

eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Ingeniería Informática de Gestión y Sistemas de Información

Desarrollo Avanzado de Software

LibreBook

Estudiante:

Gabiña Barañano, Xabier

22 de abril de 2025

Índice general

1. Introducción	3
2. Objetivos	4
2.1. Elementos obligatorios	4
2.2. Elementos opcionales	4
3. Descripción de la aplicación	6
3.1. Clases	6
3.2. Base de datos	7
4. Manual de usuario	8
5. Dificultades	9
5.1. Base de datos remota	9
5.2. Servicio de tiempo de lectura	9
6. Conclusiones	10

Índice de figuras

3.1. Diagrama de clases de la aplicación LibreBook	6
3.2. Diagrama de la base de datos de la aplicación LibreBook	7

1. Introducción

Al igual que en la primera entrega, LibreBook es una aplicación móvil para Android diseñada para los amantes de la lectura. Es una plataforma que permite a los usuarios crear una biblioteca personal digital donde pueden registrar, organizar y seguir su progreso en los libros que están leyendo o desean leer.

La aplicación está desarrollada completamente en Java y sigue las mejores prácticas de desarrollo para Android, incluyendo el uso de Room para la persistencia de datos[2], arquitectura MVVM[1], y componentes de la biblioteca de Material Design[3] para ofrecer una interfaz moderna y funcional.

El repositorio de la aplicación se encuentra en GitHub y está disponible para su descarga y uso bajo la licencia MIT.

[Repositorio de la aplicación](#)

Además en el repositorio, también está disponible en GitHub el binario de la aplicación en formato apk para su descarga e instalación en dispositivos Android.

[Archivo APK de la aplicación](#)

En la aplicación existen por defecto únicamente 10 libros.

- | | |
|---------------------------|---------------------------|
| 1. Crimen y Castigo | 6. Los Demonios |
| 2. Los Hermanos Karamazov | 7. Humillados y Ofendidos |
| 3. El Idiota | 8. El eterno marido |
| 4. Memorias del Subsuelo | 9. Noches Blancas |
| 5. El Jugador | 10. El doble |

Todos de Dostoievski. Tenlo en cuenta a la hora de buscar los libros ya que no se pueden añadir libros directamente desde la aplicación.

La aplicación cuenta con dos cuentas de usuario por defectos:

- **Administrador**

- Email: admin@xabierland.com
- Contraseña: admin

- **Xabier**

- Email: xabierland@gmail.com
- Contraseña: 123456

No obstante, se pueden crear nuevas cuentas de usuario mediante el registro en la aplicación.

2. Objetivos

2.1. Elementos obligatorios

- **Uso de una base de datos remota para el registro y la identificación de usuarios mediante registro.**
 - He movido la base de datos de local a remota.
 - La definición de la base de datos se puede encontrar en `/sql/librebook_db.sql`
 - También se ha implementado una API REST en PHP para que la aplicación pueda comunicarse con la base de datos `/php/api.php`
- **Integrar los servicios Google Maps y Open Street Map y Geolocalización en una actividad.**
 - He implementado Open Street Map en la actividad de
- **Captar imágenes desde la cámara, guardarlas en el servidor y mostrarlas en la aplicación. Por ejemplo, una foto de perfil.**

2.2. Elementos opcionales

- **Uso de algún Content Provider para añadir, modificar o eliminar datos.**
 - Dado que el uso de Content Provider es el de permitir el acceso a los datos de la aplicación desde otras aplicaciones no he usado el Content Provider para añadir, modificar o eliminar datos de la aplicación si no en la base de datos propia del Content Provider. No me queda claro si este era el objetivo del ejercicio en base al enunciado...
 - El uso de Content Provider se ha implementado en las actividades de (ProfileActivity.java y BookActivity.java).
 - En el ProfileActivity.java se ha implementado al mantener presionado un libro lo que abre un dialogo para compartirlos en otras aplicaciones bien en forma de texto plano o de archivo.
 - En el BookActivity.java se ha implementado un boton de compartir en menu de opciones que permite compartir el libro en otras aplicaciones de la misma forma que la anterior.
- **Implementación de un servicio en primer plano y gestión de mensajes broadcast durante el servicio**
 -
- **Uso de mensajería FCM. Se debe incluir alguna forma de que se pueda probar de forma externa (por ejemplo, con un servicio web PHP adicional en el servidor de la asignatura).**
 - La logica para implementar la mensajería FCM se encuentra en (LibreBookMessagingService.java).

