

**Universidad Nacional Politécnica, UNP**

Programación Móvil

Investigación III: Patrón de arquitectura (Model View View-Model)

Informe de investigación

Programación Móvil

Fecha: 22 de junio de 2024

Profesor: Luis Guido Calderon

Ingeniería en computación V año

Joselin Raquel Morales Sotelo

**Patrón de arquitectura (Model View View-Model)**

**Informe de investigación**

**Programación Móvil**

El patrón de arquitectura Model-View-ViewModel (MVVM) es una metodología de diseño ampliamente adoptada en el desarrollo de aplicaciones móviles, especialmente en plataformas como Android. MVVM promueve una separación clara de responsabilidades entre los componentes de la aplicación, facilitando la mantenibilidad, la prueba y la escalabilidad del código. Este informe explorará los fundamentos, componentes clave, beneficios y aplicación práctica del patrón MVVM en el contexto de Android.

MVVM, por sus siglas en ingles Model View ViewModel, es un patrón de diseño que tiene como finalidad separar la parte de la interfaz del usuario, de la parte de la lógica del negocio, logrando asi que la parte visual sea totalmente independiente. El otro componente es el ViewModel que es la parte que va a interactuar como puente entre la vista y el modelo.

**Ventajas del Uso de MVVM**

El uso del patrón **MVVM** proporciona múltiples ventajas importantes en el desarrollo de aplicaciones. A continuación, se destacan algunas de las principales **ventajas**:

1. **Separación de Problemas:** Al dividir el código en Model, View y ViewModel, se consigue una clara separación de problemas, lo que facilita la mantenibilidad y escalabilidad de la aplicación.
2. **Reutilización de Código:** La separación de responsabilidades permite una mayor reutilización del código, ya que los Models y ViewModels pueden ser fácilmente probados y reutilizados en diferentes partes de la aplicación o en diferentes aplicaciones.
3. **Pruebas Unitarias:** La lógica de negocios y la lógica de UI están contenidas en el ViewModel, facilitando la realización de pruebas unitarias en el código. Esto aumenta la calidad y fiabilidad del software.
4. **Desarrollo Paralelo:** Los desarrolladores de software y los diseñadores de interfaz de usuario pueden trabajar de manera paralela en el Model y la View sin interferencias, gracias a la independencia de los componentes.
5. **Flexibilidad y Extensibilidad:** La arquitectura de MVVM se puede adaptar fácilmente a cambios en los requisitos del negocio o en la interfaz de usuario, proporcionando una mayor flexibilidad y extensibilidad.

**Componentes Principales del MVVM**

El patrón MVVM se compone de tres componentes esenciales: **Model**, **View** y **ViewModel**. A continuación, se describe la función de cada uno de estos componentes:

1. **Model:** El Model representa la lógica de negocio y los datos de la aplicación. Es responsable de la gestión de los datos, incluyendo la recuperación, persistencia y actualización de la información. El Model es independiente de la UI y se encarga de la lógica y las reglas de negocio.
2. **View:** La View se encarga de la presentación de los datos y de la interacción del usuario. Es una representación visual de la información contenida en el ViewModel. La View implementa los elementos de la interfaz de usuario y es responsable de capturar las acciones del usuario y transmitirlas al ViewModel.
3. **ViewModel:** El ViewModel actúa como un intermediario entre el Model y la View. Contiene la lógica de presentación que prepara los datos del Model para la View y maneja las interacciones del usuario. El ViewModel notifica a la View de cualquier cambio en los datos, asegurando una sincronización en tiempo real.