

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

16 de marzo de 2025

Índice general

| | |
|---|-----------|
| 1. Introducción | 3 |
| 2. Objetivos | 4 |
| 2.1. Elementos obligatorios | 4 |
| 2.2. Elementos opcionales | 5 |
| 3. Descripción de la aplicación | 7 |
| 3.1. Clases | 7 |
| 3.1.1. Activities | 7 |
| 3.1.2. Fragments | 7 |
| 3.1.3. Adaptadores | 7 |
| 3.1.4. Modelos y Entidades | 8 |
| 3.1.5. Data Access Objects (DAOs) | 8 |
| 3.1.6. Repositorios | 8 |
| 3.1.7. Utilidades | 8 |
| 3.2. Base de datos | 9 |
| 3.2.1. Estructura de la base de datos | 9 |
| 3.2.2. Operaciones principales | 10 |
| 3.2.3. Inicialización y precarga | 10 |
| 3.2.4. Transacciones y concurrencia | 10 |
| 4. Manual de usuario | 11 |
| 4.1. Pantalla de Inicio | 11 |
| 4.2. Menú de Navegación Lateral | 12 |
| 4.3. Creación de Cuenta e Inicio de Sesión | 13 |
| 4.3.1. Registro de Nueva Cuenta | 13 |
| 4.3.2. Inicio de Sesión | 14 |
| 4.4. Perfil de Usuario y Biblioteca Personal | 15 |
| 4.4.1. Cambiar Foto de Perfil | 15 |
| 4.5. Búsqueda de Libros y Usuarios | 16 |
| 4.6. Detalles de Libro y Gestión de Biblioteca | 17 |
| 4.6.1. Añadir Libro a la Biblioteca | 18 |
| 4.7. Ajustes de la Aplicación | 19 |
| 4.7.1. Configuración de idioma | 19 |
| 4.7.2. Configuración de tema | 19 |
| 4.8. Cierre de Sesión | 20 |
| 5. Dificultades | 21 |
| 5.1. Implementación de Room y arquitectura de datos | 21 |
| 5.2. Manejo de multi-idioma y temas | 21 |
| 5.3. Concurrencia y operaciones asíncronas | 21 |
| 6. Conclusiones | 22 |

Índice de figuras

| | |
|--|----|
| 3.1. Diagrama de clases de la aplicación LibreBook | 7 |
| 3.2. Diagrama de la base de datos de la aplicación LibreBook | 9 |
| 4.1. Pantalla de inicio con recomendaciones de libros | 11 |
| 4.2. Menú para usuario anónimo (izquierda) y usuario autenticado (derecha) | 12 |
| 4.3. Pantalla de registro con formulario de datos (izquierda) y notificación de bienvenida (derecha) . . | 13 |
| 4.4. Pantalla de inicio de sesión | 14 |
| 4.5. Pantalla de perfil de usuario con biblioteca organizada | 15 |
| 4.6. Búsqueda de libros (izquierda) y usuarios (derecha) | 16 |
| 4.7. Pantalla de detalle de libro con información completa (izquierda) y vista horizontal (derecha) . . | 17 |
| 4.8. Diálogo para añadir libros | 18 |
| 4.9. Pantalla de ajustes con opciones de personalización | 19 |
| 4.10. Diálogo de confirmación de cierre de sesión | 20 |

1. Introducción

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[7], arquitectura MVVM[1], y componentes de la biblioteca de Material Design[8] 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 ListView+CardView personalizado o de RecyclerView+CardView para mostrar listados de elementos con diferentes características.**
 - En la actividad principal (MainActivity.java) se muestra una lista de cuatro libros aleatorios de la base de datos mediante RecyclerView.
 - En la actividad de búsqueda (SearchActivity.java), dependiendo de la posición del switch, se muestra una lista de libros o de usuarios usando el mismo RecyclerView.
 - En la actividad del perfil (ProfileActivity.java), se muestran tres listas horizontales de libros separadas por categoría (leyendo, por leer, leídos) mediante RecyclerView.
 - Todas estas RecyclerViews usan un adaptador personalizado (BookCardAdapter.java, UsuarioAdapter, LibroAdapter) para mostrar los CardViews con la información de cada elemento.
- **Usar una base de datos local, para listar, añadir y modificar elementos y características de cada elemento.**
 - Se implementa Room para gestionar la base de datos SQLite (AppDatabase.java).
 - Se han creado entidades como Libro.java, Usuario.java y UsuarioLibro.java con sus respectivas relaciones.
 - Los DAOs correspondientes (LibroDao.java, UsuarioDao.java, UsuarioLibroDao.java) contienen métodos para insertar, actualizar, eliminar y consultar datos.
 - Los repositorios (LibroRepository.java, UsuarioRepository.java, BibliotecaRepository.java) proporcionan una capa adicional de abstracción para el acceso a datos.
- **Uso de diálogos.**
 - En SettingsActivity.java se implementan diálogos para cambiar el tema y el idioma de la aplicación.
 - En BookDetailActivity.java se utiliza un diálogo personalizado (dialog_add_book.xml) para añadir libros a la biblioteca y elegir su estado (leído, leyendo, por leer).
 - En BaseActivity.java se muestra un diálogo de confirmación al querer cerrar sesión.
 - En ProfileActivity.java se implementa un diálogo para solicitar permisos de acceso a la galería.
 - Se utilizan diálogos de carga (dialog_loading.xml) durante las operaciones de búsqueda que puedan llevar tiempo.
- **Usar notificaciones locales.**
 - En RegisterActivity.java, tras el registro exitoso de un usuario, se muestra una notificación de bienvenida[3].
 - La clase NotificationUtils.java implementa métodos para crear el canal de notificaciones y mostrar la notificación.
- **Control de la pila de actividades.**
 - En BaseActivity.java, método handleNavigationItemSelected(), se controla la navegación entre actividades.
 - Se utilizan flags como FLAG_ACTIVITY_CLEAR_TOP y FLAG_ACTIVITY_SINGLE_TOP para gestionar la pila de actividades.
 - Se comprueba si ya estamos en la actividad a la que queremos ir para evitar crear instancias innecesarias.
 - En los métodos setLanguageMode() y setThemeMode() se reinicia la pila de actividades para aplicar los cambios globalmente.
 - En métodos como logoutUser() se controla correctamente el regreso a la actividad principal.

