Cambio de Arquitectura y Pruebas de Microservicios en EcoMarket SPA

Ricardo Gaete, Damián Vergara, Felipe Villalón

Instituto Profesional Duoc UC – Sede Maipú

# Índice

1. Diagrama de arquitectura de microservicios

2. Plan de pruebas

2.1 Pruebas Unitarias

2.2 Pruebas de integración

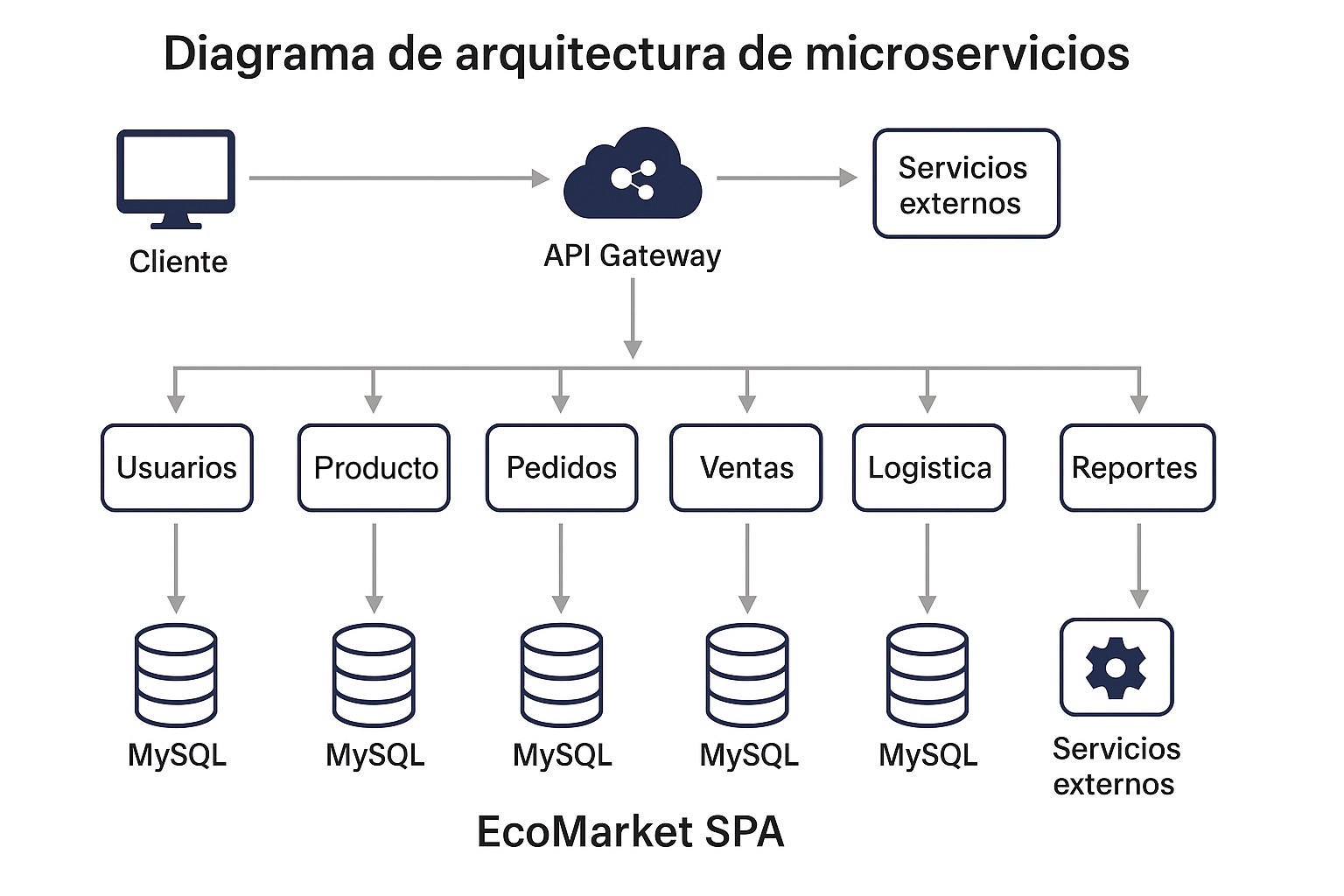
3. Ejecución de pruebas

4. Git – GitHub

5. Documentacion

6. Conclusión

# 1. Diagrama de arquitectura de microservicios



Este diagrama ilustra cómo se dividen los microservicios de EcoMarket SPA, con un API Gateway central y servicios independientes como usuarios, pedidos, inventario, ventas, logística y reportes. Cada servicio tiene su propia base de datos, siguiendo el principio de independencia.

