

Andorid1

1. Descripción General del Proyecto

Esta aplicación Android está diseñada para la gestión de un inventario de mangas, permitiendo realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre una base de datos local (Room) y sincronizar con un servidor JSON externo mediante Retrofit.

2. Arquitectura de la Aplicación

La aplicación sigue el patrón de arquitectura MVVM (Model-View-ViewModel) recomendado por Google para aplicaciones Android.

Capas de la Aplicación

Capa	Descripción
UI(View)	Pantallas composable con Jetpack Compose (PantallaInicio, PantallaInsertar, PantallaActualizar)
ViewModel	MangaViewModel: Gestiona el estado de la UI y la lógica de negocio.
Repository	RepositorioJSON y RepositorioBBDD: Abstracción de fuentes de datos
Data Source	ServicioApi(Retrofit) para JSON Server y InventarioDao(Room) para base de datos local.

3. Estructura del Proyecto

El proyecto está organizado en paquetes siguiendo las mejores prácticas de Android:

Paquete	Contenido
modelo/	Clases de datos: MangaJSON, MangaBBDD
conexion/	ServicioApi(Retrofit), InventarioBaseDatos(Room)
dao/	InventarioDao – Interfaces de acceso a datos Room
datos/	Repositorios, ContenedorApp, DrawerMenu
pantallas/	PantallaInicio, PantallaInsertar, PantallaActualizar
ui/	MangaApp, MangaViewModel, theme/

4. Modelo de Datos

MangaJSON (Modelo para API REST)

Data class utilizada para serialización/deserialización con Retrofit y Kotlinx

Serialization:

Campo	Tipo	Descripción
id	String	Identificador único del manga.
nombre	String	Título del manga.
autor	String	Autor o mangaka.
genero	List<String>	Lista de géneros (shonen, seinen, etc).
precio	Double	Precio unitario en euros.
stock	Int	Cantidad disponible en inventario.
agotado	Boolean	Indica si el producto está agotado.

5. Capa de Datos y Repositorios

ServicioApi (Retrofit)

Interface que define los endpoints de la API REST:

- **GET /mangas:** Obtener todos los mangas
- **POST /mangas:** Insertar nuevo manga
- **PUT /mangas/{id}:** Actualizar manga existente
- **DELETE /mangas/{id}:** Eliminar manga

RepositorioJSON

Implementa el patrón Repository para las operaciones con la API externa.

Proporciona una capa de abstracción entre el ViewModel y la fuente de datos remota.

RepositorioBBDD

Repository pattern para la base de datos local Room. Facilita las operaciones CRUD sobre la base de datos SQLite.

6. Interfaz de Usuario (Pantallas)

Pantalla Inicio

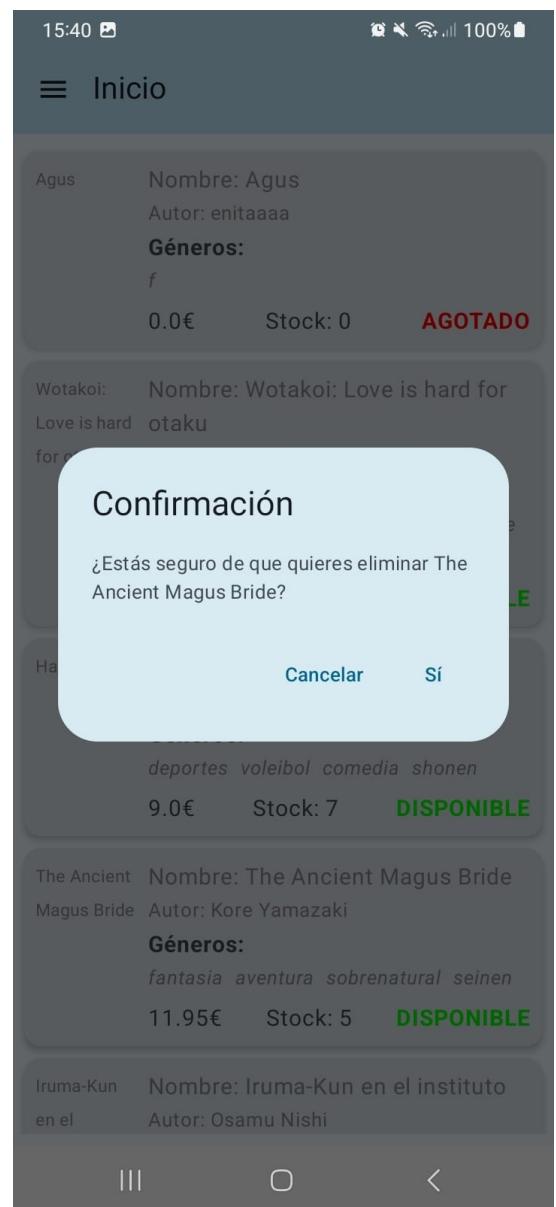
Funcionalidad:

- Muestra la lista completa de mangas en un LazyColumn
- Cada item es una Card con información del manga
- Click simple: navega a pantalla de actualización
- Long press: muestra diálogo de confirmación para eliminar
- Estados: Cargando, Error, Éxito con datos

Al inicializar la ejecución, aparece la pantalla de Inicio como predeterminada.