2.2. Elementos opcionales

- **Permitir que una misma funcionalidad se comporte de manera distinta dependiendo de la orientación (o del tamaño) del dispositivo mediante el uso de Fragments.**
 - En la actividad de detalle del libro (BookDetailActivity.java) se utilizan dos fragments: BookInfoFragment y BookActionsFragment[4].
 - En orientación vertical (layout activity_book_detail.xml), los fragments se muestran uno sobre otro.
 - En orientación horizontal (layout-land/activity_book_detail.xml), los fragments se muestran uno al lado del otro, aprovechando mejor el espacio horizontal.
 - También el diálogo de añadir un libro a la biblioteca es un fragment el cual además almacena el estado del libro en caso de llamadas o rotaciones.
 - Los fragments permiten una mejor separación de responsabilidades: BookInfoFragment muestra la información detallada del libro, mientras que BookActionsFragment maneja las acciones que el usuario puede realizar.
- **Hacer la aplicación multiidioma y añadir la opción de cambiar de idioma en la propia aplicación.**
 - La aplicación soporta tres idiomas: inglés (values/strings.xml), español (values-es/strings.xml) y euskera (values-eu/strings.xml)[6].
 - En SettingsActivity.java se implementa la opción para cambiar el idioma mediante un diálogo.
 - En BaseActivity.java, los métodos loadLocale(), applyLanguage() y setLanguageMode() gestionan el cambio de idioma.
 - El idioma seleccionado se guarda en SharedPreferences y se aplica en toda la aplicación.
- **Uso de ficheros de texto.**
 - En ProfileActivity.java se implementa la funcionalidad para guardar y cargar imágenes de perfil como archivos en el almacenamiento interno.
 - En el método saveProfileImage() se guarda la imagen de perfil seleccionada como un archivo JPEG.
 - En DatabaseInitializer.java se crean archivos de imagen para los usuarios predeterminados.
 - En el método updateNavigationHeader() de BaseActivity.java se cargan las imágenes de perfil desde archivos.
- **Uso de Preferencias, para guardar las preferencias del usuario en cuanto a mostrar/esconder cierta información, elegir colores para la aplicación, o cualquier otra cosa relacionada con la visualización de la aplicación.**
 - En BaseActivity.java se guardan preferencias para el idioma y el tema seleccionados.
 - En LoginActivity.java y RegisterActivity.java se guardan datos de sesión del usuario.
 - Las preferencias permiten persistir la configuración elegida por el usuario entre sesiones.
- **Crear estilos y temas propios, para personalizar fondos, botones, etc.**
 - Se han definido temas personalizados en values/themes.xml y values-night/themes.xml[9].
 - Se han creado estilos específicos para el modo claro (Theme.LibreBook.Light) y el modo oscuro (Theme.LibreBook.Dark).
 - Se han definido atributos personalizados en attrs.xml como backgroundColor, cardBackgroundColor y textColor.
 - Se utilizan drawables personalizados como rounded_background.xml para dar estilo a elementos visuales.
 - Los temas aplican colores consistentes a lo largo de toda la aplicación.
- **Usar intents implícitos para abrir otras aplicaciones, contactos, etc.**
 - En ProfileActivity.java se utiliza un intent implícito para abrir la galería de imágenes (ACTION_PICK).
 - En el método onRequestPermissionsResult() se utiliza un intent implícito para abrir los ajustes de la aplicación cuando se deniegan permisos.

- El intent `Intent(android.provider.Settings.ACTION_APPLICATION_DETAILS_SETTINGS)` permite al usuario ir directamente a los ajustes de permisos de la aplicación.
 - En `ImageLoader.java` se accede implícitamente a recursos de internet para cargar imágenes desde URLs.
- **Añadir una barra de herramientas (ToolBar) personalizada en la aplicación así como un panel de navegación (Navigation Drawer)**
- En `BaseActivity.java` se implementa un `ToolBar` personalizado que se hereda en todas las actividades.
 - El método `setupToolBar()` configura la barra de herramientas con el título específico de cada actividad.
 - Se implementa un `Navigation Drawer` (panel lateral de navegación) en el método `setupDrawer()`.
 - El `Navigation Drawer` incluye un encabezado personalizado (`nav_header_main.xml`) que muestra información del usuario.
 - El menú del `Navigation Drawer` (`drawer_menu.xml`) incluye opciones diferentes según el estado de autenticación del usuario.

