

Universidad de Las Américas

Facultad de Ingenierías y Ciencias Agropecuarias

Ingeniería de Software

Informe de laboratorio

1. DATOS DEL ALUMNO:

Esteban Cueva

2. TEMA DE LA PRÁCTICA:

Implementación de Microservicios Core (Product & Order)

3. OBJETIVO DE LA PRÁCTICA

Desarrollar y desplegar microservicios independientes para la gestión de productos y órdenes dentro de un sistema de comercio electrónico, aplicando arquitectura de microservicios con Spring Boot y JPA/Hibernate.

4. OBJETIVOS ESPECÍFICOS

- Implementar product-service para la gestión de productos.
- Implementar order-service para la gestión de órdenes de clientes.
- Configurar cada microservicio con su propia base de datos y puerto independiente.
- Probar la funcionalidad de los microservicios mediante herramientas de prueba de APIs.

5. MATERIALES Y MÉTODOS

Materiales: Dentro del proyecto se utilizaron cosas como: Java 17, Spring Boot 3.5.7, Spring Data JPA, H2/MySQL, Maven

Métodos: Crear la estructura del proyecto dividiendo las carpetas por microservicio: product, service y order. Configurar el archivo application.properties de cada uno con su propia base de datos y puerto. Implementar las entidades, repositorios, servicios y controladores correspondientes. Finalmente, ejecutar y probar cada microservicio de manera independiente utilizando el comando:
mvn spring-boot:run

6. DESARROLLO DE LA PRÁCTICA Y RESULTADOS

Cada microservicio **se ejecuta en un puerto específico** configurado dentro de su archivo `application.properties`:

- **customer-service (puerto 8081)**: Gestiona la información de los clientes. Este servicio ya se encontraba implementado previamente.
- **product-service (puerto 8082)**: Microservicio independiente encargado de la gestión de productos de la tienda. Se implementó la entidad `Product`, que contiene información como el nombre, descripción, precio y stock disponible.
- **order-service (puerto 8083)**: Microservicio independiente para la gestión de órdenes realizadas por los clientes. Se creó la entidad `Order`, con campos como el ID del cliente, fecha de la orden y monto total.

Para acceder a la documentación Swagger de cada microservicio, se debe agregar al final del número de puerto la ruta:

swagger-ui/index.html

Ejemplo:

- `http://localhost:8081/swagger-ui/index.html`
- `http://localhost:8082/swagger-ui/index.html`
- `http://localhost:8083/swagger-ui/index.html`

Repositorio GitHub:

<https://github.com/EstebanCueva/Microservicios-Core/tree/main>

7. OPINIÓN PERSONAL

La **implementación de microservicios** ofrece una arquitectura modular que **separa claramente las responsabilidades** de cada componente dentro del sistema, lo que facilita tanto el **mantenimiento** como la **escalabilidad** de la aplicación. Cada microservicio puede **desarrollarse, desplegarse, actualizarse y probarse de forma independiente**, sin afectar al resto del sistema, lo que permite una **mayor flexibilidad y agilidad** en el ciclo de desarrollo.

Además, esta estructura favorece la **especialización de los equipos**, ya que cada grupo de trabajo puede enfocarse en un servicio concreto (por ejemplo, clientes, productos u órdenes), optimizando el tiempo de desarrollo y reduciendo la complejidad del código. En conjunto, el enfoque basado en microservicios **mejora la productividad**, facilita la **integración continua** y contribuye a una

organización más clara y eficiente del proyecto.

8. ANEXOS (Ai se requiere)