



## **Ingeniería económica.**

**Nombre:** Luis Ismael López Urbina

**Docente:** Luis Guido.

**Fecha:** 12 de Julio de 2024.

## Contenido

Introducción.....	3
Componentes del MVVM .....	4
Model.....	4
View o vista: .....	4
ViewModel: .....	4
Beneficios del MVVM .....	5
Desventajas del MVVM .....	5
Esquema General del MVVM .....	6

## Introducción

El patrón de arquitectura MVVM (Model-View-ViewModel) es una evolución del patrón MVC (Model-View-Controller) y MVP (Model-View-Presenter) de las aplicaciones web, el patrón está diseñado para facilitar la separación de la lógica de presentación y la lógica de negocio en aplicaciones de software, particularmente en el desarrollo de aplicaciones con interfaces de usuario ricas y complejas. MVVM es ampliamente utilizado en el desarrollo de aplicaciones para plataformas como Android, iOS, y frameworks webs como Angular, React y Vue.js.

En este informe abordaremos un poco acerca del patron MVVM y su utilización en el desarrollo de aplicaciones de Android.

## **Componentes del MVVM**

**Model:** Representa la capa de datos y la lógica de negocio de la aplicación. Incluye estructuras de datos, servicios, y operaciones que manejan los datos.

Gestionar la recuperación, almacenamiento y manipulación de datos son la responsabilidad de este componente, así como también el proveer datos y lógica de negocio a la capa de ViewModel.

Por ejemplo, en una aplicación de lista de tareas, el Model podría ser una clase Task y un repositorio TaskRepository que maneja las operaciones CRUD.

### **View o vista:**

Es la capa de presentación que interactúa directamente con el usuario. Representa la interfaz de usuario (UI) de la aplicación.

La función de esta capa a como se especifica en lo anterior es mostrar datos y capturar las interacciones del usuario a través de la interfaz, para lograr esto la Vista debe enlazarse a la ViewModel para reflejar cambios en los datos.

Por ejemplo en una aplicación de lista de tareas, la View podría ser una actividad o fragmento en Android que muestra una lista de tareas.

### **ViewModel:**

El ViewModel actúa como un intermediario entre el Model y la View. Contiene la lógica de presentación y gestiona el estado de la interfaz de usuario, su principal función es proveer datos a la View mediante el enlace de datos (data binding) así como el procesar las acciones del usuario y actualizar el Model.

## Beneficios del MVVM

El modelo de arquitectura MVVM al igual que otros similares ofrece una serie de beneficios a la hora de construir una aplicación, MVVM facilita la separación de la lógica de negocio (Model), la lógica de presentación (ViewModel), y la UI (View). Esto mejora la mantenibilidad y escalabilidad del código.

La lógica de presentación en la ViewModel puede ser probada de manera aislada sin necesidad de interactuar con la UI, lo que facilita la escritura de pruebas unitarias. También el ViewModel puede ser reutilizado entre diferentes Views, promoviendo la reutilización de código y reduciendo la duplicación.