3.1.4. Modelos y Entidades

- **Libro:** Entidad que representa un libro con sus atributos (título, autor, ISBN, descripción, etc.).
- **Usuario:** Entidad que representa un usuario de la aplicación (nombre, email, contraseña, foto de perfil).
- **UsuarioLibro:** Entidad de relación entre usuarios y libros que almacena estado de lectura, calificaciones, notas, etc.
- **LibroConEstado:** Clase POJO (Plain Old Java Object) que combina información de un libro y su estado en la biblioteca de un usuario.

3.1.5. Data Access Objects (DAOs)

- **LibroDao:** Interfaz que define operaciones de acceso a datos para la entidad Libro.
- **UsuarioDao:** Interfaz para operaciones de acceso a datos relacionadas con usuarios.
- **UsuarioLibroDao:** Interfaz para operaciones de acceso a datos de la relación usuario-libro.

3.1.6. Repositorios

- **LibroRepository:** Encapsula la lógica de acceso a datos para libros, proporcionando un API limpia a las actividades.
- **UsuarioRepository:** Gestiona el acceso a datos para usuarios, incluyendo operaciones como registro y autenticación.
- **BibliotecaRepository:** Maneja las operaciones relacionadas con la biblioteca personal de un usuario.

3.1.7. Utilidades

- **ImageLoader:** Utilidad para cargar imágenes desde URLs (portadas de libros).
- **FileUtils:** Utilidad para operaciones con archivos, especialmente para gestionar imágenes de perfil.
- **NotificationUtils:** Gestiona la creación y visualización de notificaciones locales.
- **DatabaseInitializer:** Responsable de precargar datos iniciales en la base de datos.

3.2. Base de datos

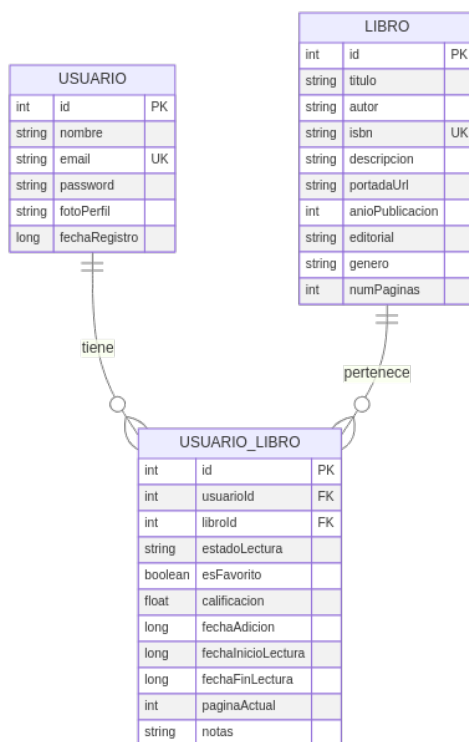


Figura 3.2: Diagrama de la base de datos de la aplicación LibreBook

Como se ilustra en la Figura 3.2, la aplicación utiliza Room como una capa de abstracción sobre SQLite para proporcionar un acceso más sencillo y seguro a los datos.

3.2.1. Estructura de la base de datos

La base de datos está definida por la clase **AppDatabase**, que extiende de `RoomDatabase`[7], y contiene las siguientes entidades:

- **Entidad Libro:** Almacena información completa sobre libros.
 - **Atributos principales:** id, titulo, autor, isbn, descripcion, portadaUrl, anioPublicacion, editorial, genero, numPaginas.
 - **Índices:** ISBN es un campo único indexado para búsquedas eficientes.
- **Entidad Usuario:** Almacena datos de los usuarios registrados.
 - **Atributos principales:** id, nombre, email, password, fotoPerfil, fechaRegistro.
 - **Índices:** email es un campo único indexado para prevenir duplicados y facilitar búsquedas.
- **Entidad UsuarioLibro:** Tabla de relación muchos a muchos que conecta usuarios con libros y almacena metadatos de la relación.
 - **Atributos principales:** id, usuarioid, libroid, estadoLectura, esFavorito, calificacion, fechaAdicion, fechaInicioLectura, fechaFinLectura, paginaActual, notas.
 - **Claves foráneas:** usuarioid referencia a Usuario.id, libroid referencia a Libro.id.
 - **Índices:** Índice compuesto sobre (usuarioid, libroid) para asegurar unicidad y optimizar consultas.

3.2.2. Operaciones principales

La base de datos soporta las siguientes operaciones a través de los DAOs:

- **Operaciones CRUD básicas:**

- Inserción, actualización y eliminación de libros, usuarios y relaciones.
- Consultas por identificador, ISBN, email, etc.
- Listados completos de entidades.

