

1. Primeros pasos

- Era la primera vez que hacía un desafío de este tipo, así que arranqué con lo que ya conocía: **NestJS**.
- Integré **Gemini**, que me resultó sencillo y práctico.
- Logré que interpretara intenciones y trajera productos/detalles desde un Excel.

2. Elección de base de datos

- Decidí sumar **Supabase** porque:
 - Usa **Postgres** (fácil de integrar).
 - Tiene **tier gratuito y interfaz simple**.
 - Para un modelo de datos sencillo no necesitaba un ORM.

3. Pruebas y limpieza

- Al inicio probé varias integraciones (ej. **OpenAI**, servicio de Excel).
- Una vez migrados los datos a Supabase, las eliminé para dejar un proyecto más claro.

4. Modularización y documentación

- Pasé de tener la lógica en un solo controlador a organizar módulos:
 - **Productos**
 - **Cart** (endpoints mínimos del challenge)
 - **Chat** (para recibir y derivar mensajes)
- Consulté con **Claude**, que me sugirió usar **Swagger** para documentar parámetros.
- Mi método: primero un **mínimo funcional**, luego agrego validaciones, documentación y robustez.
- Para la documentación aproveché **ChatGPT**, que me ayudó a generar un **README personalizado**.

- Además, diagramé el proyecto con **Mermaid**, siendo la primera vez que la usaba.

5. Refinamiento con IA

- Reinterpreté el challenge con ayuda de documentación y consultas a Claude.
- Ajusté cómo responde **Gemini** a las consultas.
- Decidí **posponer los tests en NestJS** porque me podían atrasar más de lo que aportaban en esta etapa.

6. Integraciones finales

- Configuré **Meta** y logré enviar mensajes (lo suficiente para la entrega).
- Estoy trabajando en mejorar la integración con **WhatsApp**.
- Aunque la integración está hecha, estoy teniendo problemas con el **vencimiento de los access tokens**.
- También estoy buscando una forma adecuada de hacer el **deploy de la API**.