# 2. Plan de pruebas

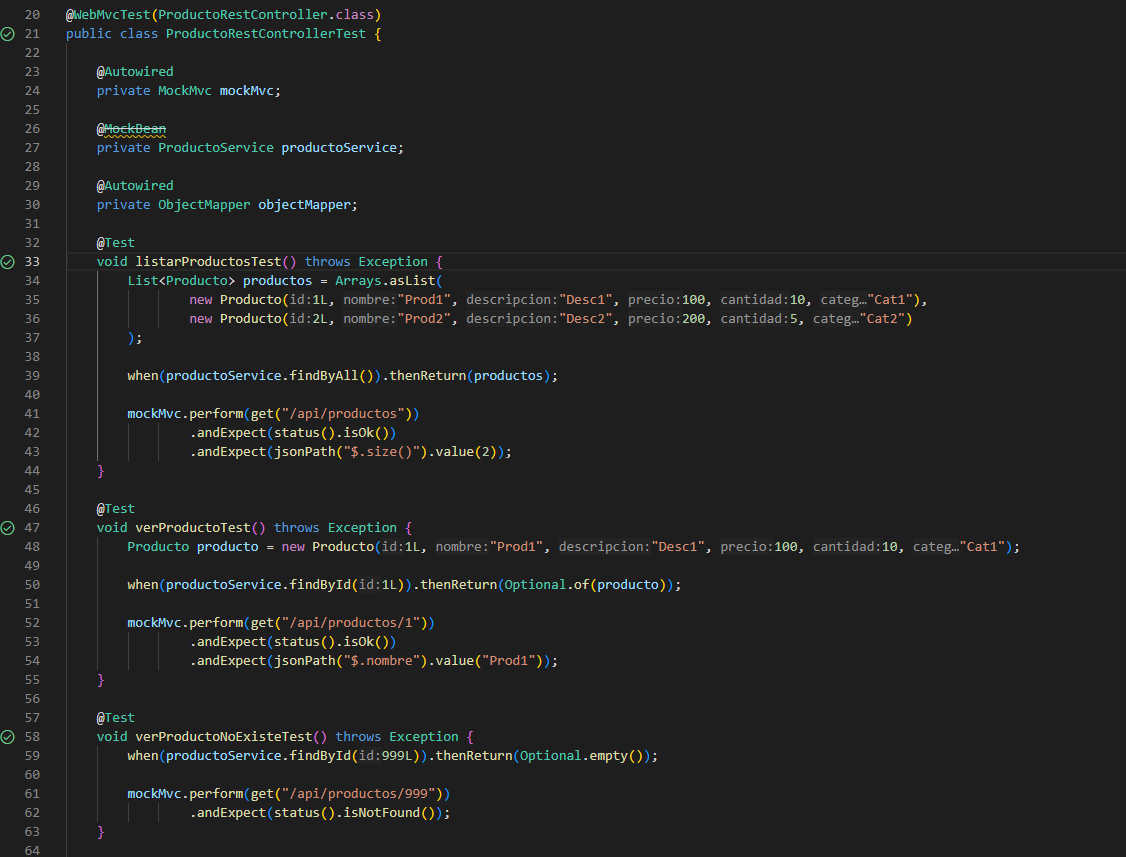
Herramientas utilizadas:

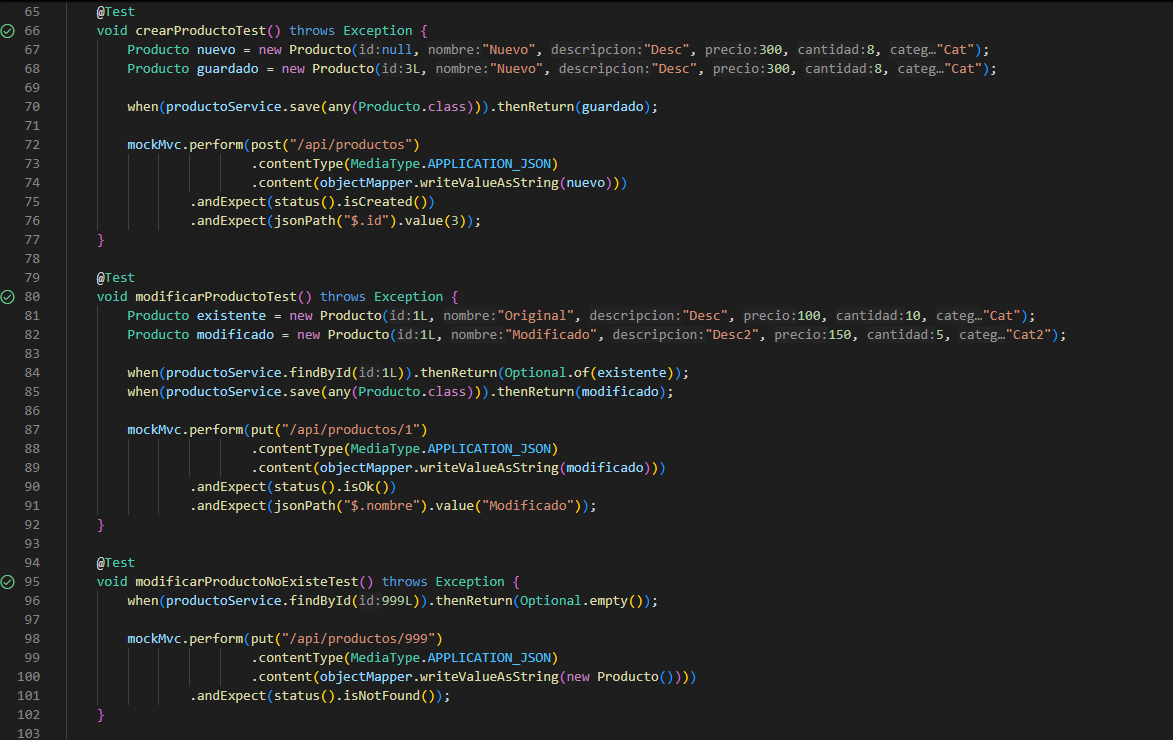
- JUnit 5: framework para pruebas unitarias.  
- Mockito: permite simular comportamientos.  
- Spring Boot Test: pruebas de integración.

## 2.1 Pruebas Unitarias

Se testearon los métodos de los servicios usando Mockito para simular la base de datos, evitando así dependencia del entorno externo.