- **Consultas específicas:**

- Búsqueda de libros por título o autor (mediante LIKE).
- Filtrado por género o autor.
- Obtención de listas de géneros y autores únicos.
- Autenticación de usuarios (validación de email y contraseña).
- Búsqueda de usuarios por nombre o email.

- **Consultas relacionales:**

- Obtención de todos los libros de un usuario, opcionalmente filtrados por estado de lectura.
- Obtención de libros favoritos.
- Cálculo de estadísticas de lectura (conteo por estado, calificación promedio).

3.2.3. Inicialización y precarga

La clase **DatabaseInitializer** se encarga de:

- Crear la base de datos si no existe.
- Precargar 10 libros de Dostoievski para tener datos de ejemplo.
- Crear usuarios predeterminados (Administrador y Xabier).
- Establecer relaciones iniciales entre el usuario Xabier y algunos libros con diferentes estados de lectura, calificaciones y notas.

3.2.4. Transacciones y concurrencia

- Todas las operaciones de base de datos se realizan en hilos secundarios mediante `ExecutorService` para no bloquear la interfaz de usuario[2].
- Los callbacks permiten notificar a la UI cuando las operaciones se completan.
- Room maneja automáticamente las transacciones para asegurar la integridad de los datos.

4. Manual de usuario

4.1. Pantalla de Inicio

Al abrir la aplicación, se muestra la pantalla de inicio con libros recomendados y un mensaje de bienvenida. Desde aquí, puedes acceder a todas las funcionalidades principales de la aplicación.

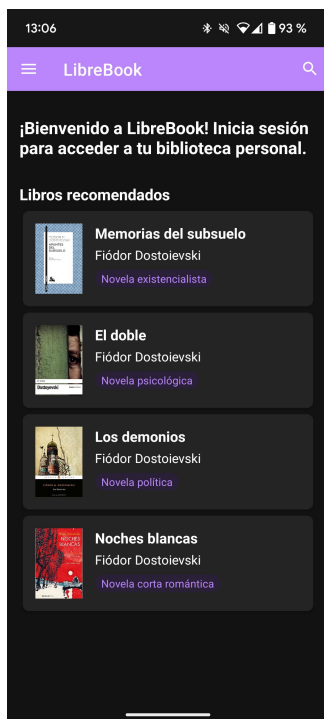


Figura 4.1: Pantalla de inicio con recomendaciones de libros

Como se observa en la Figura 4.1, la pantalla principal muestra:

- Un mensaje personalizado de bienvenida
- Una selección de libros recomendados
- Acceso al menú lateral mediante el icono superior izquierdo
- Acceso a la función de búsqueda mediante el icono de lupa

4.2. Menú de Navegación Lateral

El menú lateral permite acceder a todas las funciones de la aplicación. Las opciones disponibles varían según si has iniciado sesión o no.

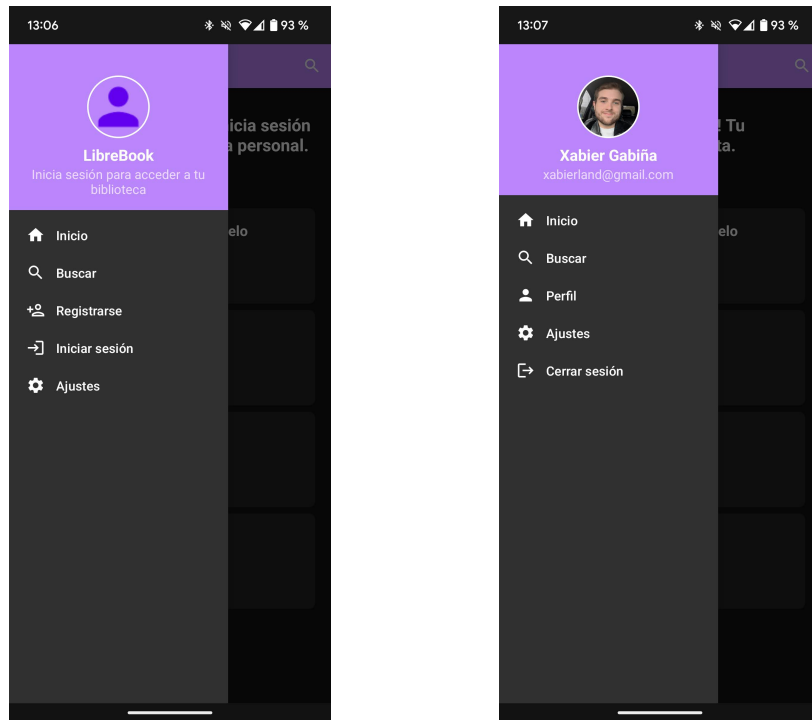


Figura 4.2: Menú para usuario anónimo (izquierda) y usuario autenticado (derecha)

Las opciones del menú incluyen:

- **Inicio:** Vuelve a la pantalla principal
- **Buscar:** Accede a la funcionalidad de búsqueda
- **Perfil:** Disponible solo para usuarios autenticados
- **Registrarse:** Para crear una nueva cuenta (solo visitantes)
- **Iniciar sesión:** Para acceder con cuenta existente (solo visitantes)
- **Ajustes:** Configuración de idioma y tema
- **Cerrar sesión:** Disponible solo para usuarios autenticados

4.3. Creación de Cuenta e Inicio de Sesión

Para acceder a todas las funcionalidades de la aplicación, debes crear una cuenta o iniciar sesión con credenciales existentes.

