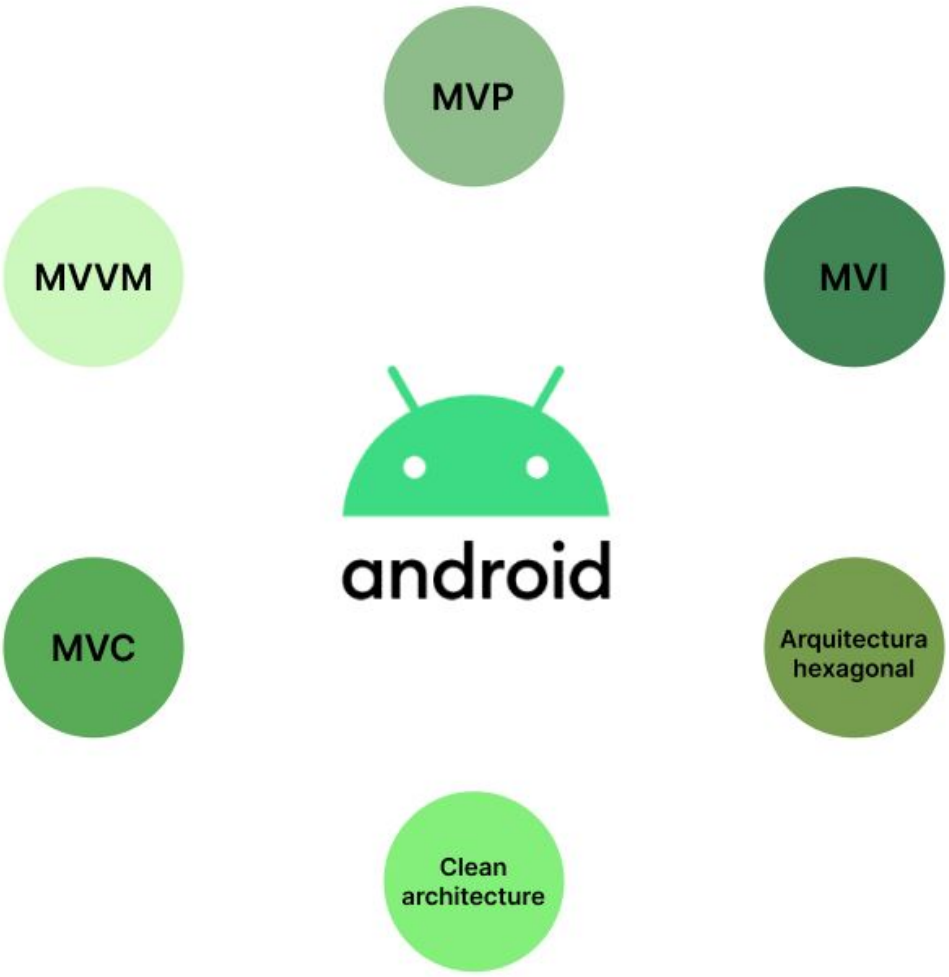
Arquitecturas del software



Ana María Jiménez Pérez

Índice

1. Introducción

En la actualidad, el desarrollo de software se lleva a cabo siguiendo un enfoque estructurado con el objetivo desarrollar un software que sea adaptable, escalable y con componentes reutilizables. Para ello se aplican arquitecturas de software que nos ayudan a diseñar una estructura coherente y eficiente para el software. Con esto se logra simplificar el proceso de desarrollo y tener como producto final sistemas mas robustos y fáciles de mantener a lo largo del tiempo.

Debido a la importancia del uso de las arquitecturas, en este documento se verán cinco casos prácticos donde se planteará una arquitectura para cada uno justificando dicha elección.

2. Arquitecturas a tener en cuenta para los supuestos prácticos

1. MVVM (Model View ViewModel): separa la lógica de presentación de la interfaz de usuario, utilizando un ViewModel para gestionar la lógica y proporcionar datos a la Vista.
2. MVP (Model View Presenter): separa la lógica de presentación de la Vista, utilizando un Presentador para actuar como intermediario entre la Vista y el Modelo.
3. MVI (Model View Intent): se centra en la gestión del estado de la aplicación mediante Intenciones que representan acciones del usuario y Modelos que reflejan el estado actual.
4. MVC (Model View Controler): divide la aplicación en tres componentes principales: el Modelo para la lógica de negocio, la Vista para la interfaz de usuario y el Controlador para gestionar las interacciones entre ellos.
5. Clean architecture: promueve la independencia de tecnología y la separación de capas para garantizar que la lógica de negocio sea independiente de la infraestructura y se pueda adaptar fácilmente.
6. Arquitectura hexagonal: se enfoca en la separación de capas y la exposición de puertos y adaptadores para interactuar con componentes externos, lo que la hace adecuada para aplicaciones con integraciones y límites claros.

