

# 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
<b>UI(View)</b>	Pantallas composables con Jetpack Compose (PantallaInicio, PantallaInsertar, PantallaActualizar)
<b>ViewModel</b>	MangaViewModel: Gestiona el estado de la UI y la lógica de negocio.
<b>Repository</b>	RepositorioJSON y RepositorioBBDD: Abstracción de fuentes de datos
<b>Data Source</b>	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
<b>modelo/</b>	Clases de datos: MangaJSON, MangaBBDD
<b>conexion/</b>	ServicioApi(Retrofit), InventarioBaseDatos(Room)
<b>dao/</b>	InventarioDao – Interfaces de acceso a datos Room
<b>datos/</b>	Repositorios, ContenedorApp, DrawerMenu
<b>pantallas/</b>	PantallaInicio, PantallaInsertar, PantallaActualizar
<b>ui/</b>	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
<b>id</b>	String	Identificador único del manga.
<b>nombre</b>	String	Título del manga.
<b>autor</b>	String	Autor o mangaka.
<b>genero</b>	List<String>	Lista de géneros (shonen, seinen, etc.
<b>precio</b>	Double	Precio unitario en euros.
<b>stock</b>	Int	Cantidad disponible en inventario.
<b>agotado</b>	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)

### PantallaInicio

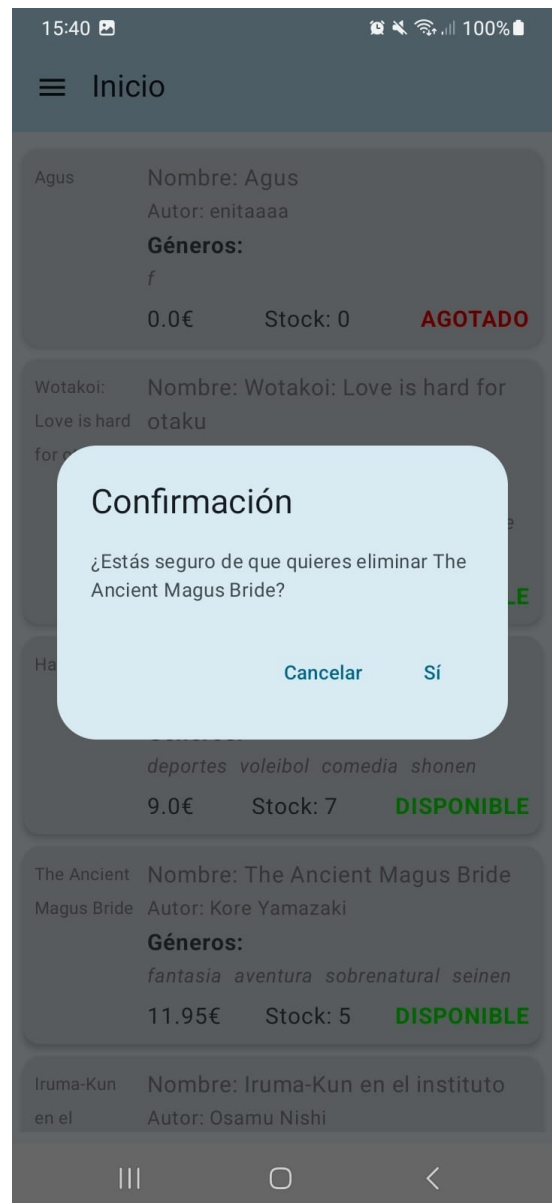
#### Funcionalidad:

- 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 <b>Géneros:</b> f	0.0€	Stock: 0 AGOTADO
Wotakoi: Love is hard for otaku	Nombre: Wotakoi: Love is hard for otaku Autor: Fujita <b>Géneros:</b> romance comedia vida cotidiana seinen	12.5€	Stock: 15 DISPONIBLE
Haikyuu!!	Nombre: Haikyuu!! Autor: Haruichi Furudate <b>Géneros:</b> deportes voleibol comedia shonen	9.0€	Stock: 7 DISPONIBLE
Iruma-Kun en el instituto	Nombre: Iruma-Kun en el instituto Autor: Osamu Nishi <b>Géneros:</b> 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.

