

Santiago Álvarez Muñoz

1. Decisiones de Diseño

Para implementar el backend de esta prueba técnica, elegí NestJS como framework principal debido a su escalabilidad, modularidad y su soporte nativo para patrones arquitectónicos como CQRS y DDD. Además, NestJS permite trabajar con monorepositorios, lo cual fue una decisión clave para mantener una organización clara del proyecto y facilitar su escalabilidad.

Dentro del monorepositorio, creé:

- Una librería compartida: shared
- Cinco aplicaciones independientes:
 - gateway
 - payments
 - products
 - reviews
 - sellers

Cada aplicación sigue el patrón de Separación de Comandos y Consultas (CQRS) y fue diseñada bajo principios de Diseño Orientado al Dominio (DDD). Esto permite mantener una separación clara de responsabilidades y facilita el mantenimiento y crecimiento futuro del sistema.

La única aplicación expuesta por HTTP es gateway, que actúa como orquestador del resto de las aplicaciones, las cuales se comunican a través de gRPC. Aunque el proyecto es sencillo en su estado actual, esta arquitectura permite una segmentación clara de microservicios y contextos, lo que facilita su escalado a futuro.

Otras decisiones importantes de diseño fueron:

- Se utilizó almacenamiento en memoria para simplificar el desarrollo, evitando la configuración de una base de datos.
- Las variables de entorno están incluidas dentro del proyecto, ya que no contienen información sensible.
- Las imágenes se sirven desde el backend como archivos estáticos, lo que simplifica la gestión de activos.

El frontend fue desarrollado en React con Tailwind CSS, utilizando una estructura simple basada en hooks personalizados para realizar las llamadas a la API y construir la interfaz requerida.

Se incluyeron archivos de configuración Docker para facilitar el despliegue y levantamiento del entorno completo de forma cómoda.

2. Retos en el Desarrollo

- **Restricción de Tiempo:** El tiempo fue uno de los principales retos durante el desarrollo. Para enfrentarlo, organicé mi agenda personal y profesional para dedicar espacios productivos que me permitieran avanzar de manera eficiente.
- **Definición de Contextos:** Una parte importante fue definir correctamente los contextos y módulos del sistema. Para esto, realicé una investigación basada en el sitio web de MercadoLibre, con el fin de comprender qué datos y procesos eran relevantes en cada dominio (productos, vendedores, pagos, reseñas, etc.).
- **Durante el desarrollo:** Identifiqué que existen soluciones mucho más robustas para abordar este tipo problema. Sin embargo, una parte esencial del trabajo fue analizar cuidadosamente qué elementos eran realmente necesarios y cuáles podían ser descartados sin afectar la calidad ni la escalabilidad del sistema.

Este proceso me permitió priorizar lo esencial y mantener el enfoque en los componentes clave del dominio, sin incorporar complejidades innecesarias. Aun así, cada decisión fue tomada buscando un balance adecuado entre simplicidad y robustez, garantizando que la arquitectura propuesta pudiera servir como una base sólida para futuras extensiones o migraciones hacia una infraestructura más avanzada.