

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

ELECCIÓN DE UNA ARQUITECTURA

*Selene González Curbelo*

Programación de Aplicaciones Móviles Nativas  
2023/2024

## **Resumen**

En el ámbito de la ingeniería de software, la elección de la arquitectura adecuada es crucial. Una arquitectura bien seleccionada no solo facilita el desarrollo y mantenimiento de la aplicación, sino que también impacta en su rendimiento, seguridad y experiencia del usuario (UX). En este ejercicio, abordaremos escenarios prácticos relacionados con el desarrollo de aplicaciones móviles, donde deberemos elegir la arquitectura más idónea según las circunstancias y restricciones presentadas.

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

Una pequeña empresa quiere lanzar su tienda online a través de una aplicación móvil nativa.

**Presupuesto:** Limitado.

**Tiempos de entrega:** 4 meses.

**Recursos humanos:** Un desarrollador principal y un diseñador.

**Rendimiento:** Se espera un tráfico moderado, pero es esencial que la aplicación sea rápida y eficiente.

En un proyecto de estas características, se puede elegir entre una arquitectura MVC o MVP. Ambas garantizan una clara separación de responsabilidades entre la lógica de negocio, la representación y el control de flujos, lo cual es fundamental para una tienda en línea. Para una PYME con recursos y tiempo limitados, la elección acertada sería MVC, ya que proporciona una estructura organizada sin añadir una complejidad innecesaria.

No es imprescindible considerar arquitecturas más sofisticadas como MVVM o Clean Architecture en este contexto, pues podrían agregar una carga adicional en un proyecto de alcance reducido. No hay justificación para recurrir a arquitecturas más complejas cuando MVC o MVP cumplen adecuadamente con los requisitos.

## 0.2. Supuesto 2: Aplicación Social Interactiva para una Startup

Una startup quiere crear una aplicación social con características interactivas, como chats en tiempo real y transmisiones en vivo.

**Presupuesto:** Moderado.

**Tiempos de entrega:** 6-8 meses.

**Recursos humanos:** Un equipo de tres desarrolladores, un diseñador y un programador backend.

**Rendimiento:** Se espera un alto tráfico y es crucial que la aplicación maneje interacciones en tiempo real.

Para una aplicación social interactiva, resultaría más apropiado optar por una arquitectura MVP o MVVM. Estas estructuras posibilitan una distinta demarcación entre la lógica de negocio y la presentación, siendo particularmente ventajosas para proyectos que implican interacciones en tiempo real. MVVM, gracias a su capacidad de enlace de datos, se revela especialmente eficaz para aplicaciones sociales interactivas que requieren actualizaciones dinámicas en la interfaz de usuario.

No se aconseja inclinarse hacia una arquitectura más sencilla como el MVC, ya que podría resultar insuficiente para gestionar la complejidad propia de una aplicación social. El MVVM ofrece una separación más nítida entre la lógica de negocio y la interfaz de usuario, lo cual resulta crucial en un contexto donde la interactividad ostenta un papel primordial.

En esta circunstancia, el MVI podría representar una excelente elección dada la naturaleza de interacción en tiempo real. Sin embargo, el MVP podría proporcionar una distinción de responsabilidades igualmente clara y efectiva.

Además, mencionar ejemplos como Instagram que utiliza MVP y WhatsApp que emplea una variante de MVP es una excelente manera de respaldar la elección de arquitecturas. Ambas aplicaciones tienen requisitos de interacción en tiempo real y necesitan una arquitectura que permita una gestión eficaz de la lógica de negocio y la interfaz de usuario.

## 0.3. Supuesto 3: Aplicación Financiera para una Gran Empresa

Una gran empresa financiera quiere desarrollar una aplicación para que sus clientes gestionen sus finanzas, con características como visualización de transacciones, transferencias y análisis financiero.

**Presupuesto:** Alto.

**Tiempos de entrega:** 10-12 meses.

**Recursos humanos:** Un equipo grande con múltiples desarrolladores, diseñadores, especialistas en seguridad y analistas.

**Rendimiento:** Se espera un tráfico muy alto y es esencial que la aplicación sea segura y eficiente.

Para una aplicación financiera de una gran empresa, se recomienda considerar la implementación de Clean Architecture o Arquitectura Hexagonal. Estas estructuras ofrecen una marcada separación de capas y posibilitan una gestión más eficiente de la complejidad inherente a una aplicación financiera. En este caso, la seguridad y la escalabilidad son aspectos críticos, y ambas arquitecturas proporcionan el nivel de flexibilidad y control necesario.

Ambas opciones arquitectónicas ofrecen un alto grado de escalabilidad, un elemento esencial para administrar grandes volúmenes de datos financieros y usuarios en una aplicación empresarial. Tanto Clean Architecture como Arquitectura Hexagonal permiten una implementación sólida de medidas de seguridad, un requisito vital en el contexto de una aplicación financiera.

Otras estructuras como MVC o MVVM pueden no ser lo suficientemente robustas en términos de seguridad y escalabilidad para una aplicación financiera a gran escala. Por lo tanto, es imperativo priorizar la seguridad y la escalabilidad en este entorno.

## 0.4. Supuesto 4: Plataforma de Salud y Bienestar para Hospitales

Un hospital de renombre desea desarrollar una aplicación móvil nativa que permita a los pacientes acceder a sus historiales médicos, programar citas, chatear con especialistas y recibir recomendaciones personalizadas basadas en su historial.