4.3.1. Registro de Nueva Cuenta

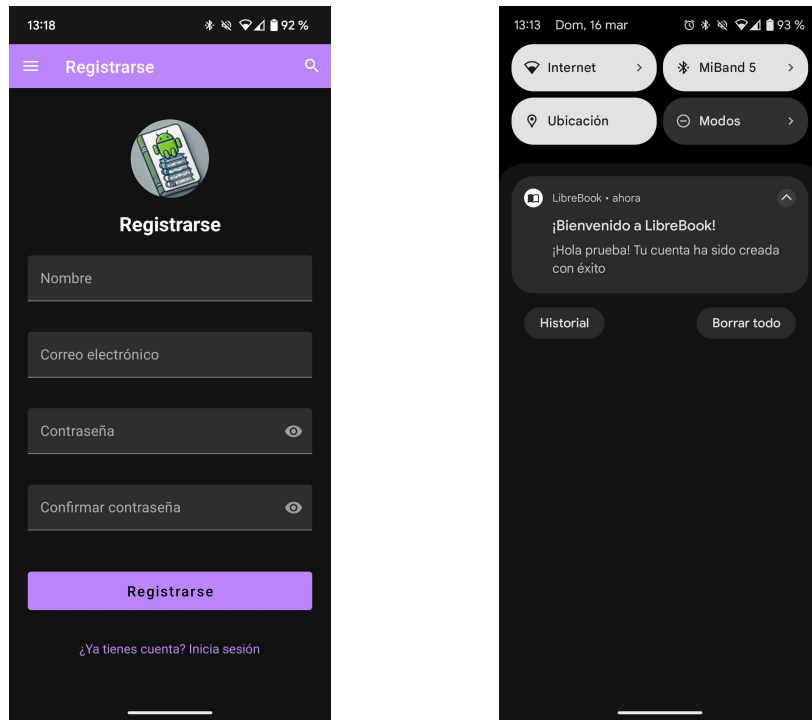


Figura 4.3: Pantalla de registro con formulario de datos (izquierda) y notificación de bienvenida (derecha)

Para registrarte:

1. Accede a la opción “Registrarse” desde el menú lateral
2. Completa todos los campos del formulario:
 - Nombre completo
 - Correo electrónico (debe ser único)
 - Contraseña (mínimo 6 caracteres)
 - Confirmación de contraseña
3. Presiona el botón “Registrarse”

Después de completar el registro exitosamente, recibirás una notificación de bienvenida:

4.3.2. Inicio de Sesión

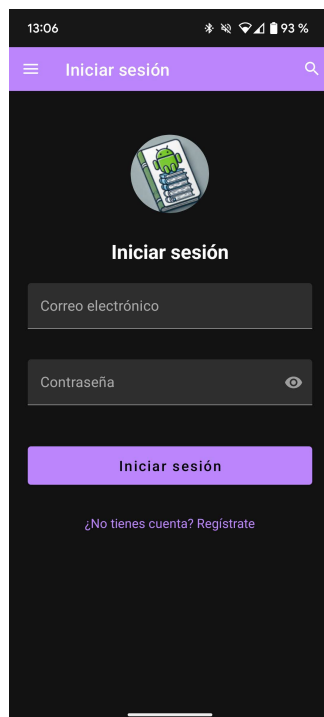


Figura 4.4: Pantalla de inicio de sesión

Para iniciar sesión:

1. Accede a la opción “Iniciar sesión” desde el menú lateral
2. Introduce tu correo electrónico registrado
3. Introduce tu contraseña
4. Presiona el botón “Iniciar sesión”

4.4. Perfil de Usuario y Biblioteca Personal

Una vez que has iniciado sesión, puedes acceder a tu perfil donde se muestran tus estadísticas de lectura y tus libros organizados en tres categorías.

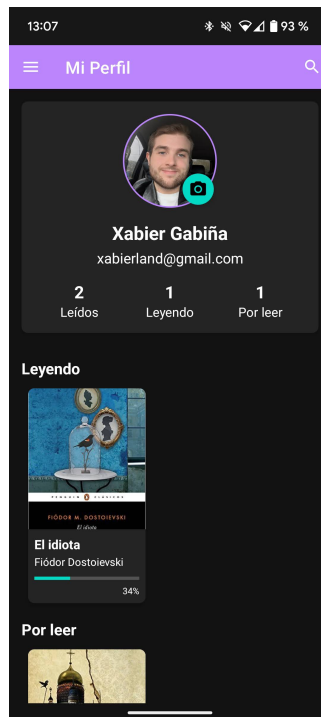


Figura 4.5: Pantalla de perfil de usuario con biblioteca organizada

En la pantalla de perfil encontrarás:

- **Información de usuario:** Foto de perfil, nombre y correo electrónico
- **Estadísticas de lectura:** Contadores de libros por estado
- **Sección “Leyendo”:** Libros que estás leyendo actualmente, con indicador de progreso
- **Sección “Por leer”:** Libros que tienes pendientes
- **Sección “Leídos”:** Libros que ya has terminado

4.4.1. Cambiar Foto de Perfil

Para cambiar tu foto de perfil:

1. Pulsa el botón de cámara junto a tu foto actual
2. Otorga permisos de acceso a la galería cuando se soliciten
3. Selecciona una imagen de tu dispositivo

4.5. Búsqueda de Libros y Usuarios

La función de búsqueda te permite encontrar libros por título o autor, así como buscar otros usuarios de la aplicación.