3. Supuestos prácticos

3.1. Supuesto 1: Aplicación de E-commerce para una PYME

Para este primer caso práctico, lo más recomendable es usar la arquitectura MVVM.

Esta arquitectura facilitará la colaboración entre los encargados del desarrollo ya que separa muy bien la lógica de negocio, la lógica de presentación y la interfaz de usuario. Por lo tanto, teniendo en cuenta el tiempo y los recursos humanos limitados, puede agilizar el desarrollo de la aplicación ya que permitiría que el equipo trabaje en paralelo, donde el diseñador puede enfocarse en la vista sin depender del desarrollador. Esto ayudaría a que el diseño de la interfaz proporcione una buena experiencia de usuario.

Además, teniendo en cuenta que es una pequeña empresa y es posible que crezca en el futuro, esta arquitectura también facilitaría la escalabilidad en caso de ser necesario. También se agilizaría el desarrollo de pruebas unitarias.

3.2. Supuesto 2: Aplicación Social Interactiva para una Startup

Para este caso práctico sería conveniente usar la arquitectura MVI ya que se centra en la gestión del estado de la aplicación y se adapta bien a situaciones donde las interacciones en tiempo real son cruciales.

Este patrón nos ayudará a evitar los conflictos que se producen al mantener estados de un componente en diferentes niveles. También facilitaría encontrar los errores ya que se puede identificar en que estado de la aplicación aparecen.

Requiere de bastante código pero como se dispone de personal suficiente y de un tiempo de desarrollo algo elevado para la cantidad de recursos humanos, esto no debería suponer un problema.

Facilitaría la escalabilidad de la aplicación. Además también se separan las responsabilidades por lo que todo el equipo podría ir trabajando en paralelo.

3.3. Supuesto 3: Aplicación Financiera para una Gran Empresa

En este caso la clean architecture es la mejor opción. Permite separar la responsabilidad entre sus tres capas principales: capa de dominio, capa de aplicación y capa de presentación.

Es una arquitectura que si no se conoce tiene una curva de aprendizaje significativa, con cierta complejidad inicial y bastante código que desarrollar, sin embargo, como se dispone de un equipo de desarrolladores grande, es muy viable aplicarla.

Gracias al uso de esta arquitectura, la aplicación financiera será mas segura gracias a las distintas capas que presenta. También facilita el control de acceso y autenticación ya que se podría gestionar fácilmente en la capa de aplicación.

3.4. Supuesto 4: Plataforma de salud y bienestar para hospitales

En este cuarto supuesto práctico, la arquitectura hexagonal es la mejor opción. Facilita la integración de múltiples servicios y sistemas lo cual es ideal ya que hay que integrar la aplicación con los centros médicos para gestionar las citas y demás información de los pacientes.

Además se asegura la privacidad de los datos y su seguridad. Esto es gracias a las diferentes capas que posee que facilitan la implementación de las medidas de seguridad.

También es escalable y adaptable, lo cual es importante en el sistema de la salud donde las regulación sanitaria puede cambiar regularmente.

3.5. Supuesto 5: Aplicación Prototipo para un Hackathon

En este caso, en el que se debe desarrollar una aplicación en un tiempo muy breve y entre estudiantes los cuales aún no son expertos en el desarrollo de aplicaciones, la arquitectura MVP es la más adecuada.

No es muy compleja de entender ni requiere de gran cantidad de código. Permite la división de responsabilidades y esto ayudará al pequeño grupo de desarrolladores a trabajar de forma paralela para cumplir con el objetivo en el tiempo tan limitado. También pueden reutilizar componentes agilizando el desarrollo.

Además, como es una idea que se desarrolla en un corto periodo de tiempo, pueden surgir nuevos requisitos en el momento y con esta arquitectura la adaptabilidad y flexibilidad es muy factible.