**Presupuesto:** muy alto.

**Tiempos de entrega:** 12-15 meses.

**Recursos humanos:** un equipo multidisciplinario compuesto por varios desarrolladores móviles, desarrolladores backend, especialistas en seguridad de la información, diseñadores UX/UI y analistas de sistemas.

**Rendimiento:** se espera un tráfico constante y alto debido a la gran cantidad de pacientes. La seguridad y privacidad de los datos es primordial.

Para una plataforma de salud y bienestar, las opciones más idóneas serían la Arquitectura Hexagonal o Clean Architecture. Dada la delicadeza de los datos médicos y la necesidad de cumplir con regulaciones estrictas, estas estructuras proporcionan un elevado nivel de seguridad y permiten una gestión eficaz de la complejidad inherente a una plataforma de esta naturaleza.

La seguridad es de primordial importancia en una aplicación de salud. Tanto Clean Architecture como Arquitectura Hexagonal permiten una implementación sólida de medidas de seguridad destinadas a resguardar la información médica.

Otras arquitecturas como MVC o MVVM pueden no ser lo suficientemente robustas en términos de seguridad y escalabilidad para una plataforma de salud que gestiona grandes volúmenes de datos médicos. La protección y privacidad de los datos son de máxima prioridad en este contexto.

## 0.5. Supuesto 5: Aplicación Prototipo para un Hackathon

Un grupo de estudiantes decide participar en un hackathon de 48 horas. Su objetivo es crear un prototipo funcional de una aplicación móvil que ayude a las personas a encontrar compañeros de viaje para compartir gastos en carreteras de peaje.

**Presupuesto:** Mínimo. Los estudiantes usarán herramientas y recursos gratuitos disponibles.

**Tiempos de entrega:** 48-72 horas.

**Recursos humanos:** Un equipo de tres estudiantes con habilidades mixtas: un desarrollador, un diseñador y alguien con habilidades de negocio.

**Rendimiento:** Como es un prototipo, no se espera un tráfico real. La aplicación debe ser lo suficientemente funcional para demostrar la idea.

Considerando el escaso plazo y el presupuesto ajustado, se aconseja optar por una arquitectura MVC o MVP para mantener la simplicidad y facilitar una implementación ágil. Otras estructuras como MVVM o Clean Architecture podrían resultar excesivas y complicadas para un proyecto que se desarrollará en un lapso de 48-72 horas. Por lo tanto, no se justifica el uso de arquitecturas más complejas cuando MVC cumple de manera adecuada con los requisitos de un prototipo.

Dado que se trata de un prototipo con un límite de tiempo para el desarrollo, la escalabilidad no es una prioridad inmediata. En este contexto, MVC proporciona una solución apta para un desarrollo veloz y eficiente.

## 0.6. Conclusiones

En resumen, al elegir la arquitectura para el desarrollo de aplicaciones móviles, es crucial realizar una evaluación cuidadosa, teniendo en cuenta los requisitos y recursos de cada proyecto. En los escenarios analizados, se consideraron situaciones que abarcan desde proyectos pequeños con presupuestos limitados hasta aplicaciones críticas de gran envergadura para empresas consolidadas.

Cada caso resaltó la arquitectura más adecuada, considerando factores como presupuesto, plazos de entrega, recursos humanos disponibles y requisitos de rendimiento y seguridad. Es evidente que no existe una solución universal, sino que cada proyecto requiere una evaluación individualizada para determinar la estructura arquitectónica más apropiada.

Finalmente, la elección de la arquitectura debe estar alineada con los objetivos y necesidades específicas del proyecto, asegurando un desarrollo eficaz, escalable y seguro. Esta consideración cuidadosa maximiza la probabilidad de éxito y garantiza la entrega de aplicaciones que cumplen con los estándares de calidad y funcionamiento esperados por los usuarios.

# Bibliografía

- [1] colaboradores de Wikipedia, “Modelo–vista–modelo de vista,” Wikipedia, La Enciclopedia Libre, Jul. 2021, [Online]. Available: [https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93modelo\\_de\\_vista](https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93modelo_de_vista)
- [2] colaboradores de Wikipedia, “Modelo–vista–presentador,” Wikipedia, La Enciclopedia Libre, Apr. 2022, [Online]. Available: <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93presentador>
- [3] S. Soni, “MVI(Model-View-Intent) pattern in Android - code Yoga - medium,” Medium, Jun. 05, 2018. [Online]. Available: <https://medium.com/code-yoga/mvi-model-view-intent-pattern-in-android-98c143d1ee7c>
- [4] “bdelespierre/laravel-clean-architecture-demo,” GitHub. <https://github.com/bdelespierre/laravel-clean-architecture-demo>
- [5] B. Delespierre, “Clean Architecture with Laravel,” DEV Community, Jul. 2023, [Online]. Available: <https://dev.to/bdelespierre/how-to-implement-clean-architecture-with-laravel-2f2i>
- [6] Iván Hernández, “Arquitectura y ciclo de vida de aplicaciones móviles nativas”, ULPGC, Oct. 2023, [Online]. Available: [https://aep24.ulpgc.es/pluginfile.php/350666/mod\\_page/content/14/arquitectura-2023-10-1-12-21-16.pdf](https://aep24.ulpgc.es/pluginfile.php/350666/mod_page/content/14/arquitectura-2023-10-1-12-21-16.pdf)