema muestra un diálogo donde puede introducir el nombre de la nueva sección, junto a un botón de cancelar y un botón de añadir que solo podrá ser seleccionado cuando el campo no esté vacío.
3. El usuario introduce un nombre que no esté vacío y selecciona la opción de crear la sección.
4. El sistema crea una nueva sección y la guarda en la base de datos. El dialogo se cierra.

**Camino alternativo:**

3. b. El usuario selecciona cancelar o selecciona fuera del dialogo. El dialogo se cierra.

**Condiciones de salida:** El dialogo se ha cerrado.

**b) Caso de uso 2: CREAR UN LIBRO MANUALMENTE**

**Nombre del caso de uso:** Crear un libro manualmente.

**Actor Principal:** Usuario.

**Condiciones de entrada:** El usuario se encuentra en biblioteca.

**Flujo de eventos:**

1. El usuario selecciona la opción de crear un nuevo libro en la barra superior.
2. El sistema cambia la vista a la pantalla de crear/editar libros.
3. El usuario introduce los datos que vea apropiado para su libro, teniendo que rellenar al menos el titulo con un titulo no vacío.
4. Al seleccionar guardar, el sistema guarda en la base de datos un nuevo libro con la información que el usuario ha rellenado.

**Camino alternativo:**

3. b. El usuario selecciona la opción de crear una nueva sección. Comienza el caso de uso 1.
4. b Al terminar el caso de uso 1, se sigue desde el paso 2

**Condiciones de salida:** Se ha guardado el nuevo libro o se ha salido de la vista.

**c) Caso de uso 3: CREAR UN MARCAPÁGINAS.**

**Nombre del caso de uso:** Crear un marcapáginas.

**Actor Principal:** Usuario.

**Condiciones de entrada:** El usuario se encuentra en los detalles de un libro local.

**Flujo de eventos:**

1. El usuario selecciona la opción de crear un nuevo marcador en la barra superior.
2. Se genera un nuevo marcador vinculado al libro y se guarda en la base de datos.

**Condiciones de salida:** Se ha guardado el nuevo marcapáginas o se ha salido de la vista.

**d) Caso de uso 4: REGISTRAR UN LIBRO DE INTERNET**

**Nombre del caso de uso:** Crear un libro de Internet.

**Actor Principal:** Usuario.

**Condiciones de entrada:** El usuario se encuentra en buscar.

**Flujo de eventos:**

1. El usuario selecciona la opción de búsqueda de la barra superior y escribe su petición.
2. El sistema realiza una petición a la API de Open Library con la petición del usuario.
3. La API devuelve los resultados que coincidan y el sistema los muestra.
4. Al seleccionar uno, se cambia de vista a la vista de ediciones.
5. El sistema realiza una petición sobre las distintas ediciones que puede tener el libro en Open Library
6. La API devuelve los resultados que coincidan y el sistema los muestra.
7. Al seleccionar uno, se cambia de vista a la vista de ediciones.

8. El usuario introduce los datos que vea apropiado para su libro, teniendo que rellenar al menos el título con un título no vacío.
9. Al seleccionar guardar, el sistema guarda en la base de datos un nuevo libro con la información que el usuario ha rellenado.

**Camino alternativo:**

3. b. No hay coincidencias para la búsqueda y no se muestran las obras. Se vuelve a 1.

6.c No hay coincidencias para la búsqueda y no se muestran las obras. Se vuelve.

**Condiciones de salida:** Se ha guardado el nuevo libro o se ha salido de la vista.

e) **Caso de uso 5: EDITAR UN LIBRO**

**Nombre del caso de uso:** Editar un libro.

**Actor Principal:** Usuario.

**Condiciones de entrada:** El usuario se encuentra en los detalles de un libro local.

**Flujo de eventos:**

1. El usuario selecciona la opción de editar de la barra superior
2. El sistema cambia la pantalla a la vista de editar/crear libro con los datos del libro del que viene introducidos.
3. El usuario realiza los cambios que desea.
4. Selecciona la opción de guardar de la barra superior.
5. El sistema guarda los cambios en la base de datos local con los cambios realizados por el usuario.