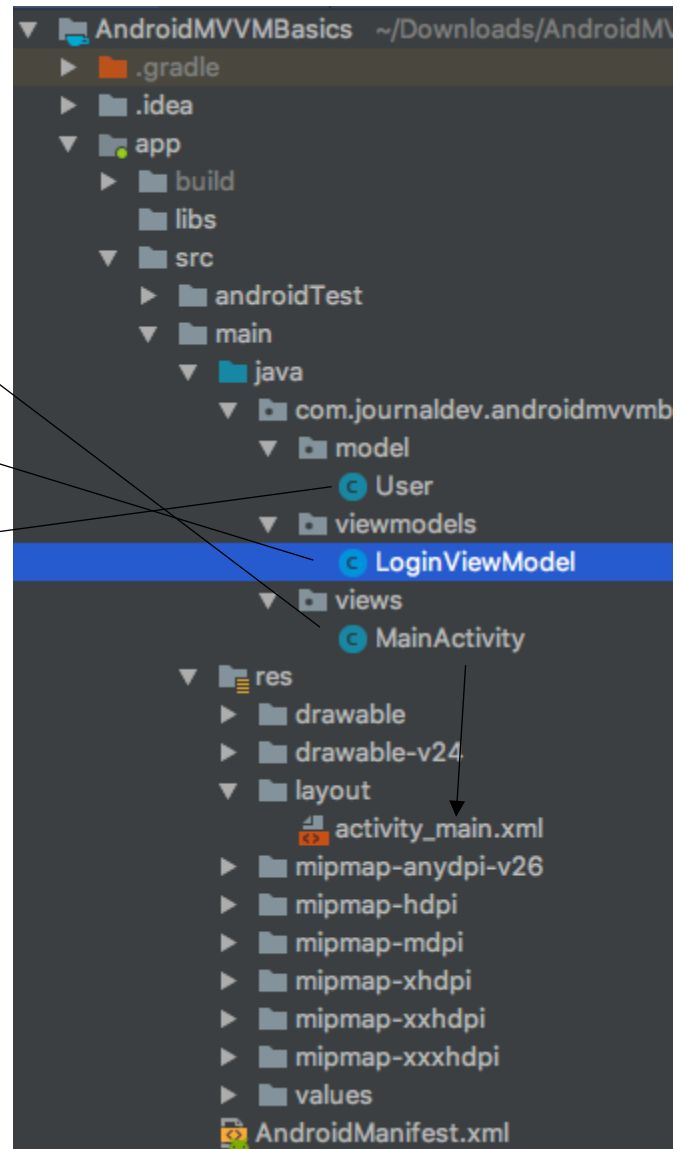
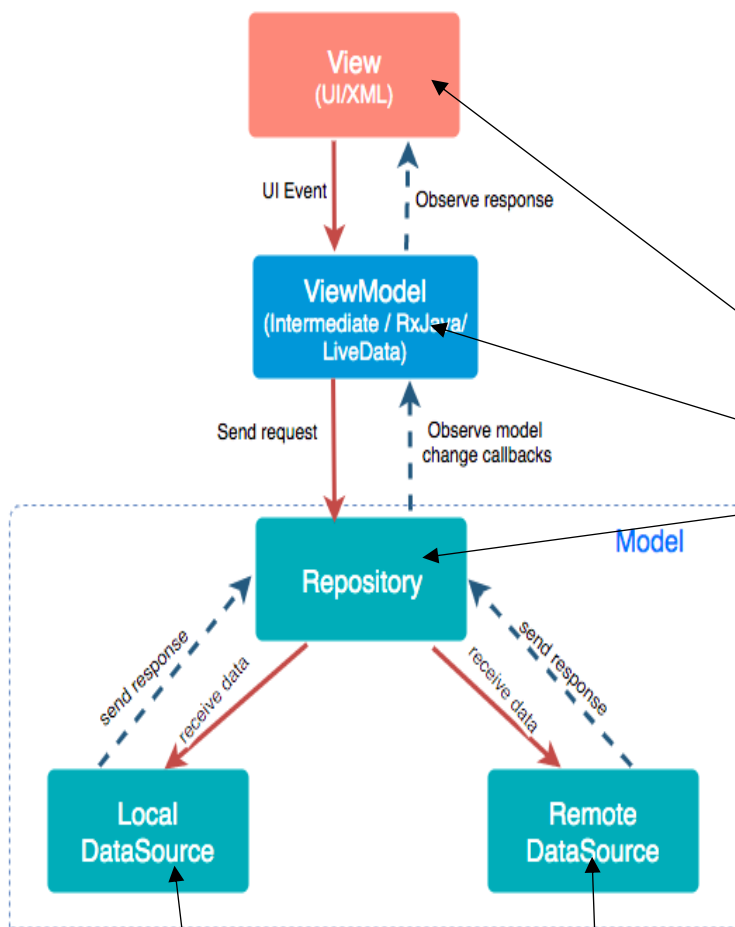
El uso del bindeo de datos o data binding permite que los cambios en los datos se reflejen automáticamente en la UI, reduciendo la necesidad de actualizar manualmente la interfaz de usuario.

## Desventajas del MVVM

La principal desventaja de MVVM es que puede ser más complejo de entender y dominar, especialmente para desarrolladores novatos.

El poco aprendizaje puede llegar a que la configuración inicial y la implementación de data binding y el mantenimiento a estas requiera de más tiempo y esfuerzo en comparación con patrones más simples.

## Esquema General del MVVM



Base de datos SQLite

Conexión a apis mediante  
Retrofit o Volley