



MVC (Modelo Vista Controlador)

Esta arquitectura de cómputo se usa comúnmente para implementar interfaces de usuario, lógica de control y manejo de datos. Funciona haciendo una separación de procesos los cuales se dividen en un modelo que maneja datos y lógica, una vista que hace la presentación y el diseño, así como un controlador que dirige diferentes comandos hacia los modelos o las vistas.

- Enfatiza una separación entre la lógica de negocios y su visualización.
- Es mejor la división de trabajo y mejora de mantenimiento.

1. Modelo: Maneja datos y lógica de negocios.
2. Vista: Se encarga del diseño y presentación.
3. Controlador: Enruta comandos a los modelos y vistas.

MVVM (Model View View Model)

Esta arquitectura de cómputo separa la lógica de presentación y trabajo de la interfaz de usuario, esto la hace más fácil de mantener, evolucionar y hacer pruebas al igual que permite mantener el proceso de diseño de la interfaz de usuario más dinámico y activo durante todo el proceso de desarrollo. Esta arquitectura se divide en tres componentes, la vista que se encarga de definir todo lo que el usuario ve en la pantalla, el view model que define comandos y propiedades que se ligan con diferentes datos de la vista y finalmente el modelo que maneja toda la lógica del programa.

- Con esta arquitectura es más fácil cambiar la vista del programa sin disrumpir la lógica.
- Se pueden llevar a cabo pruebas de unidades del modelo y del viewmodel sin tener que usar la vista.
- Los diseñadores y programadores pueden trabajar de manera paralela de manera independiente.

Comparación de Arquitectura

MVC	MVVM
Una de las arquitecturas de software más antiguas	Patrón de arquitectura para aplicaciones reconocido en la industria
La interfaz de usuario (vista) y el mecanismo de acceso a datos (modelo) están estrechamente relacionados.	Este patrón de arquitectura está más orientado a eventos, ya que utiliza el enlace de datos y, por lo tanto, facilita la separación de la lógica comercial principal de la vista.
El controlador y la vista existen con la relación de uno a muchos. Un controlador puede seleccionar una vista diferente según la operación requerida.	La vista múltiple se puede mapear con un solo modelo de vista y, por lo tanto, existe una relación de uno a muchos entre la vista y el modelo de vista.
Es difícil hacer cambios y modificar las funciones de la aplicación ya que las capas de código están estrechamente acopladas.	Fácil de hacer cambios en la aplicación. Sin embargo, si la lógica de vinculación de datos es demasiado compleja, será un poco más difícil depurar la aplicación.
Las entradas del usuario son manejadas por el controlador.	La vista toma la entrada del usuario y actúa como el punto de entrada de la aplicación.
Esta arquitectura tiene una alta dependencia de las API de Android.	Depende poco o nada de las API de Android.

Arquitectura utilizada en TC2005B

Debido al poco tiempo que se tuvo para desarrollar la aplicación web de TC2005B, se utilizó el método espagueti, ya que no utilizamos del todo los componentes de la arquitectura MVC o MVVM, esto porque no terminamos de separar o lograr que trabajaran de manera separada la vista del modelo que teníamos para obtener los datos y la lógica que se le darían a estos, al contrario, el código se podría decir que está revuelto como espagueti, como lo es la parte de la obtención y vista de los datos que el usuario podría llegar a solicitar.