Longo Press: Si mantenemos pulsado uno de los elementos de la lista, aparecerá una ventana emergente, para aceptar o rechazar la eliminación de ese elemento.



En este caso aceptamos para comprobar que se puede eliminar:

15:47 100%

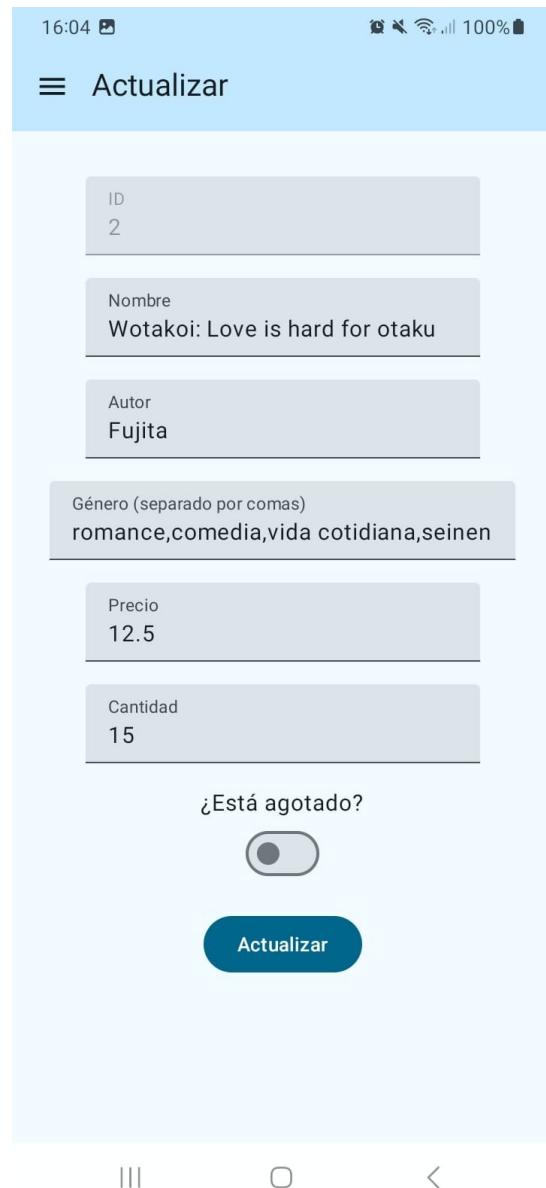
≡ Inicio

Agus	Nombre: Agus Autor: enitaaaa Géneros: f 0.0€ Stock: 0	AGOTADO
Wotakoi: Love is hard for otaku	Nombre: Wotakoi: Love is hard for otaku Autor: Fujita Géneros: romance comedia vida cotidiana seine n 12.5€ Stock: 15	DISPONIBLE
Haikyuu!!	Nombre: Haikyuu!! Autor: Haruichi Furudate Géneros: deportes voleibol comedia shonen 9.0€ Stock: 7	DISPONIBLE
Iruma-Kun en el instituto	Nombre: Iruma-Kun en el instituto Autor: Osamu Nishi Géneros: comedia fantasia escolar shonen 19.98€ Stock: 0	AGOTADO
One Piece	Nombre: One Piece Autor: Eiichiro Oda	

||| ○ <

Click simple:

Para poder actualizar hay que pulsar o hacer un click en uno de los elementos de la lista, este abrirá la ventana Actualizar, con el ID fijado, ya que no se puede modificar, todos los demás campos son modificables.