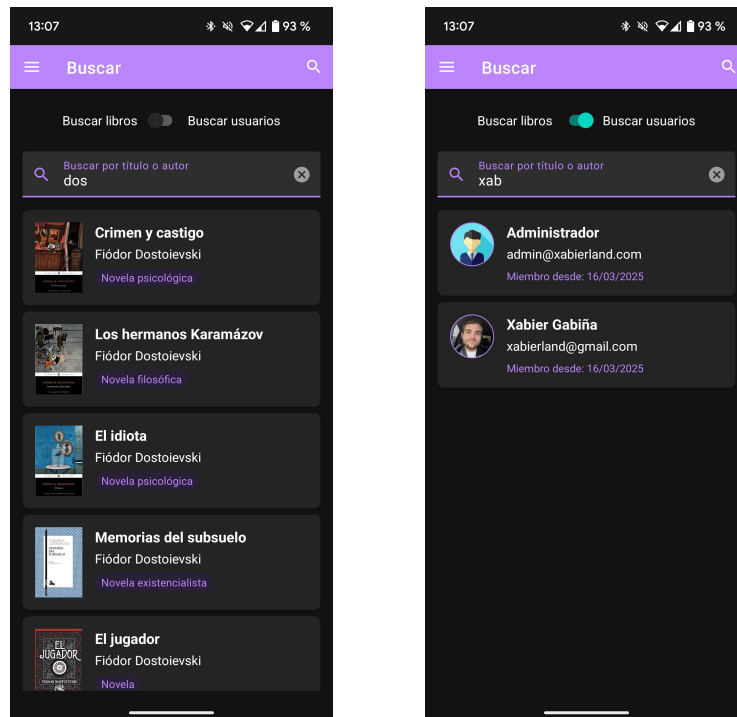


Figura 4.6: Búsqueda de libros (izquierda) y usuarios (derecha)

Para usar la búsqueda:

1. Accede a la función desde el icono de lupa o desde el menú lateral
2. Selecciona el tipo de búsqueda mediante el interruptor superior (libros o usuarios)
3. Escribe los términos de búsqueda en el campo de texto
4. Los resultados se mostrarán automáticamente a medida que escribes

4.6. Detalles de Libro y Gestión de Biblioteca

Al seleccionar un libro, puedes ver su información detallada y gestionarlo en tu biblioteca personal.

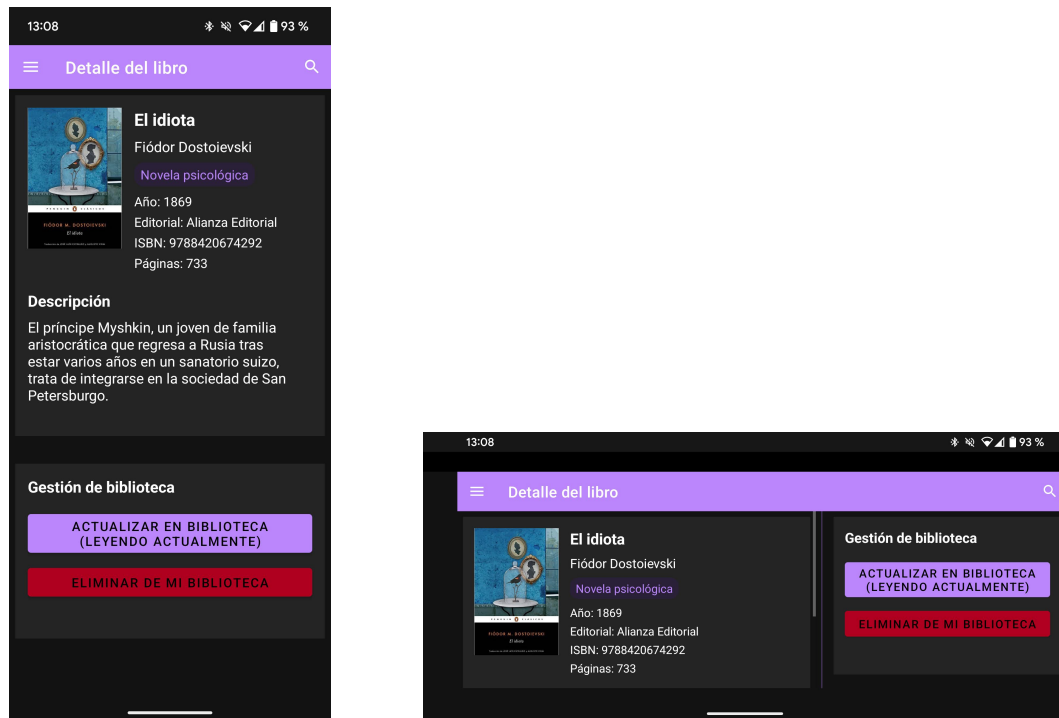


Figura 4.7: Pantalla de detalle de libro con información completa (izquierda) y vista horizontal (derecha)

La pantalla de detalles muestra:

- Portada del libro
- Título y autor
- Género literario
- Año de publicación y editorial
- Número de páginas e ISBN
- Descripción completa
- Opciones para gestionar el libro en tu biblioteca

Amba información se muestra en orientación vertical y horizontal, como se muestra en la Figura 4.7.