**Camino alternativo:**

- 5 b. El usuario ha dejado el campo título vacío. Se muestra un mensaje avisando que no se ha introducido un título. Se vuelve a 3.

**Condiciones de salida:** Se han guardado los cambios al libro o se ha salido de la vista.

f) **Caso de uso 6: EDITAR UN MARCAPÁGINAS**

**Nombre del caso de uso:** Editar un marcapáginas.

**Actor Principal:** Usuario.

**Condiciones de entrada:** El usuario se encuentra en los detalles de un libro local con al menos un marcapáginas.

**Flujo de eventos:**

1. El usuario selecciona la opción de editar del marcapáginas
2. El sistema cambia los campos de texto por entradas de texto y muestra un botón de cancelar y uno de confirmar.
3. El usuario realiza los cambios que desea.
4. Selecciona la opción de confirmar.
5. El sistema guarda los cambios en la base de datos local con los cambios realizados por el usuario.

**Camino alternativo:**

- 3 b. El usuario selecciona el botón de cancelar.

**Condiciones de salida:** Se han guardado los cambios al marcapáginas, se ha cancelado la modificación o se ha salido de la vista.

## CAPÍTULO 3: DISEÑO DEL SISTEMA

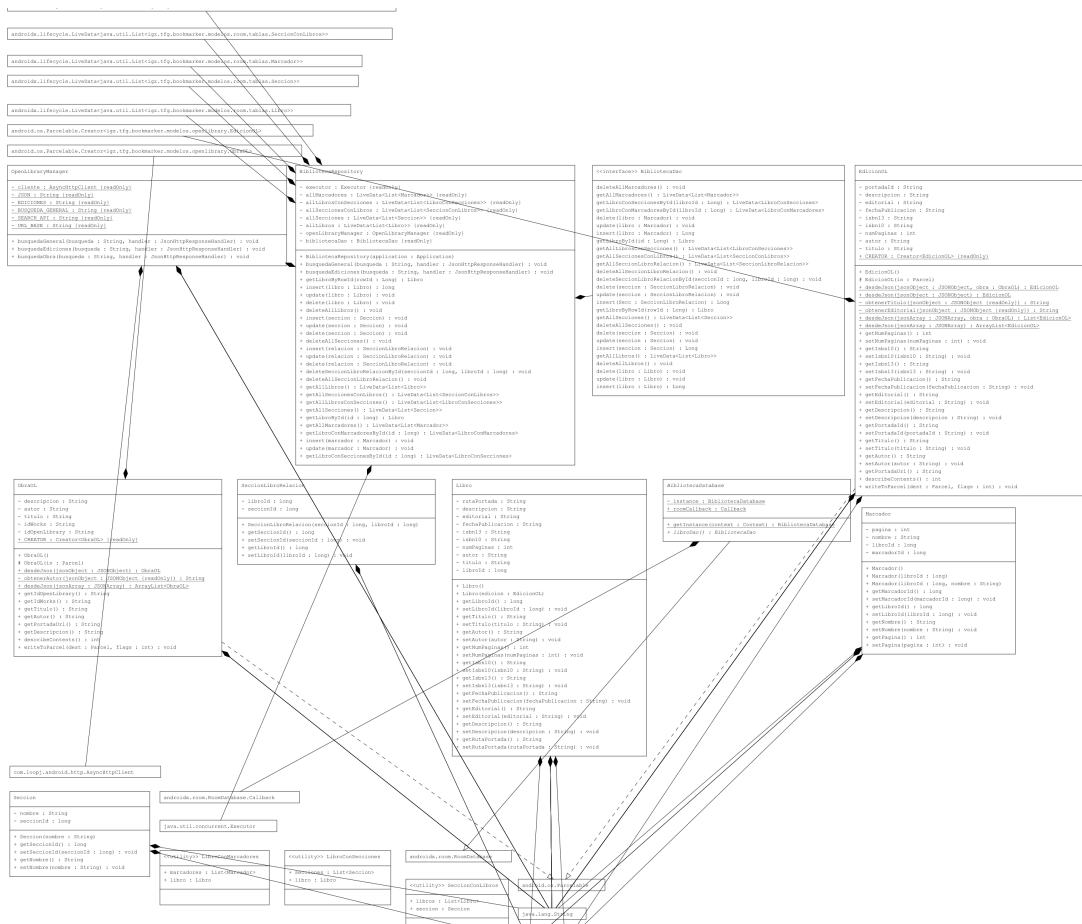
---

### 3.1.- INTRODUCCIÓN.

En los siguientes apartados se podrá ver como se ha diseñado la solución a los requisitos anteriores cumpliendo los casos de uso.

### 3.2 DIAGRAMA DE CLASES.

Debido a la complejidad de un proyecto Android, a continuación se mostrará el diagrama de clases incluyendo solo los modelos y procesos que intervienen en la base de datos y en la consulta a Open Library. Adjunto como imagen separada el diagrama de clases con todas las clases.



### 3.3 DISEÑO DE LA BASE DE DATOS.

La base de datos local del proyecto es una base de datos de SQLite con la siguiente estructura.

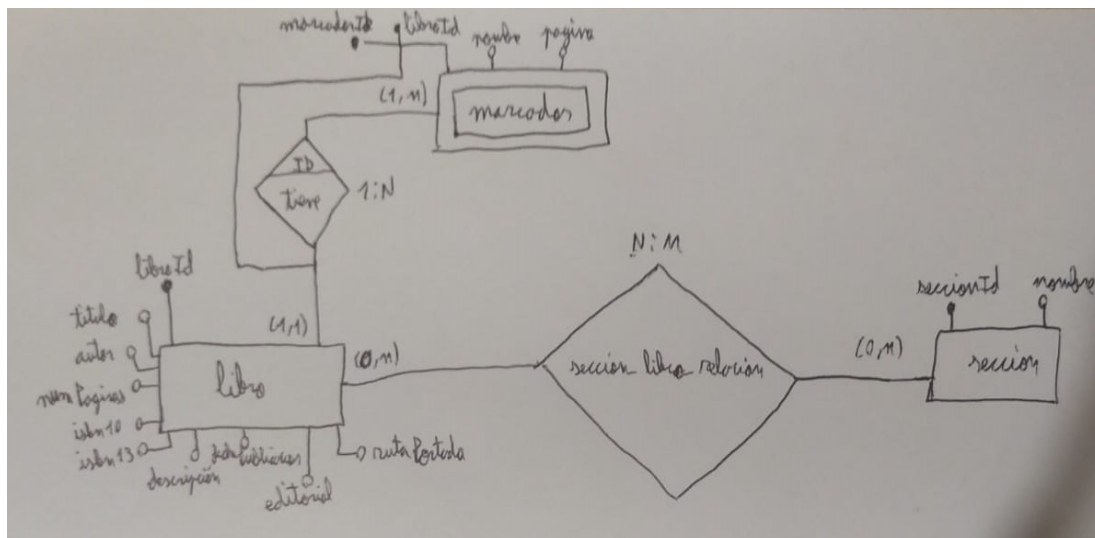
### 3.4.1.- DISEÑO LÓGICO.