- Para hacer uso de la mensajería FCM accede a [esta web](#) y escribe el mensaje y el título que se mostraran como notificación.
- Desarrollar un widget que tenga, al menos, un elemento que se actualice automáticamente de manera periódica.
- Uso de algún servicio o tarea programada mediante alarma (no valen las alarmas del widget).

3. Descripción de la aplicación

3.1. Classes

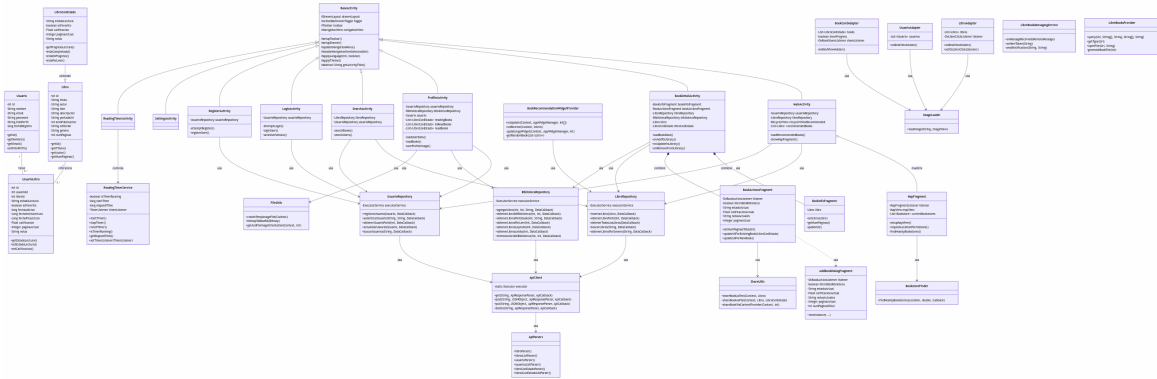


Figura 3.1: Diagrama de clases de la aplicación LibreBook

3.2. Base de datos

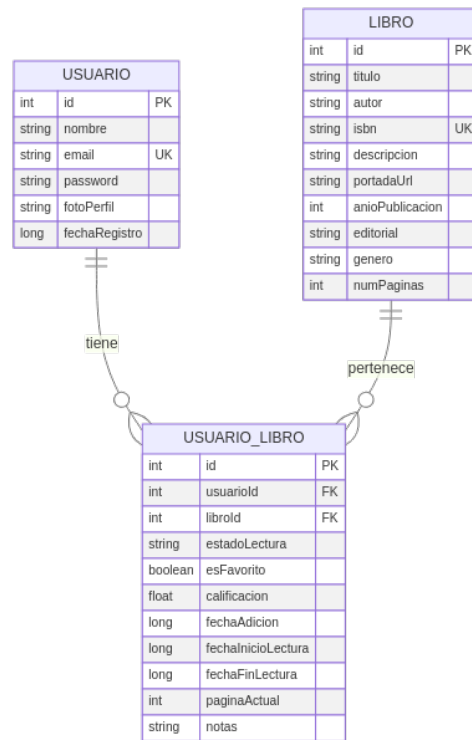


Figura 3.2: Diagrama de la base de datos de la aplicación LibreBook

La base de datos no ha sufrido cambios respecto a la entrada anterior.

4. Manual de usuario

5. Dificultades

5.1. Base de datos remota

Mover la base de datos de local a remota ha supuesto un reto importante. Primero al usar la arquitectura Room no tenia unas sentencias SQL definidas para la estructura de la base de datos por lo que tuve que crearla de cero. Ademas, era necesario el uso de una API REST para poder interactuar con la base de datos.

5.2. Servicio de tiempo de lectura

Implementar el servicio de lectura tubo sus dificultades especialmente la parte de broadcast. Yo queria implementar una notificacion que constantemente se estuviera actualizando para que estando fuera de la aplicacion y de actividad fuera posible saber el tiempo de lectura e incluso detenerlo. Lo primero que intente fue usar un WorkManager y poner el PeriodicWorkRequest a 1 segundo pero Android pone que el tiempo minimo es de 15 minutos por lo que tuve que reescribir el servicio u usar un ForegroundService. Esto supuso reescribir el servicio y la actividad de tiempo de lectura completamente.

6. Conclusiones

Aunque esta entrega haya sido más corta en cuanto a funcionalidades se me ha hecho bastante más complejas de implementar.

Bibliografía

- [1] Microsoft. *The MVVM Pattern*. 2012. URL: [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/hh848246\(v=pandp.10\)](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/hh848246(v=pandp.10)).
- [2] Android Developers. *Room Persistence Library*. 2023. URL: <https://developer.android.com/training/data-storage/room>.
- [3] Google. *Material Design*. 2023. URL: <https://material.io/design>.