4.6.1. Añadir Libro a la Biblioteca

Para añadir un libro a tu biblioteca, presiona el botón “Añadir a mi biblioteca” y selecciona uno de los tres estados posibles:

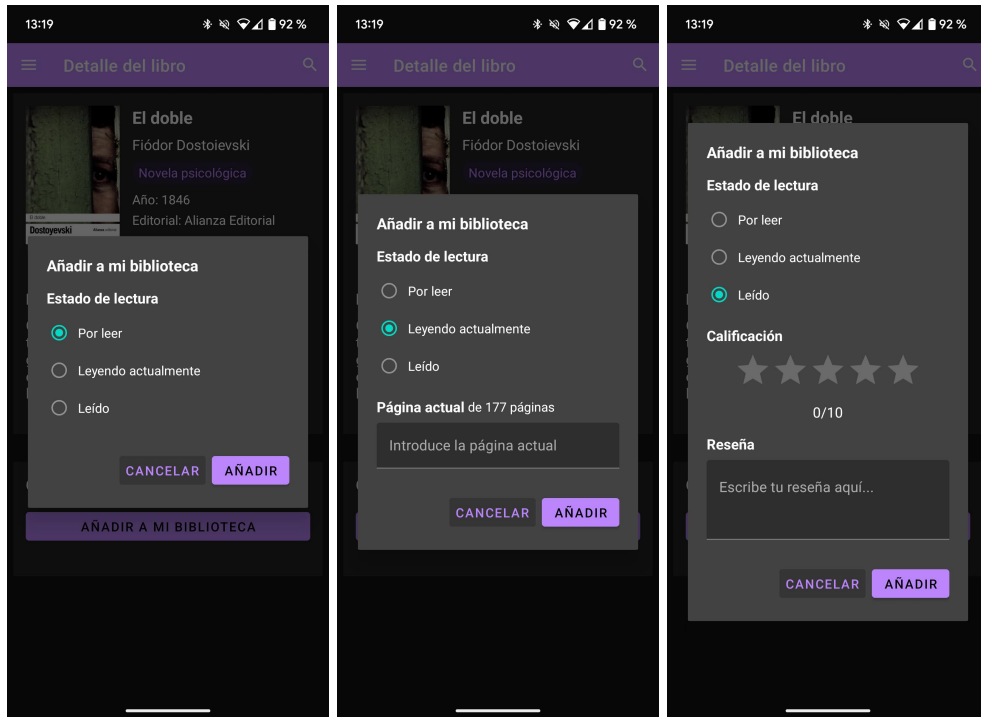


Figura 4.8: Diálogo para añadir libros

1. **Por leer:** Para libros que quieres leer en el futuro
2. **Leyendo actualmente:** Para libros que estás leyendo ahora
3. **Leído:** Para libros que ya has terminado

4.7. Ajustes de la Aplicación

La aplicación permite personalizar el idioma y el tema visual según tus preferencias.

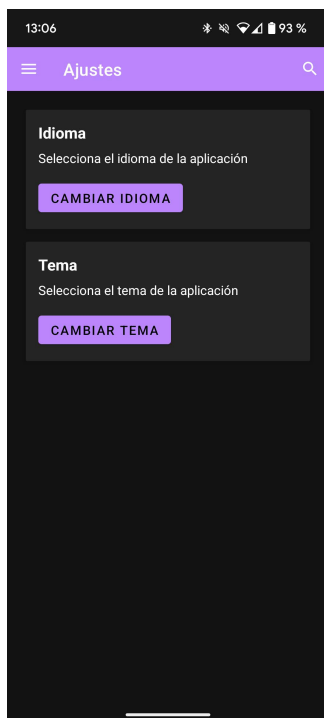


Figura 4.9: Pantalla de ajustes con opciones de personalización

4.7.1. Configuración de idioma

Para cambiar el idioma:

1. Accede a Ajustes desde el menú lateral
2. Pulsa en “Cambiar idioma”
3. Selecciona entre:
 - Predeterminado del sistema
 - English (Inglés)
 - Español
 - Euskera

4.7.2. Configuración de tema

Para cambiar el tema visual:

1. Accede a Ajustes desde el menú lateral
2. Pulsa en “Cambiar tema”
3. Selecciona entre:
 - Predeterminado del sistema
 - Tema claro
 - Tema oscuro

4.8. Cierre de Sesión

Para salir de tu cuenta:

1. Abre el menú lateral
2. Selecciona “Cerrar sesión”
3. Confirma la acción en el diálogo que aparece

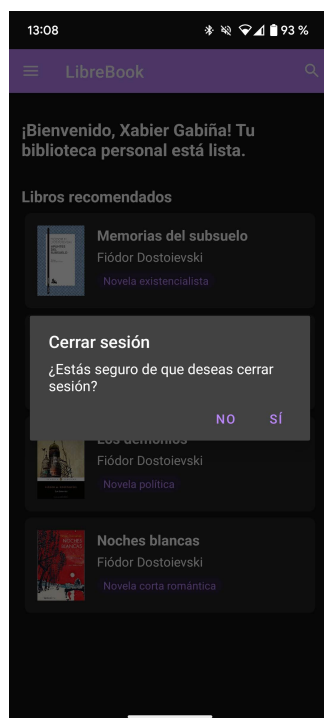


Figura 4.10: Diálogo de confirmación de cierre de sesión

5. Dificultades

Durante el desarrollo de LibreBook, me enfrenté a varios desafíos técnicos que requirieron soluciones específicas:

5.1. Implementación de Room y arquitectura de datos

Una de las primeras dificultades fue diseñar correctamente la estructura de la base de datos Room[7]. La relación entre usuarios y libros requería un esquema que permitiera:

- Mantener la integridad referencial entre tablas
- Almacenar metadatos específicos de la relación (estado de lectura, calificaciones, notas)
- Realizar consultas eficientes para obtener libros por estado o características

La solución implicó crear la entidad intermedia `UsuarioLibro` con claves foráneas y definir consultas SQL complejas, especialmente para la obtención de datos combinados (como en `LibroConEstado`). La anotación `@RewriteQueriesToDropUnusedColumns` fue crucial para manejar correctamente las proyecciones en las consultas JOIN.

5.2. Manejo de multi-idioma y temas

Implementar soporte para múltiples idiomas (inglés, español y euskera) presentó desafíos como[6]:

- Traducir correctamente todas las cadenas de texto manteniendo la coherencia
- Gestionar el cambio dinámico de idioma sin reiniciar la aplicación completa
- Manejar configuraciones regionales específicas y valores por defecto

De manera similar, la implementación de temas personalizados (claro, oscuro y modo sistema) requirió[9]:

- Definir atributos personalizados en `attrs.xml`
- Crear variantes de temas en `themes.xml` y `themes-night.xml`
- Implementar un mecanismo para aplicar los cambios de tema inmediatamente

La solución implicó utilizar `recreate()` para reiniciar las actividades, pero manteniendo el estado mediante `Intent extras`.

5.3. Concurrencia y operaciones asíncronas

Un desafío constante fue evitar bloquear la interfaz de usuario durante operaciones de base de datos. La solución adoptada fue[2]:

- Implementar un patrón de repositorio con `ExecutorService` para todas las operaciones de I/O
- Utilizar callbacks para notificar a la UI cuando las operaciones se completaban
- Actualizar la interfaz de usuario en el hilo principal mediante `runOnUiThread()`

Este enfoque mejoró significativamente la experiencia del usuario, pero añadió complejidad al código con múltiples niveles de callbacks anidados.

6. Conclusiones

Hacer esta aplicación me ha servido para aprender mucho sobre programación en Android. He podido juntar varias cosas que había aprendido por separado y ver cómo funcionan todas juntas en un proyecto real.

Ordenar bien el código desde el principio me ayudó bastante. Cuando tenía que arreglar algo o añadir una función nueva, no tenía que tocar todo el proyecto. Si empezara de nuevo, cambiaría algunas cosas, pero en general estoy contento con cómo quedó organizado.

La parte más difícil fue la base de datos. No había trabajado con Room antes y me costó entender cómo funcionaba. Al final, creo que conseguí hacerlo bien y me siento más cómodo con el manejo de bases de datos en Android.

Me hubiera gustado poder añadir funcionalidades para que los usuarios pudieran seguirse entre ellos y compartir sus opiniones sobre los libros. No obstante no he tenido tiempo suficiente para implementarlas.

En general, estoy satisfecho con el resultado y creo que he cumplido con los objetivos del proyecto. Espero que esta aplicación sea útil para los amantes de la lectura y que pueda seguir mejorándola en el futuro.

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] Futurice. *Android Best Practices*. 2022. URL: <https://github.com/futurice/android-best-practices>.
- [3] Android Developers. *Create and manage notifications*. 2023. URL: <https://developer.android.com/develop/ui/views/notifications>.
- [4] Android Developers. *Fragments*. 2023. URL: <https://developer.android.com/guide/fragments>.
- [5] Android Developers. *Guide to app architecture*. 2023. URL: <https://developer.android.com/topic/architecture>.
- [6] Android Developers. *Localize your app*. 2023. URL: <https://developer.android.com/guide/topics/resources/localization>.
- [7] Android Developers. *Room Persistence Library*. 2023. URL: <https://developer.android.com/training/data-storage/room>.
- [8] Google. *Material Design*. 2023. URL: <https://material.io/design>.
- [9] Material.io. *Android UI/UX Guidelines*. 2023. URL: <https://material.io/design/guidelines-overview>.