-PRODUCTO

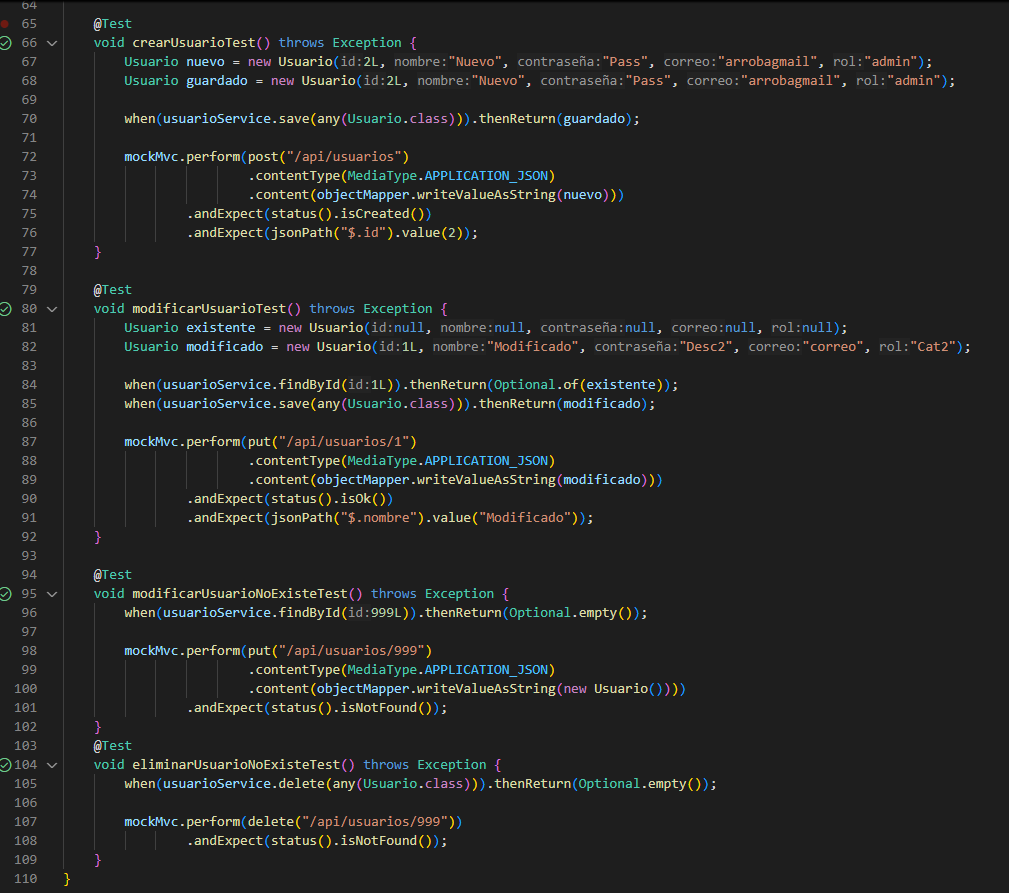






-USUARIO





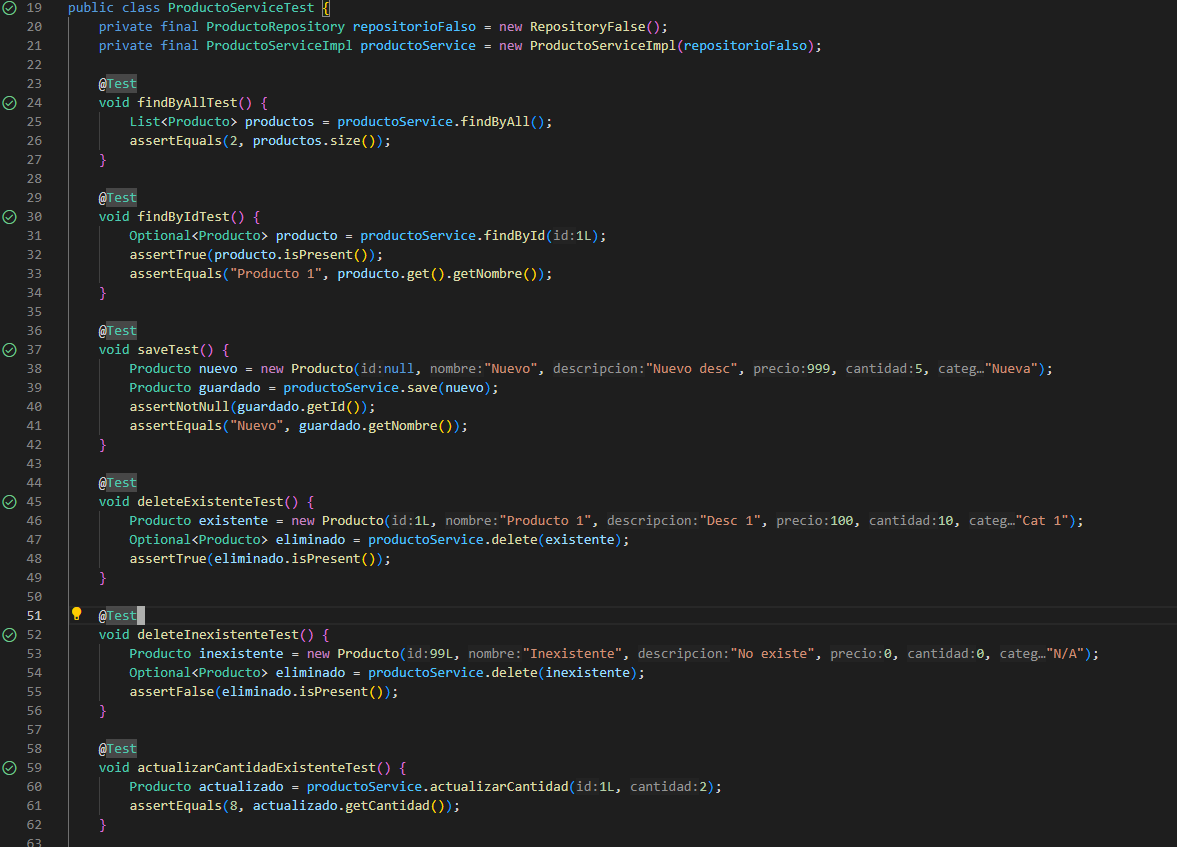
## 2.2 Pruebas de integración

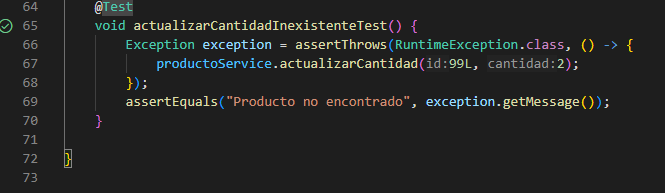
Se implementaron pruebas de integración en controladores utilizando Spring Boot Test. Se validaron los endpoints mediante llamadas simuladas y se verificó el comportamiento esperado.