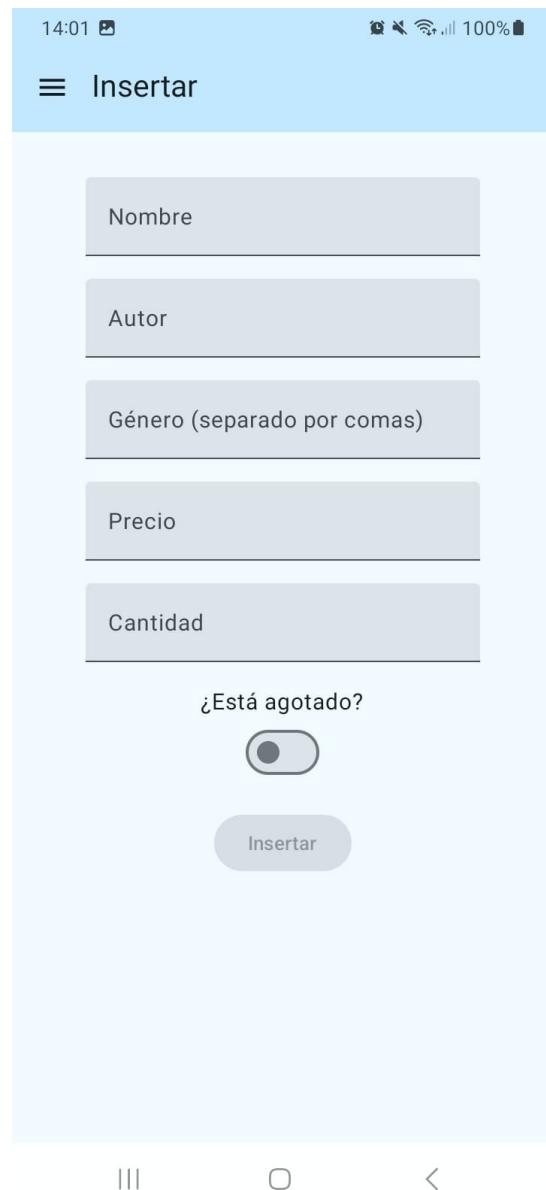


||| □ <

Pantalla Insertar

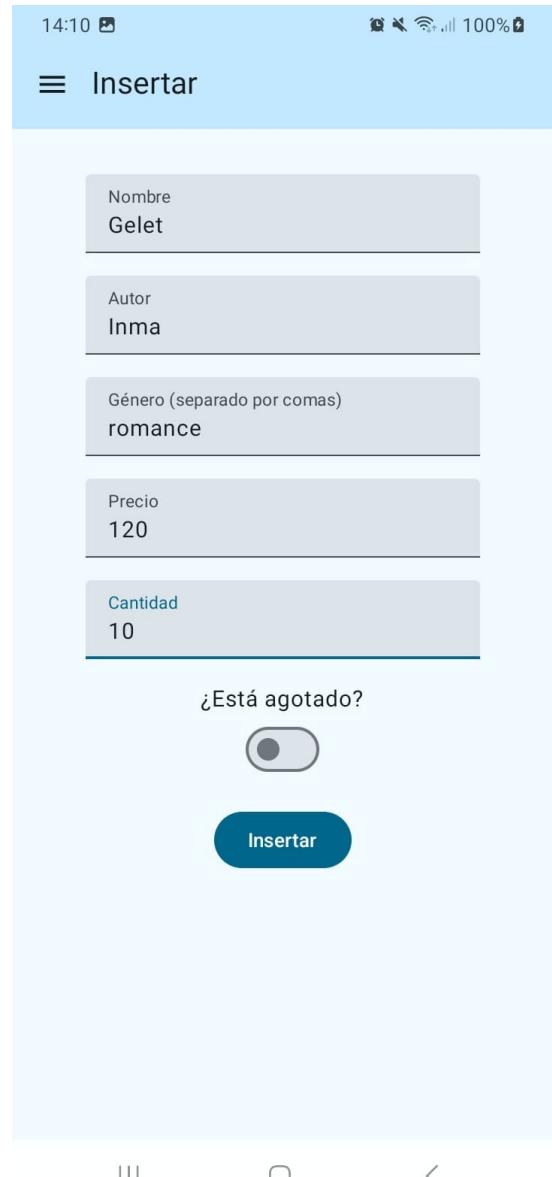
Funcionalidad:

- Formulario para crear un nuevo manga
- Campos: nombre, autor, géneros, precio, stock
- Validación de datos antes de enviar
- Cálculo automático del campo 'agotado' basado en stock



Esta hará una inserción en la BBDD.

Por ejemplo:



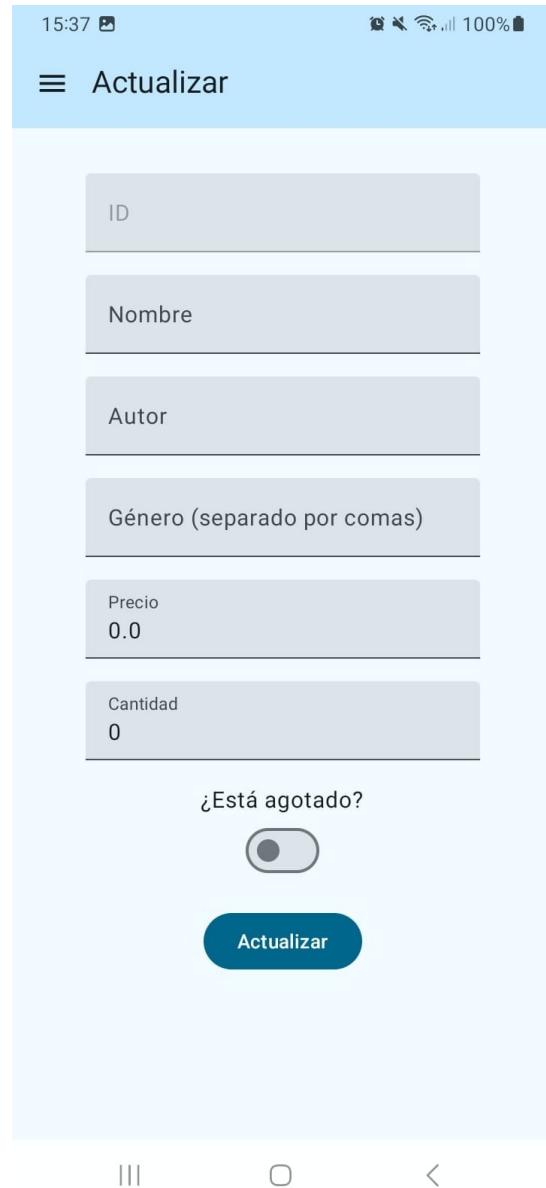
Insertamos esta información, y comprobamos en App Inspector si se ha insertado correctamente.

Mangas						
		Live updates ↗				
	id	nombre	autor	precio	cantidad	agotado
1	1	Gelet	Inma	120.0	10	0

PantallaActualizar

Funcionalidad:

- Formulario precargado con datos del manga seleccionado
- Permite modificar todos los campos excepto el ID
- Actualización en tiempo real en JSON Server



Otras pantallas:

-Menú desplegable:

En este menú, podemos acceder a tres ventanas, inicio, insertar y Actualizar



7. Funcionalidades CRUD

Operación	Método	Descripción
Create	insertarMangaBBDD()	Inserta un nuevo manga en la base de datos local
Read	obtenerMangasJSON()	Obtiene todos los mangas desde JSON Server
Update	actualizarMangaJSON()	Actualiza un manga existente en JSON Server
Delete	eliminarMangaJSON()	Elimina un manga por ID de JSON Server

8. Estructura JSON de Datos

A continuación, se presenta la estructura completa del archivo db.json que debe utilizarse con JSON Server:

```
{  
  "mangas": [  
    {  
      "id": "1",  
      "nombre": "Agus",  
      "autor": "enitaaaa",  
      "genero": ["f"],  
      "precio": 0,  
      "stock": 0,  
      "agotado": true  
    },  
    {  
      "id": "2",  
      "nombre": "Wotakoi: Love is hard for otaku",  
      "autor": "Fujita",  
      "genero": ["romance", "comedia", "vida cotidiana", "seinen"],  
      "precio": 12.5,  
      "stock": 15,  
      "agotado": false  
    },  
    {  
      "id": "3",  
      "nombre": "Haikyuu!!",  
      "autor": "Haruichi Furudate",  
      "genero": ["deportes", "voleibol", "comedia", "shonen"],  
      "precio": 9,  
      "stock": 7,  
    }  
  ]  
}
```

```
"agotado": false
},
{
"id": "4",
"nombre": "The Ancient Magus Bride",
"autor": "Kore Yamazaki",
"genero": ["fantasia", "aventura", "sobrenatural", "seinen"],
"precio": 11.95,
"stock": 5,
"agotado": false
},
{
"id": "5",
"nombre": "Iruma-Kun en el instituto",
"autor": "Osamu Nishi",
"genero": ["comedia", "fantasia", "escolar", "shonen"],
"precio": 19.98,
"stock": 0,
"agotado": true
},
{
"id": "6",
"nombre": "One Piece",
"autor": "Eiichiro Oda",
"genero": ["accion", "aventura", "fantasia", "shonen"],
"precio": 10,
"stock": 15,
"agotado": false
},
```

```

    "id": "acf2",
    "nombre": "Lycoreco",
    "autor": "a",
    "genero": ["a"],
    "precio": 2,
    "stock": 2,
    "agotado": false
}

]
}

```

9. Solución de Problemas

Problema	Solución
Error de conexión al servidor	Verificar que JSON Server esté ejecutándose en el puerto 3000. Comprobar que la URL sea http://10.0.2.2:3000/ para emulador Android.
Lista vacía en la app	Verificar que db.json contenga datos. Reiniciar JSON Server. Revisar logs del ViewModel para mensajes de error.
Error de serialización JSON	Asegurar que la estructura del JSON coincida exactamente con MangaJSON. Revisar que todos los campos requeridos estén presentes.
Problemas con Room Database	Limpiar caché de la app (Settings > Apps > Clear Data). Verificar que las migraciones de base de datos estén correctamente implementadas si se modificó el esquema.
Error de permisos de red	Verificar que AndroidManifest.xml incluya el permiso INTERNET. Comprobar que network_security_config.xml esté correctamente referenciado.