

OPEN MARKET



**Universidad
del Cauca**

Vigilada Mineducación

Ingeniería de Software II

Presentado por:

Brayan Majin Gaviria

Freider Escobar Cueltan

Jorge Andres Ayerbe Caicedo

Julian Ruano Majin

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Ingeniería de Sistemas

Popayán, 6 jul 2023

Índice

- 1. Arquitectura de la aplicación "Open Market">
 - 1.1. Desventajas de una arquitectura monolítica
 - 1.2. Ventajas de la arquitectura Hexagonal
- 2. Decisiones de diseño arquitectónico
 - 2.1. Patrones de diseño arquitectónico
 - 2.1.1. Arquitectura basada en microservicios (Microservices Architecture)
- 3. Diagramas de la arquitectura hexagonal

ARQUITECTURA DE LA APLICACIÓN "OPEN MARKET"

La primera decisión es la de utilizar una arquitectura hexagonal en lugar de una arquitectura monolítica.

Desventajas de una arquitectura monolítica

La arquitectura monolítica tiene una serie de desventajas notables en el contexto de las plataformas de ventas de productos, que afectan su funcionamiento y mantenimiento. Algunos de los problemas más comunes con la arquitectura monolítica son:

Fuerte acoplamiento: Todos los componentes del sistema están estrechamente conectados en una arquitectura monolítica. Esto significa que cualquier cambio en una parte del sistema puede tener un impacto imprevisto en otras partes, lo que dificulta el desarrollo y el mantenimiento del sistema en su conjunto.

Riesgo de fallo del sistema: Debido al acoplamiento, una falla puede afectar a todo el sistema. El reinicio completo del sistema puede ser necesario debido a una sola falla, lo que prolonga el tiempo de recuperación y aumenta la probabilidad de interrupciones prolongadas..

Escalabilidad limitada: La arquitectura monolítica no puede escalar horizontalmente porque se construye en una sola vez. El sistema puede enfrentar limitaciones físicas y económicas en el servidor a medida que aumenta la carga de trabajo, lo que dificulta la escalabilidad eficiente.

Dificultad en el desarrollo colaborativo: La arquitectura monolítica puede dificultar el desarrollo colaborativo cuando varios equipos deben trabajar en diferentes partes del código al mismo tiempo. La probabilidad de conflictos en el código aumenta y la integración continua se dificulta al compilar y desplegar todo el código en un solo repositorio.

Ventajas de la arquitectura Hexagonal

La arquitectura hexagonal ofrece ventajas como la separación de responsabilidades, flexibilidad, modularidad y facilidad de prueba y adaptación. Estas características contribuyen a un desarrollo más eficiente, mantenible y escalable de las aplicaciones.

- Permite centrar el desarrollo en la lógica del dominio y es una buena combinación con un enfoque de diseño dirigido por el dominio
- Permite pruebas automatizadas más rápidas y enfocadas para la lógica del dominio, así como la integración con bases de datos y otros servicios externos

- Permite la evolución independiente e incremental de preocupaciones: podemos dejar que nuestras API evolucionen a su propio ritmo, el mapeo del adaptador facilita el desacoplamiento.
- Permite la evolución del modelo de dominio para adaptarse mejor a la lógica empresarial, sin tener que romper las API o tener que migrar una base de datos en cada pequeña refactorización.

Separación de responsabilidades: La arquitectura hexagonal permite una clara separación de responsabilidades entre las capas de la aplicación. La lógica de negocio se encuentra en el núcleo (dominio) de la arquitectura, mientras que los adaptadores se encargan de la comunicación con componentes externos. Esto facilita la comprensión, el mantenimiento y la evolución del sistema.

Flexibilidad y adaptabilidad: la arquitectura hexagonal permite cambios en la infraestructura técnica sin afectar la lógica comercial. Esto facilita la adopción de nuevos marcos o tecnologías y la integración con sistemas externos. Además, permite realizar pruebas unitarias de la lógica comercial sin depender de la infraestructura.

Modularidad y reutilización: La arquitectura hexagonal fomenta la creación de componentes modulares y reutilizables. Los adaptadores se pueden intercambiar fácilmente para adaptarse a una variedad de contextos o requisitos técnicos, lo que facilita la escalabilidad y la evolución del sistema.

Pruebas unitarias: la realización de pruebas unitarias y de integración se facilita al separar la lógica de negocio de los detalles de la infraestructura técnica. Se puede realizar una prueba de capa de dominio de forma aislada, lo que mejora la cobertura de pruebas y ayuda a garantizar la calidad del software.

Conexión con sistemas externos: la arquitectura hexagonal permite la integración de adaptadores con sistemas externos. Estos adaptadores traducen los mensajes entre el sistema y los componentes externos, lo que hace que sea más fácil interoperar y comunicarse con servicios web, bases de datos u otros sistemas.

DECISIONES DE DISEÑO ARQUITECTÓNICO

Patrones de diseño arquitectónico

- La arquitectura hexagonal permite que cada microservicio implemente una parte específica del dominio de la aplicación. Esto significa que cada microservicio tendrá una capa de dominio separada que establecerá las reglas e interfaces comerciales.

- Los microservicios pueden comunicarse a través de API con protocolos como HTTP/REST o mensajería asíncrona. Cada microservicio tiene la capacidad de exponer sus propias APIs, lo que permite a otros microservicios utilizar sus servicios.
- La lógica necesaria para interactuar con bases de datos, sistemas de archivos y servicios de terceros se puede implementar en la capa de adaptadores de cada microservicio. Estos adaptadores serán responsables de traducir los datos y operaciones del dominio interno a los formatos y protocolos apropiados para la comunicación externa.
- Para evitar acoplamientos innecesarios y garantizar una interoperabilidad efectiva, es fundamental establecer una comunicación clara y bien definida entre los microservicios. Para crear y registrar las interfaces de microservicios, puede usar los contratos de API, las especificaciones de Swagger u otros mecanismos.
- Al combinar la arquitectura hexagonal con la arquitectura basada en microservicios, se puede lograr un diseño flexible, escalable y desacoplado que permite la evolución independiente de diversas partes del sistema. Sin embargo, es importante recordar que esta combinación presenta desafíos adicionales, como administrar la complejidad de la comunicación entre servicios.

Diagramas de la arquitectura hexagonal