-USUARIO

# 

-PRODUCTO

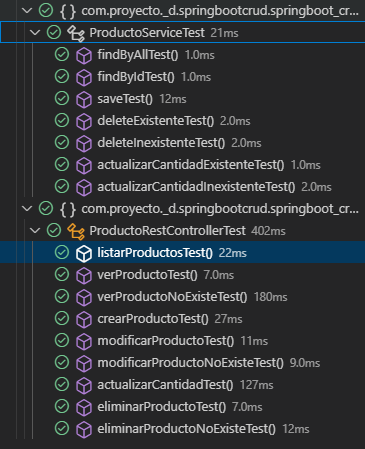




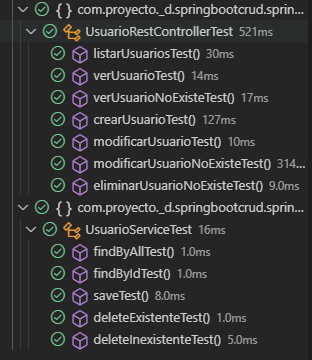
# 3. Ejecución de pruebas

Aqui se puede ver como las pruebas de los microservicios fueron ejecutadas y funcionaron correctamente.

-PRODUCTO

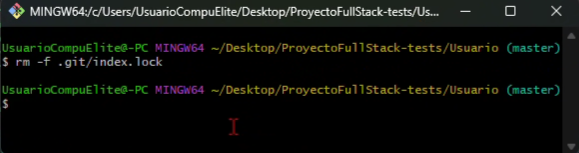


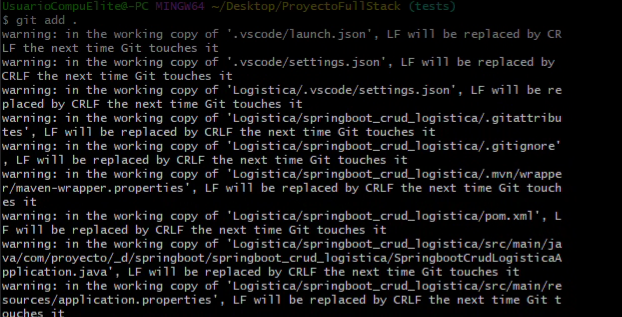
-USUARIO



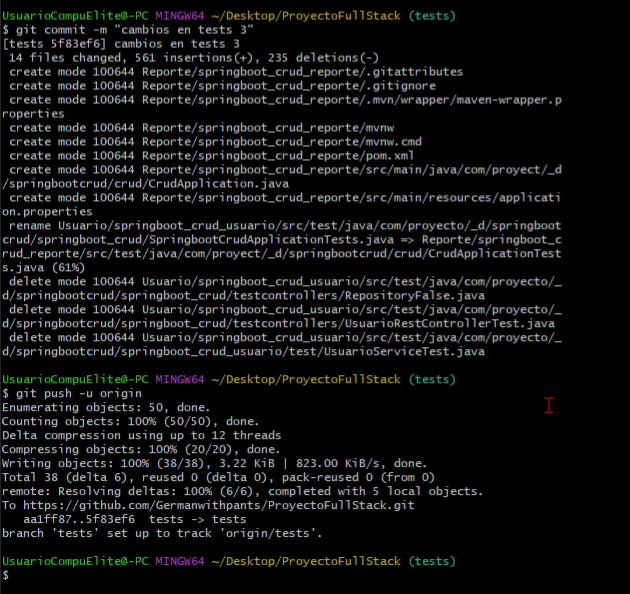
# 4. Git – GitHub

Comandos utilizados:



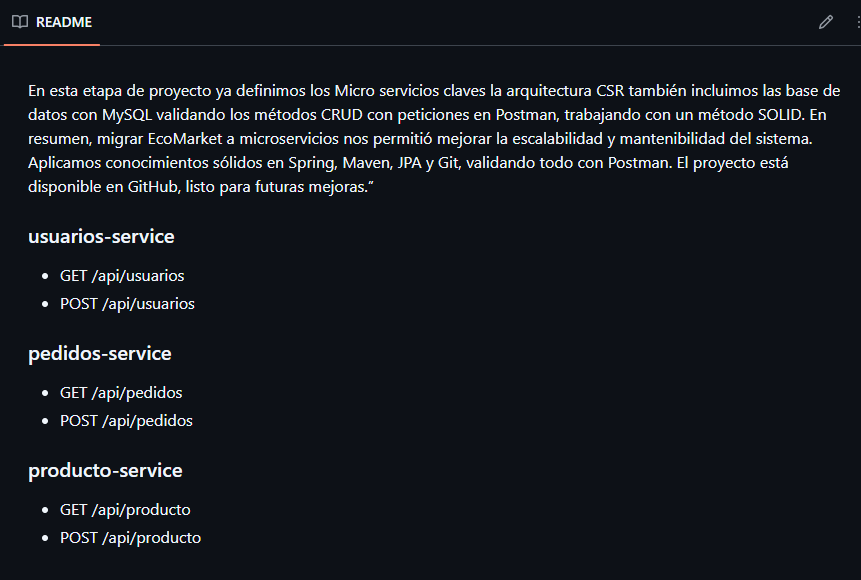
se utiliza para **eliminar el archivo de bloqueo** que Git crea automáticamente.

El comando git add le dice a Git que **prepares archivos para ser incluidos en el próximo commit**



-Git commit –m ”Cambios en tests 3”: este comando hace que todo lo seleccionado en el git add y lo guarda en el repositorio con el mensaje de “Cambios en tests 3”.