16:04 100%

≡ Actualizar

ID  
2

Nombre  
Wotakoi: Love is hard for otaku

Autor  
Fujita

Género (separado por comas)  
romance,comedia,vida cotidiana,seinen

Precio  
12.5

Cantidad  
15

¿Está agotado?  
☐

Actualizar

## 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

14:01

Insertar

Nombre

Autor

Género (separado por comas)

Precio

Cantidad

¿Está agotado?

Insertar

Esta hará una inserción en la BBDD.

Por ejemplo:

14:10

Insertar

Nombre  
Gelet

Autor  
Inma

Género (separado por comas)  
romance

Precio  
120

Cantidad  
10

¿Está agotado?

Insertar

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

50



## Pantalla Actualizar

### Funcionalidad:

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

15:37 100%

≡ Actualizar

ID

Nombre

Autor

Género (separado por comas)

Precio  
0.0

Cantidad  
0

¿Está agotado?

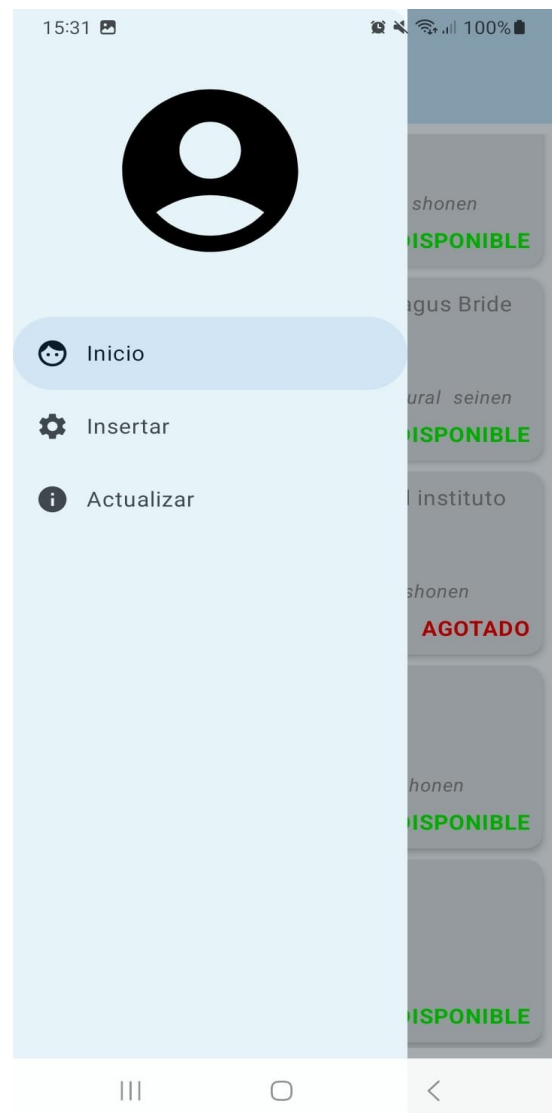
☐

Actualizar

### Otras pantallas:

-Menú desplegable:

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



## 7. Funcionalidades CRUD

Operación	Método	Descripción
<b>Create</b>	insertarMangaBBDD()	Inserta un nuevo manga en la base de datos local
<b>Read</b>	obtenerMangasJSON()	Obtiene todos los mangas desde JSON Server
<b>Update</b>	actualizarMangaJSON()	Actualiza un manga existente en JSON Server
<b>Delete</b>	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
<b>Error de conexión al servidor</b>	Verificar que JSON Server esté ejecutándose en el puerto 3000. Comprobar que la URL sea <code>http://10.0.2.2:3000/</code> para emulador Android.
<b>Lista vacía en la app</b>	Verificar que <code>db.json</code> contenga datos. Reiniciar JSON Server. Revisar logs del ViewModel para mensajes de error.
<b>Error de serialización JSON</b>	Asegurar que la estructura del JSON coincida exactamente con MangaJSON. Revisar que todos los campos requeridos estén presentes.
<b>Problemas con Room Database</b>	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.
<b>Error de permisos de red</b>	Verificar que AndroidManifest.xml incluya el permiso INTERNET. Comprobar que <code>network_security_config.xml</code> esté correctamente referenciado.