

## Instrucciones

Microservicio 1: *product-service*, debe administrar el catálogo de productos, el producto debe guardarse en una tabla H2.

Microservicio 2: *order-service*, gestiona una orden de compra, recibe la orden de compra y debe calcular el total. Debe guardar la orden y la relación de los productos de la orden en una base de datos H2.

Actividades:

1. Complete el código para que funcione de forma correcta con las anotaciones faltantes o correcciones pertinentes.
2. Implementar las operaciones CRUD del microservicio *product-service*: crear, actualizar, recuperar y eliminar productos y guardar la información en una tabla en memoria H2.
3. Utilizando la interfaz funcional del microservicio *product-service* `com.products.catalog.service.ProductCondition` genere un método que tenga como parámetro *ProductCondition* y que haga un filtrado dado el nombre del producto, y regrese la lista de productos con las coincidencias usando programación funcional. Exponga la API.
4. Utilizando un servidor Eureka configure los microservicios *product-service* y *order-service* para que se registren en el. Agregue las anotaciones y/o dependencias faltantes. Puede usar `spring initializr` para crear el proyecto.
5. Implemente el método `createOrder` de la interfaz `com.products.order.service.OrderService`, que recibe como parámetro el objeto `CreateOrderRequest`.
  - A. Calcule el total de la orden, el precio de un producto se encuentra en el catálogo de productos.
  - B. Guarde la orden y los productos de la compra. Con JPA asegúrese de guardar de forma correcta las relaciones entre las tablas de la orden y los item orden.
6. Corrija cualquier mala práctica presente en el código.

Requerimientos:

- \* Cuenta en GitHub para descargar el proyecto.
- \* Java 17.
- \* IDE de su preferencia
- \* Maven, mas reciente