

Diseño arquitectónico de una Aplicación

● Sirius Protectora animal

Mariana Surós Álvarez

15/10/2023



ULPGC

Escuela de
Ingeniería Informática



Contents

Introducción	2
Supuesto 1: Aplicación de E-commerce para una PYME	2
Recomendación de arquitectura.....	2
Justificación	2
Supuesto 2: Aplicación Social Interactiva para una Startup	2
Recomendación de arquitectura.....	2
Justificación	2
Supuesto 3: Aplicación financiera para una gran empresa.....	3
Recomendación de arquitectura.....	3
Justificación	3
Supuesto 4: Plataforma de salud y bienestar para hospitales	3
Recomendación de arquitectura.....	3
Justificación	3
Supuesto 5: Aplicación prototipo para un hackathon	3
Recomendación de arquitectura.....	3
Justificación	4

Introducción

En este informe se ofrecen recomendaciones de arquitectura para diferentes escenarios de desarrollo de aplicaciones, teniendo en cuenta las necesidades y limitaciones de cada proyecto.

Supuesto 1: Aplicación de E-commerce para una PYME

Recomendación de arquitectura

Modelo-Vista-Controlador (MVC)

Justificación

Para el desarrollo de una aplicación de comercio electrónico destinada a una pequeña empresa con recursos y presupuesto limitados, se ha optado por MVC debido a su simplicidad y facilidad de implementación.

Esta arquitectura divide la aplicación en tres partes fundamentales: el Modelo, que se encarga de la lógica de negocio y los datos; la Vista, que se ocupa de la interfaz de usuario; y el Controlador, que coordina la comunicación entre el Modelo y la Vista. Esta división de responsabilidades permite que el equipo se enfoque en su área de experiencia, ya sea la lógica de negocio o el diseño de la interfaz de usuario.

El plazo de entrega de 4 meses es adecuado para esta arquitectura, ya que es conocida por ser relativamente sencilla y rápida de implementar.

Supuesto 2: Aplicación Social Interactiva para una Startup

Recomendación de arquitectura

Modelo-Vista-VistaModelo (MVVM)

Justificación

En el contexto de un “*startup*” con un presupuesto moderado y la necesidad de desarrollar una aplicación social altamente interactiva con un alto tráfico de usuarios, la elección de la arquitectura MVVM es perfecta.

MVVM se destaca por separar de manera efectiva la lógica de presentación de la vista, lo que resulta fundamental para gestionar de manera eficiente la interfaz de usuario y las actualizaciones en tiempo real. Una de sus ventajas principales es su capacidad para facilitar la comunicación bidireccional entre la Vista y el “*ViewModel*”, lo que garantiza una experiencia de usuario fluida.

La escalabilidad de MVVM es otra ventaja importante, ya que puede manejar el alto tráfico esperado en una aplicación social interactiva.

Dado que el equipo de desarrollo cuenta con tres desarrolladores, un diseñador y un programador “*backend*”, tiene el tamaño adecuado para abordar la complejidad de MVVM en un período propuesto.

Supuesto 3: Aplicación financiera para una gran empresa

Recomendación de arquitectura

Arquitectura Hexagonal (*Hexagonal Architecture*)

Justificación

Si consideramos una gran empresa financiera con un presupuesto alto y la necesidad de desarrollar una aplicación segura y eficiente, la elección de la Arquitectura Hexagonal es sólida.

La Arquitectura Hexagonal se destaca por su enfoque en la separación de capas en la aplicación, lo que permite una alta adaptabilidad en un entorno financiero en constante cambio, facilitando la evolución y la introducción de nuevas funcionalidades sin afectar negativamente a otras partes del sistema.

Además, esta arquitectura se integra fácilmente con medidas de seguridad sólidas, esenciales en el sector financiero. El equipo disponible es lo suficientemente grande y tiene la experiencia y la capacidad necesarias para implementar esta arquitectura en el período de tiempo propuesto.

Supuesto 4: Plataforma de salud y bienestar para hospitales

Recomendación de arquitectura

Arquitectura limpia (*Clean Architecture*)

Justificación

Para desarrollar una plataforma de salud y bienestar para hospitales con un presupuesto muy alto, la necesidad de un rendimiento constante y alto, así como la importancia crítica de la seguridad y la privacidad de los datos de los pacientes, la elección de la arquitectura “*Clean Architecture*” es la más acertada.

Esta arquitectura destaca por su enfoque en la separación de preocupaciones y la independencia de los detalles de implementación, lo cual es fundamental en una aplicación donde la integridad de los datos y la seguridad de la información son prioritarias. “*Clean Architecture*” permite implementar medidas de seguridad robustas y garantiza que la lógica de negocio se mantenga completamente independiente de las capas de detalles técnicos.

Además, el equipo multidisciplinario, compuesto por varios desarrolladores móviles, desarrolladores “*backend*”, especialistas en seguridad de la información, diseñadores UX/UI y analistas de sistemas, cuenta con la experiencia necesaria para implementar eficazmente esta arquitectura en el período propuesto.

Esta elección proporcionará una base sólida para el desarrollo de una plataforma de salud y bienestar confiable y segura para hospitales de renombre, cumpliendo con los estándares más exigentes de la industria.

Supuesto 5: Aplicación prototipo para un hackathon

Recomendación de arquitectura

Modelo-Vista-Presentador (MVP)

Justificación

Considerando el presupuesto mínimo y el tiempo de entrega extremadamente corto, Para desarrollar de un prototipo en un hackathon de 48-72 horas, el modelo MVP es la elección más adecuada y efectiva.

MVP ofrece una solución más simple en comparación con arquitecturas como MVVM y “*Clean Architecture*”, lo que resulta beneficioso para un desarrollo rápido y la iteración constante en un entorno de prototipado. En este escenario, la simplicidad es clave para cumplir con los plazos ajustados y los recursos limitados.

El equipo de tres estudiantes con habilidades mixtas puede trabajar de manera eficiente con MVP, ya que esta arquitectura proporciona una separación básica entre la lógica de presentación y la lógica de negocio. Esto permite una rápida implementación de las funcionalidades esenciales de la aplicación sin complicaciones innecesarias.