Esquema lógico de la base de datos:



### 3.4.2.- DISEÑO RELACIONAL.

Esquema relacional de la base de datos:



### 3.4 DISEÑO DE LA INTERFAZ.

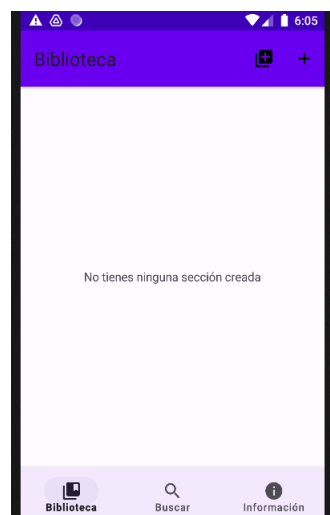
La interfaz de Bookmarker está constituida por una vista principal que realiza la función de contenedor al resto de pantallas. En esta pantalla principal se encuentra

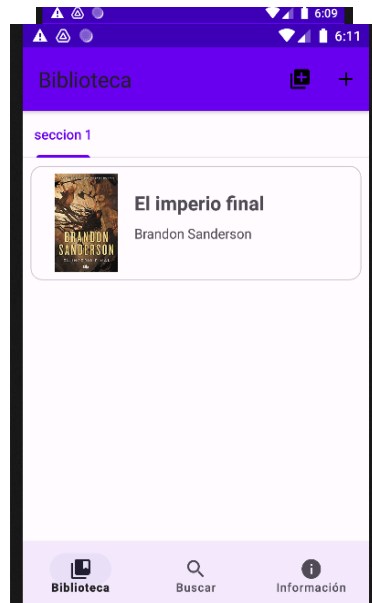


la barra de tareas en la zona superior y una barra de navegación con 3 elementos, estos elementos representan las principales vistas de la aplicación.

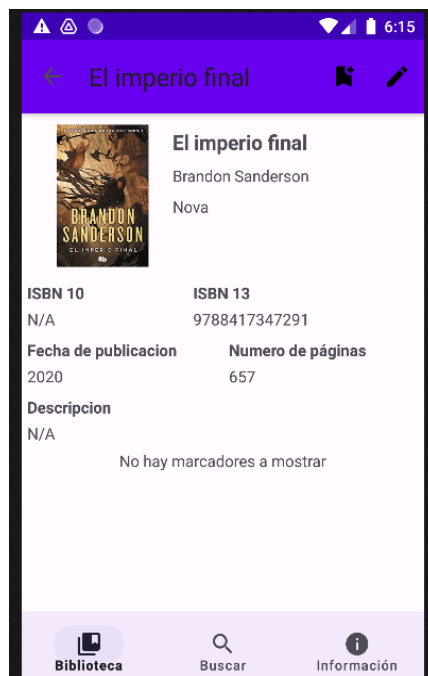


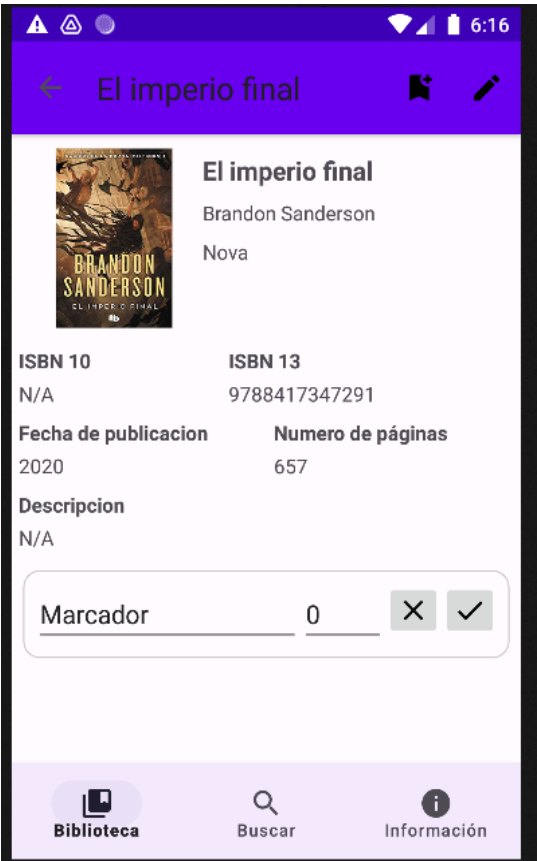
La primera opción del menú corresponde con la pantalla de biblioteca. Cuando no hay ninguna sección creada se muestra un mensaje informando de ello, si hay secciones se mostrará una barra con las distintas secciones. A su vez, si no hay libros registrados en esa sección se mostrará un mensaje, en caso contrario se mostrarán los libros. Al estar en esta pantalla aparecen dos botones superiores con los que se pueden crear secciones y libros.