-Git push –u origin: Sube el commit realizado anteriormente al repositorio remoto

Extracto del archivo README.md:**

# 5.Documentacion

-PRODUCTOS:

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Forma, Rectángulo

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

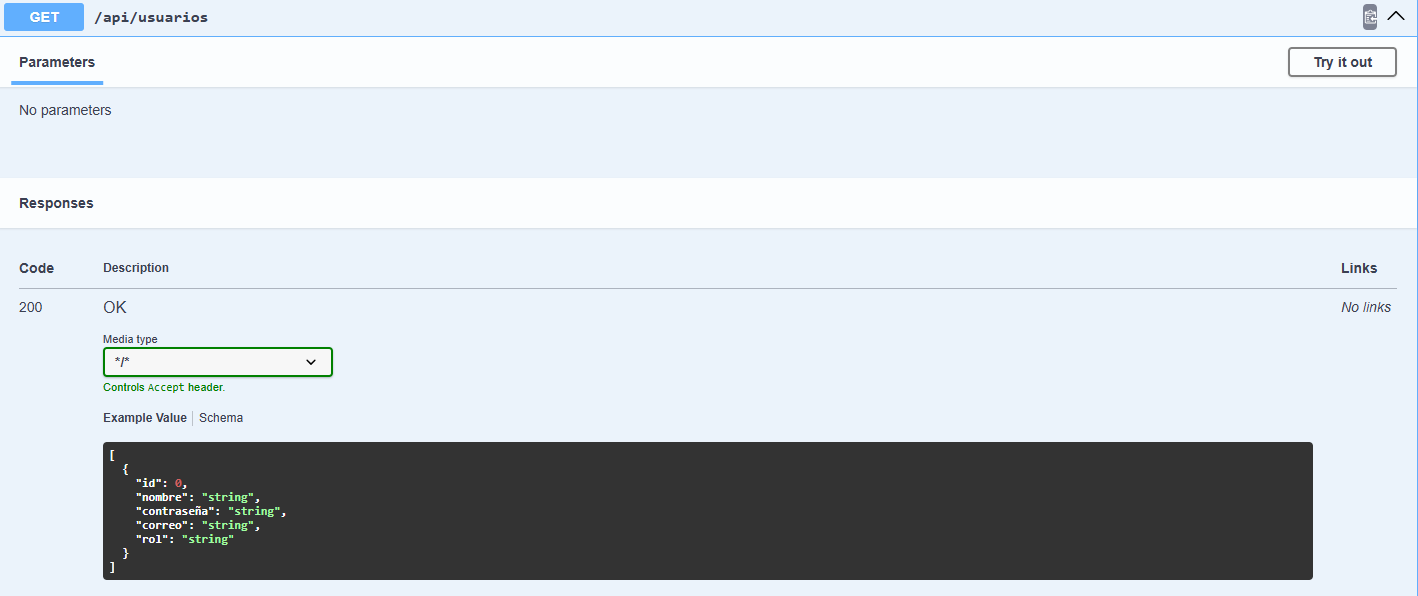
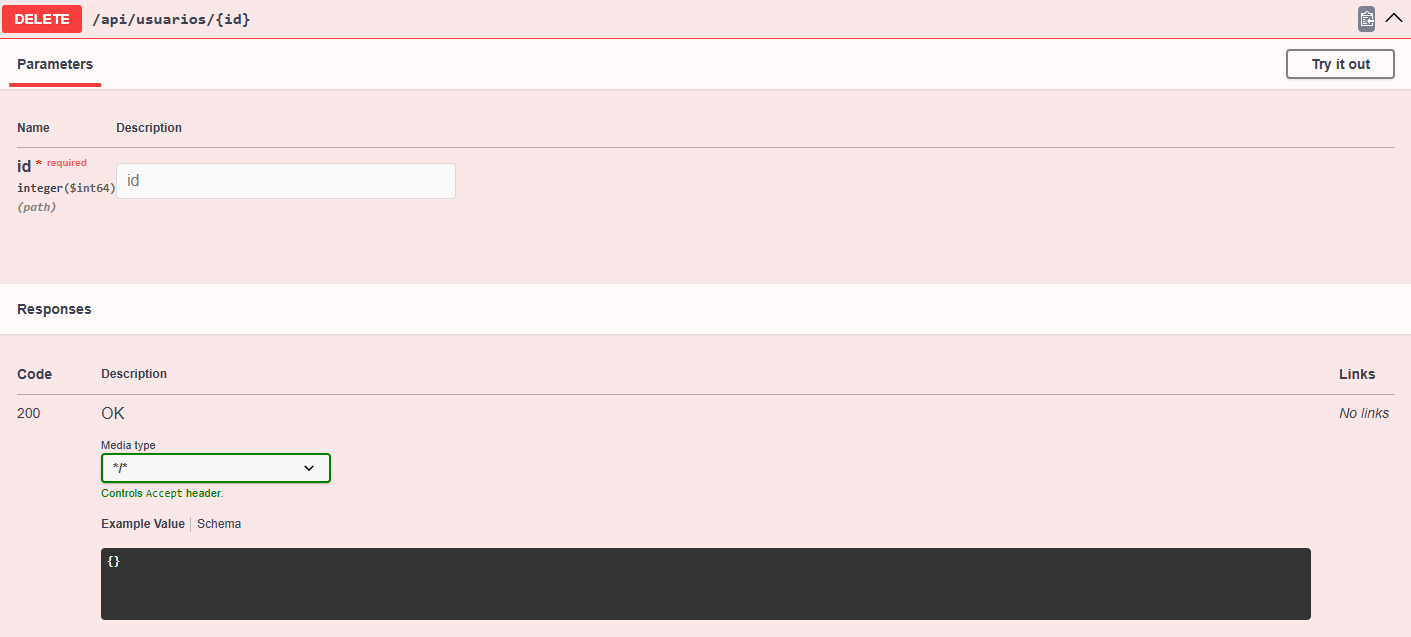
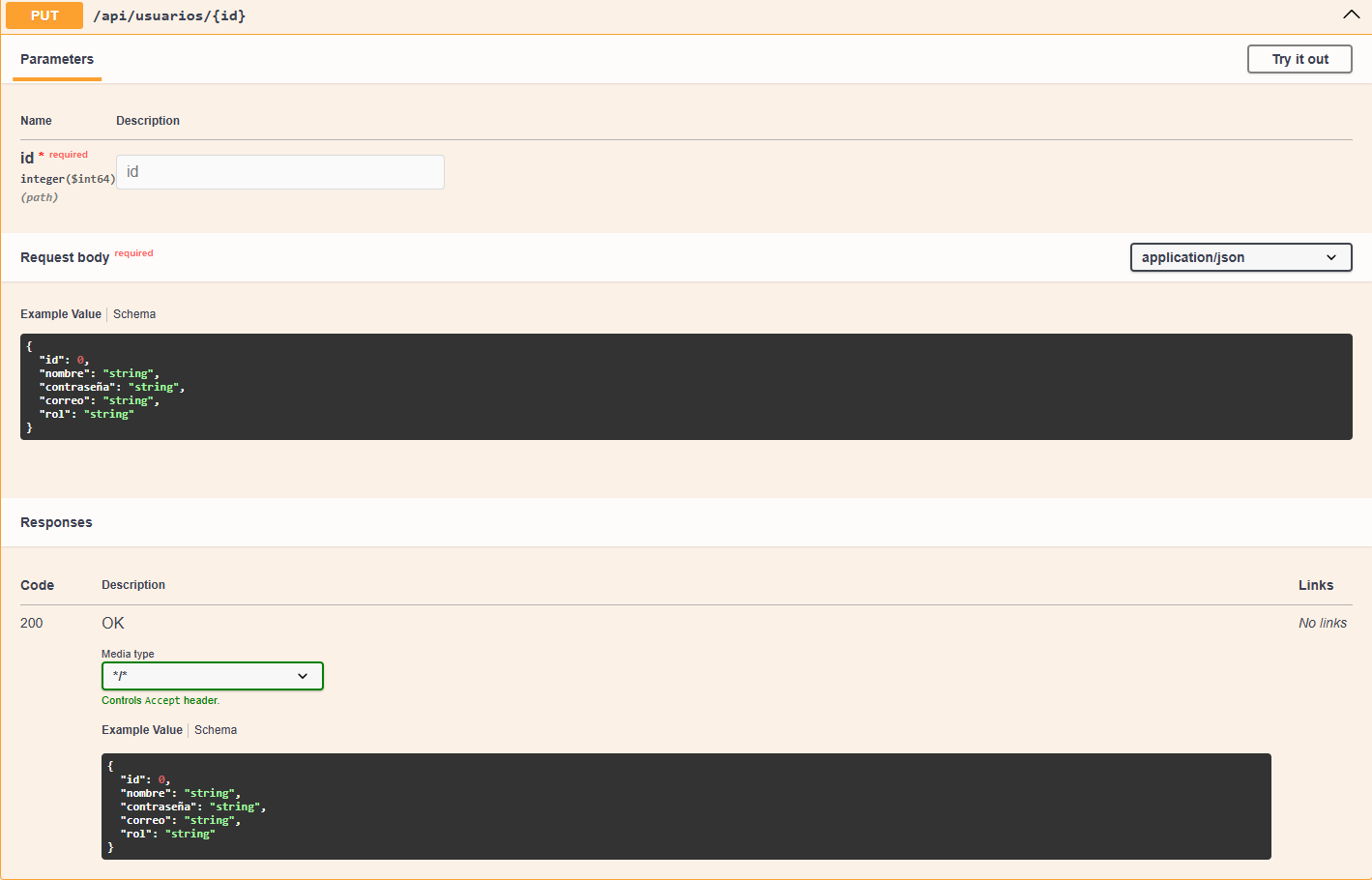
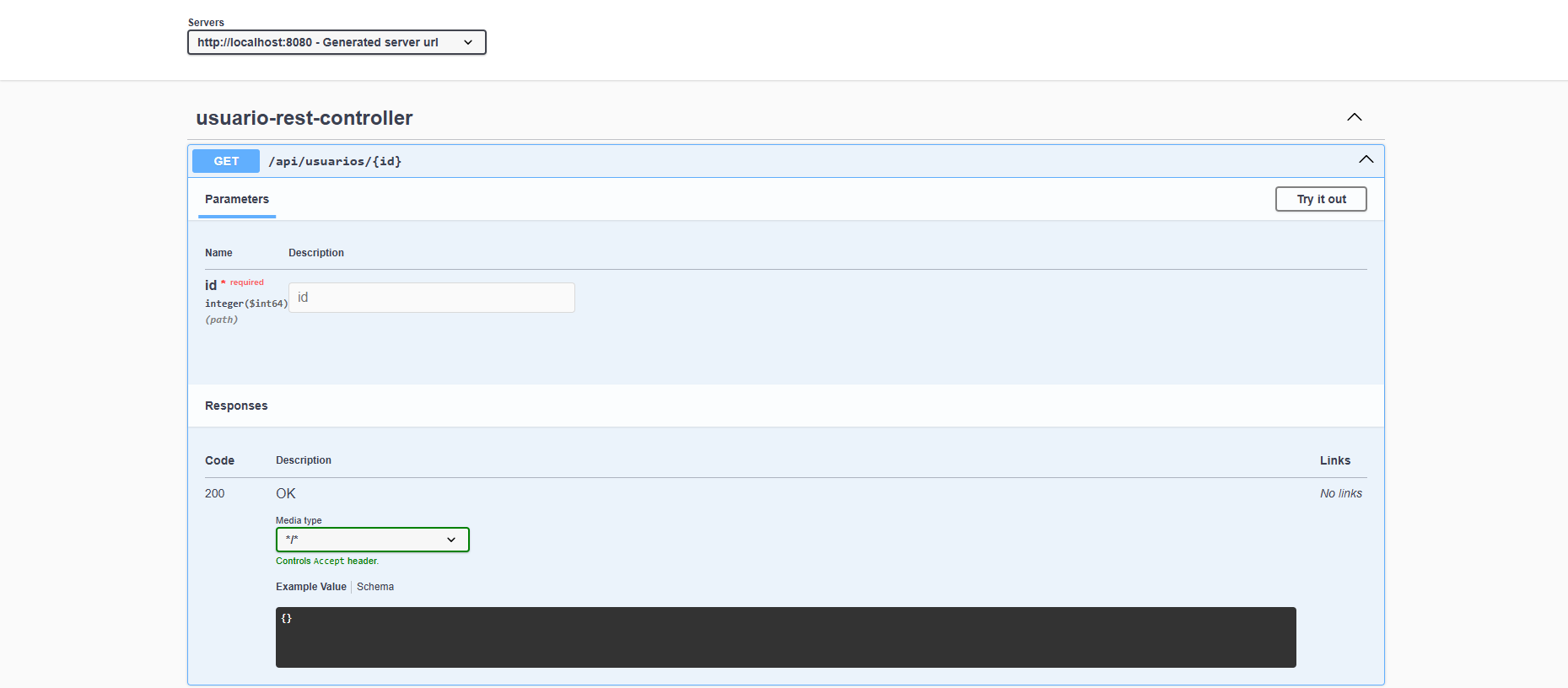
Interfaz de usuario gráfica