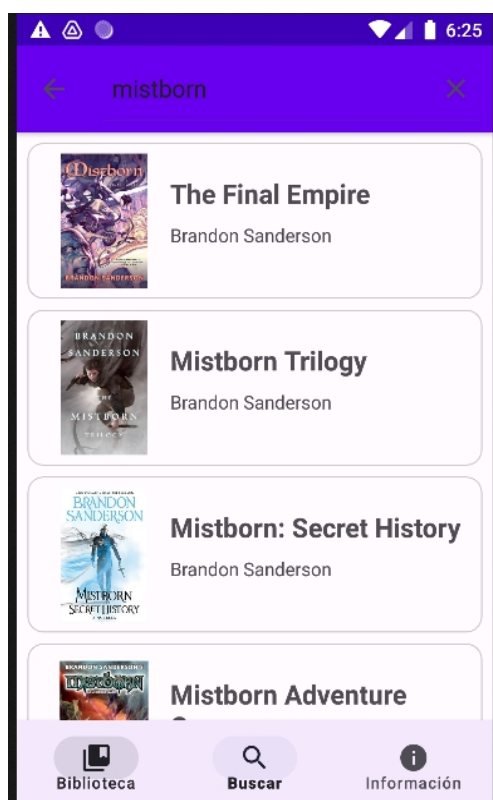
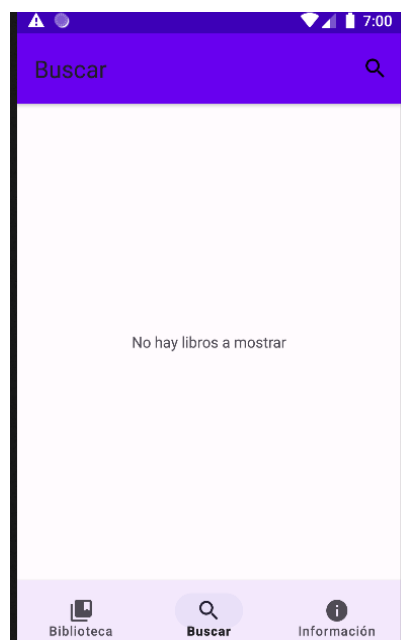


Si seleccionamos un libro de la sección nos enviará a su página de detalles donde se muestran sus datos y los marcadores en el caso de que tenga alguno creado. Los botones de la parte superior en esta pestaña permiten crear un marcador nuevo y editar el libro. Seleccionando el botón de editar marcador podremos cambiar sus datos.

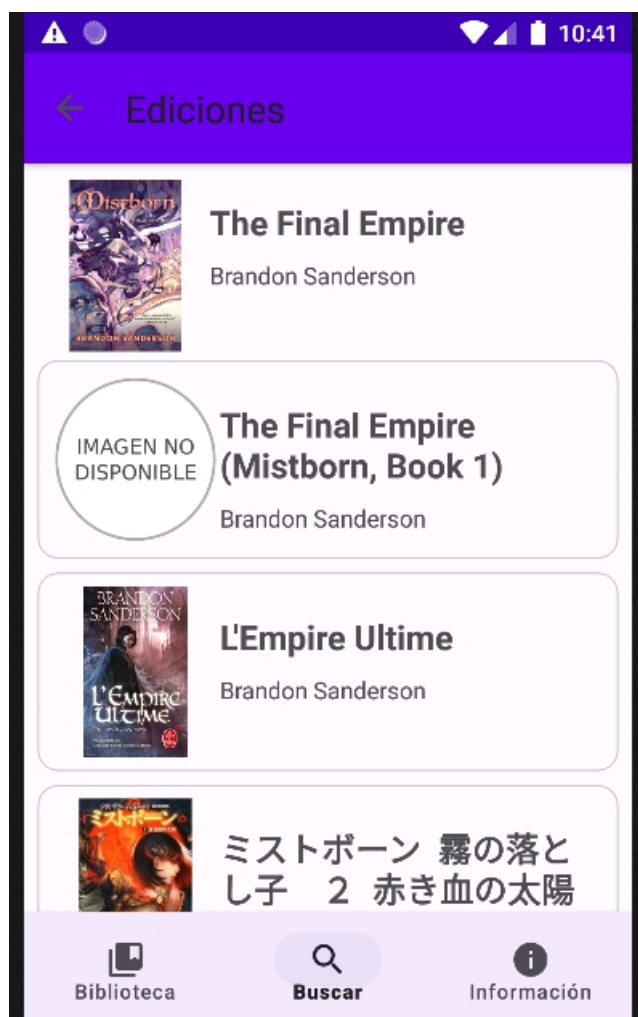




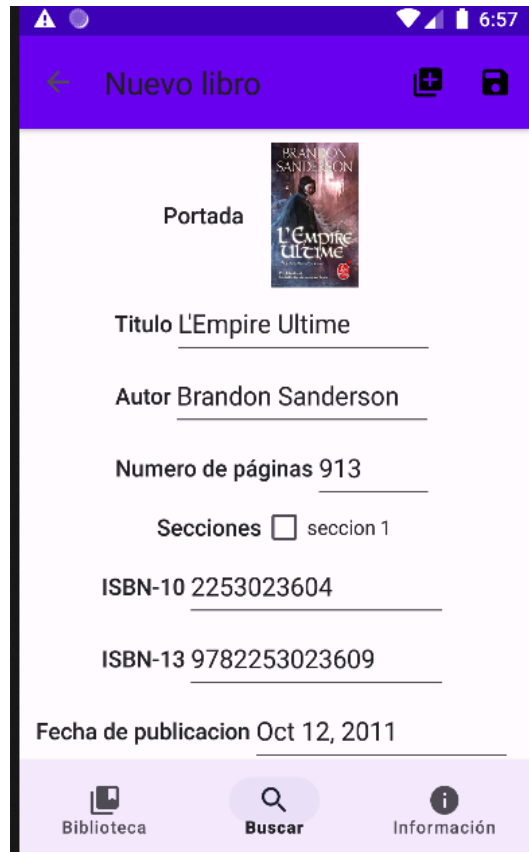
Al seleccionar la segunda opción del menú inferior podemos dirigirnos a las búsquedas de Open Library. Una vez hayamos realizado una búsqueda al pulsar la opción superior se mostrarán los resultados relevantes.



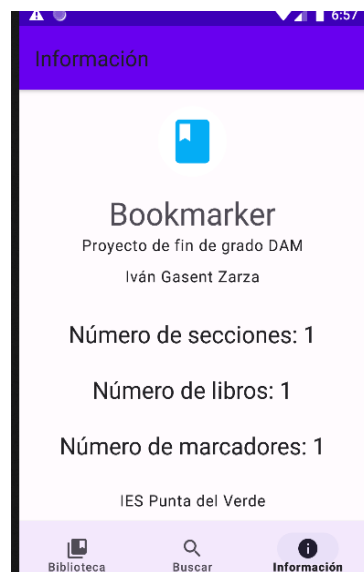
Si seleccionamos uno de los resultados de la búsqueda nos enviará a una pantalla con la información sobre el libro y las distintas ediciones que hay. Al hacer click en una de ellas nos enviará a la pantalla de crear libro.