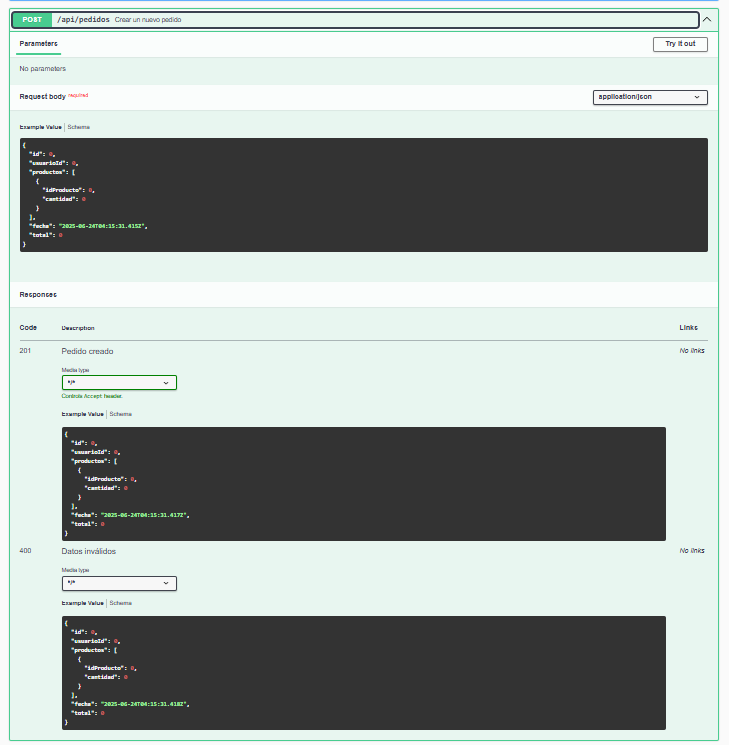
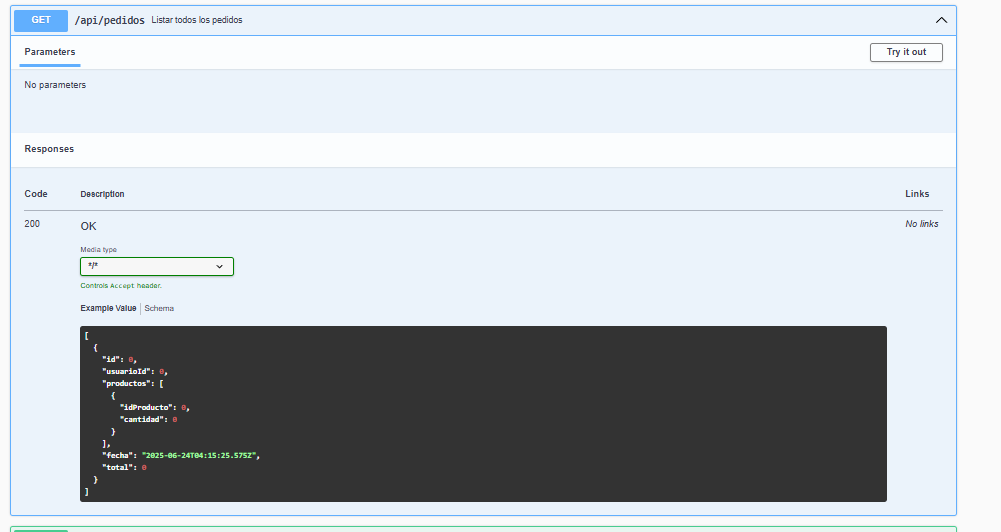
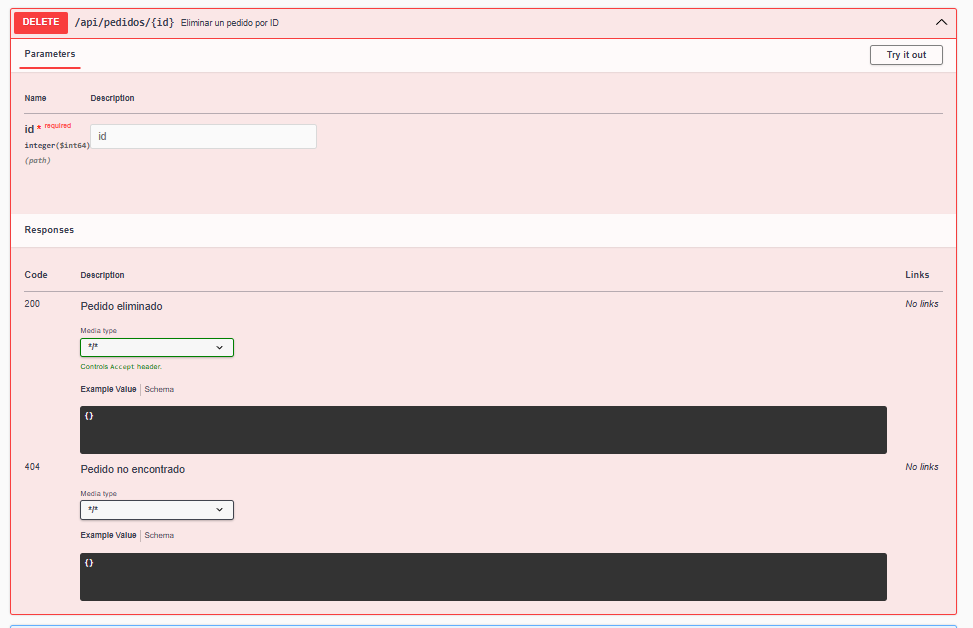
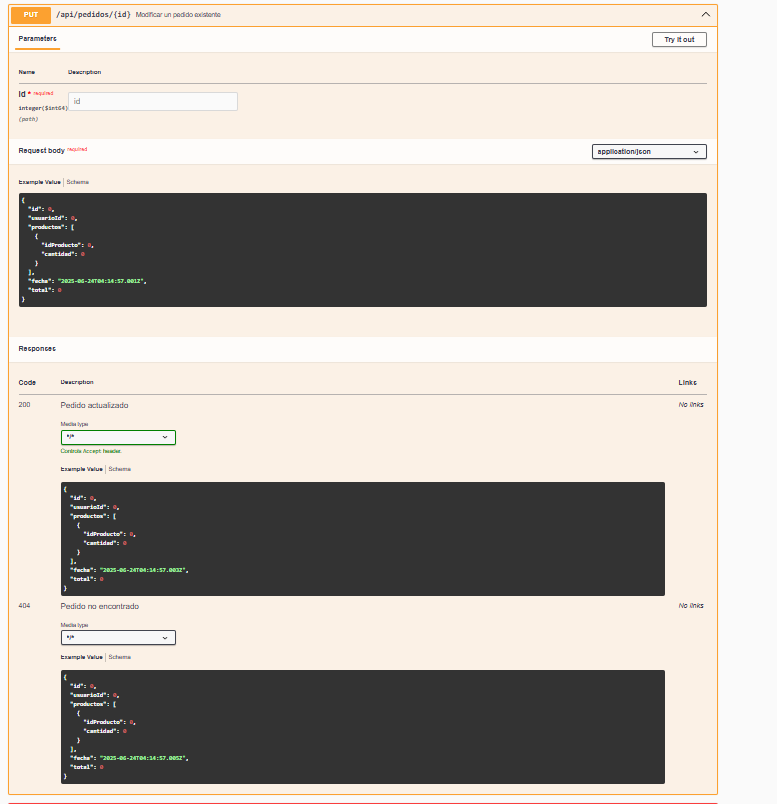
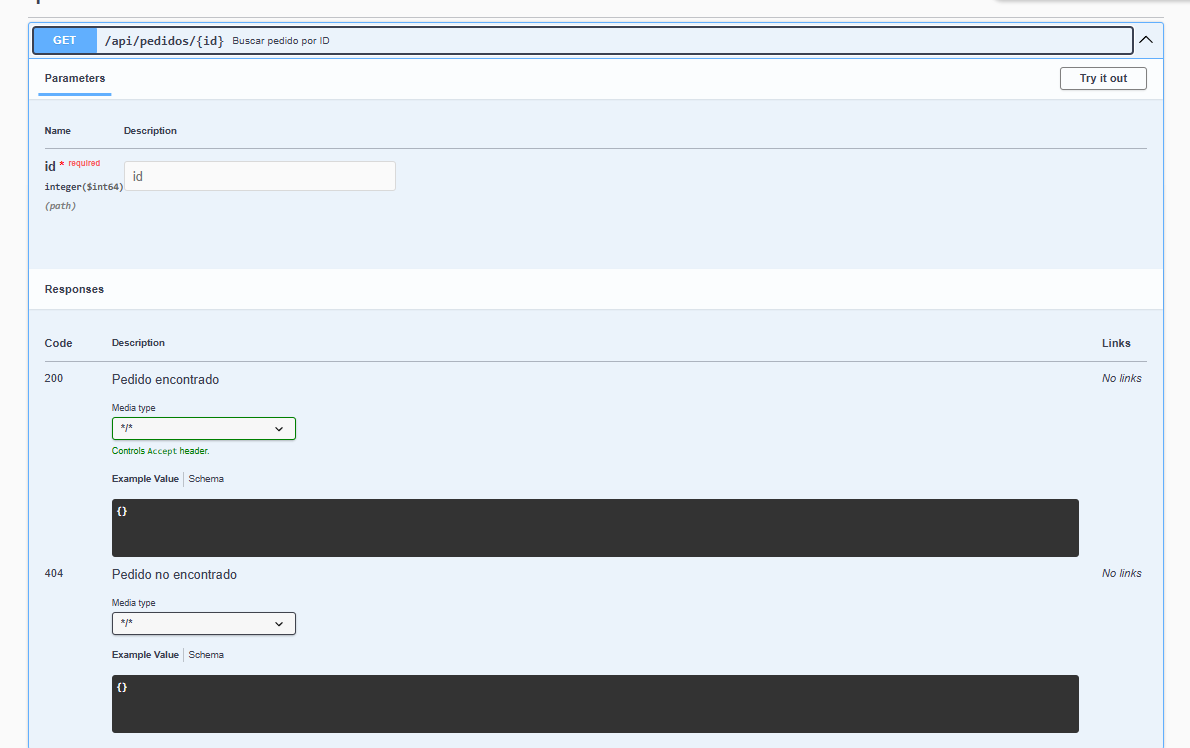
El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

-USUARIOS:



-PEDIDO

# 6. Conclusión

El proceso de implementación de pruebas con JUnit y Mockito permitió validar la lógica del sistema de microservicios de forma modular, segura y sin dependencias externas. Además, el uso de Git y GitHub favoreció un desarrollo colaborativo, versionado y organizado.  
  
Esta práctica garantiza que el sistema de EcoMarket SPA no solo está correctamente diseñado, sino también validado en su comportamiento esperado, permitiendo escalabilidad y mantenimiento a largo plazo.