Tanto si hemos decidido crear una nueva entrada local, si estamos editando un libro ya existente o estamos registrando uno desde Internet, llegaremos a la pantalla de creación/edición de libro. Esta pantalla permite editar los datos del libro y se ha llegado desde una búsqueda o desde un libro existente, los campos se rellenarán automáticamente. Las dos opciones superiores permiten crear una sección nueva y guardar en la base de datos.



La tercera opción del menú inferior lleva a una pantalla con información sobre la aplicación.



## **CAPÍTULO 4: IMPLEMENTACIÓN**

---

#### **4.1 INTRODUCCIÓN.**

Ahora que ya se tienen planteados el problema y la solución, el siguiente paso es convertir las ideas de las actividades anteriores en código fuente.

#### **4.2 ARQUITECTURA CLIENTE/SERVIDOR**

La arquitectura que se usa en el proyecto es una arquitectura basada en el patrón de diseño Model View ViewModel (MVVM) enfocado a Android. En esta arquitectura la vista es la capa encargada de mostrar la interfaz de usuario al usuario, siendo las interfaces las que hemos visto en el apartado anterior.

El modelo se encarga de obtener los datos y realizar la lógica de negocio, estando en esta capa el repositorio, que se encarga de hacer las peticiones tanto a la base de datos local como a la base de datos externa mediante peticiones por internet recibiendo el resultado en formato json.

El ViewModel hace de intermediario entre el modelo y la vista. Además, el ViewModel es consciente del ciclo de vida de la vista por lo que puede conservar los datos hasta ser destruido el fragmento, solucionando algunos problemas que se podían encontrar con otras arquitecturas.

#### **4.3 LENGUAJES DE PROGRAMACIÓN.**

El código de Bookmark se ha desarrollado en Java, un lenguaje de programación orientado a objetos con el que tengo experiencia. Al ser uno de los principales lenguajes utilizados para el desarrollo de aplicaciones en Android, Java tiene una gran biblioteca de herramientas que facilitan la creación de aplicaciones, tanto en escritorio como en Android. Debido a su popularidad también dispone de gran cantidad de recursos en la web, aunque encontrar documentación específica para problemas recientes puede ser difícil por el gran soporte que se le da a Kotlin sobre Java.

#### **4.4 HERRAMIENTAS DE DESARROLLO.**

En el desarrollo de Bookmark en Java, se utilizó Android Studio en el sistema operativo Windows 10. Android Studio, es la principal herramienta recomendada por Google para el desarrollo de aplicaciones Android y dispone de características específicas para el desarrollo móvil como puede ser la emulación de dispositivos, las ayudas para la edición de las interfaces o gráficos para visualizar las navegaciones en la aplicación.



#### **4.5 CODIFICACIÓN.**

Todo el código desarrollado se encuentra disponible en el la carpeta compartida que se adjunta con esta memoria.

## CAPÍTULO 5: PRUEBAS DE SOFTWARE

---

## **5.1.-INTRODUCCIÓN.**

Las pruebas de software son actividades necesarias en el desarrollo de una aplicación. Estas pruebas permiten al desarrollador asegurarse de que la aplicación funcione correctamente, cumpla con los requisitos establecidos y se comporte de acuerdo a las expectativas del usuario.

## **5.2.- TÉCNICAS DE PRUEBA.**

Hay dos tipos principales de pruebas de software, las pruebas de caja blanca y de caja negra.

Las pruebas de caja blanca consisten en comprobar el funcionamiento correcto de los procesos, asegurándose que realizan todas las operaciones necesarias en el orden correcto.

Las pruebas de caja negra tratan de verificar las expresiones según su salida. En estas se comprueba que dado un valor, se reciba un resultado en concreto sin importar el proceso que tome.

### **5.2.1.- PRUEBAS DE CAJA BLANCA O ENFOQUE ESTRUCTURAL.**

Debido a la estructura de este proyecto, se ha optado por hacer comprobaciones manuales a los distintos componentes durante su desarrollo, validando el correcto funcionamiento en distintos casos. Estas pruebas no solo han servido para comprobar el funcionamiento esperado de los distintos componentes, si no que también han resultado una buena forma de entender los ciclos de vida de los fragmentos y las actividades.

### **5.2.2.- PRUEBAS DE CAJA NEGRA O ENFOQUE FUNCIONAL.**

Dado que Bookmarker no tiene ningún método del que se dependa donde se recibe un dato y se espera una salida concreta excepto por las operaciones de la base de datos, se ha optado por no realizar ningún test automatizado de caja negra.

## **CAPÍTULO 6: CONCLUSIONES.**

---

## **6.1.- CONCLUSIONES**

Bookmarker es un proyecto con una base que permite muchas ampliaciones y gracias su arquitectura no supondrá grandes problemas.

## **6.2.- PROPUESTAS FUTURAS (OPCIONAL).**

Algunas ampliaciones que se pueden realizar son:

- Añadir tipos distintos de marcadores: anotaciones de texto, marcapáginas con la posibilidad de añadir textos, notas de voz, una valoración o un marcador al que se le pueda añadir imágenes.
- Cambios de idioma.
- Mejorar las operaciones CRUD.
- Añadir un apartado social, poder recomendar libros o recibir recomendaciones.
- Añadir distintos repositorios desde los que buscar libros.
- Mejorar la búsqueda añadiendo filtros, búsquedas por ISBN, editorial u autor.
- Añadir una base de datos en la nube donde tener una copia de seguridad.
- Añadir lector de libros.

## **CAPÍTULO 7: BIBLIOGRAFÍA Y REFERENCIAS**

---

## 7.1.-REFERENCIAS BIBLIOGRÁFICAS.

En este apartado se citan las referencias bibliográficas, los artículos electrónicos y las páginas Web utilizadas en el proyecto.

API de Open Library: <https://openlibrary.org/developers/api>

Librería de peticiones asíncronas de HTTP: <https://loopj.com/android-async-http/>

Picasso: <https://square.github.io/picasso/>

Material Design: <https://m3.material.io/get-started>

Room (librería de la BDD): <https://developer.android.com/training/data-storage/room/accessing-data>

Base para la búsqueda en Open Library: <https://github.com/silicaRich/Android-Book-Search>

Guía de instalación de Android Studio: <https://developer.android.com/codelabs/basic-android-kotlin-compose-install-android-studio?hl=es-419#0>

## **ANEXO 1: MANUAL DE INSTALACIÓN**

---

Para poder instalar la aplicación deberemos habilitar en nuestro dispositivo la instalación de aplicaciones desconocidas. Esta opción se encuentra en los ajustes, pero dependiendo de tu versión de Android se encontrará en un lugar distinto.

Una vez permitida la instalación de aplicaciones desconocidas, podremos instalar Bookmarker si tenemos la APK en nuestro dispositivo e interactuamos con ella desde nuestro gestor de archivos.

Para generar el APK se necesitará Android Studio y el código fuente de la aplicación. Una vez tengamos el proyecto, podemos crear el APK desde Build > Build Bundle(s)/APK(s) > Build APK. Por defecto se creará en app / build / outputs / apk / debug con el nombre app-debug.apk.



## **ANEXO 2: MANUAL DE USUARIO**

---

En Bookmarker solo tienes que tener en cuenta que para poder ver un libro, este debe de estar asociado a una sección. Si no tienes ninguna sección puedes crearla tanto en biblioteca como en la vista de creación/edición de libros con el botón de crear secciones en la barra superior.

Para buscar libros solo tienes que introducir el nombre en la pantalla Buscar, elegir la obra y luego la edición a guardar. Una vez tienes la edición que quieres guardar, puedes editar el título, la imagen de portada, los ISBNs...