

Prueba Técnica para Desarrollador Backend SpringBoot V2.0

Procedimiento

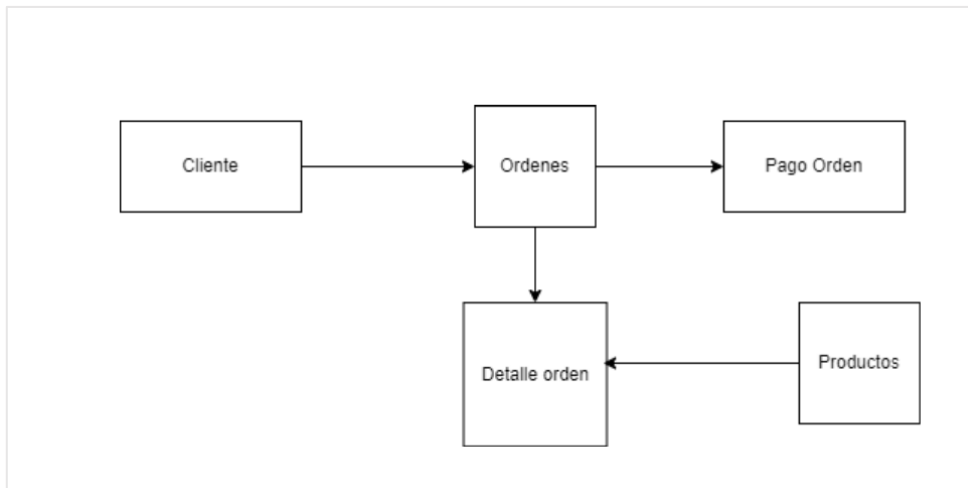
Desarrollar una solución basada en microservicios para un carrito de compras siguiendo las siguientes directrices:

- Puede usar el IDE de su elección
- Deberá implementar la solución utilizando Springboot. Se permite la elección de componentes adicionales a Springboot que se consideren más apropiados para el desarrollo.
- Creación de las APIs, distribuidas en distintos endpoints:
 - Endpoints Productos
 - Esta API funcionará como un proxy hacia la API externa: <https://fakestoreapi.com>. Su propósito será obtener la información de productos necesaria para el carrito de compras.
 - Endpoints Ordenes
 - Endpoints Pago
 - Para esta sección, se realizará una simulación del proceso de pago.
 - Clientes y Detalles de Ordenes.
 - Solo se requerirán las Entidades y Dtos para implementar el mecanismo de pago, no es requerido servicios, ni endpoints de Clientes o Detalle de Ordenes.
 - Endpoints Seguridad , Autenticación y Autorización (Opcional)
 - Este es un desarrollo que se valorará como puntos extras en la evaluación. Sin embargo, solo será evaluado si el resto de puntos ya han sido implementados

Estos endpoints se encargarán de gestionar las distintas funcionalidades del carrito de compras, desde la interacción con la información de clientes y productos, hasta la creación de órdenes simulación del proceso de pago.

Cabe destacar que el desarrollo de estas APIs se debe realizar mediante la utilización de componentes y tecnologías proporcionadas por el framework Spring, garantizando así un sistema robusto, seguro y eficiente.

Se agrega diagrama de ejemplo:



Revisaremos tu código y la aplicación en su conjunto, tomando en consideración los siguientes criterios:

1. Desarrollo de Microservicios en lenguaje Java, utilizando Spring Boot como base para la implementación.
2. Evaluaremos la estructura y organización del proyecto, destacando la importancia de una disposición coherente y lógica de los componentes para facilitar la mantenibilidad y comprensión del código.
3. Analizaremos la incorporación de patrones de diseño de software actualizados, asegurando que se utilicen prácticas modernas que optimicen la arquitectura y promuevan la escalabilidad del sistema.
4. Verificaremos la implementación de los principios SOLID, asegurando que los Microservicios sigan buenas prácticas de diseño y estén orientados a objetos de manera eficiente.
5. Evaluar el manejo y validación de errores, garantizando que el código pueda gestionar situaciones inesperadas de manera adecuada, mejorando así la robustez y confiabilidad de la aplicación.
6. Se verificará que el idioma utilizado para el nombrado de variables, funciones, comentarios y commits sea exclusivamente en inglés, asegurando consistencia y facilitando la colaboración internacional.

Este análisis exhaustivo abordará los aspectos clave para garantizar un código de calidad, una aplicación eficiente y una colaboración fluida en el equipo de desarrollo.

Entregables:

- Proporcionar el enlace del proyecto publicado en un repositorio de Git que contenga el código fuente completo de la solución. Preferiblemente, utilizar plataformas como GitHub, GitLab o BitBucket para la gestión eficiente del código.
- Adjuntar una colección de Postman que contenga todos los endpoints creados. Esto permitirá a los usuarios probar y validar las API de manera rápida y sencilla, garantizando la correcta funcionalidad de cada endpoint.
- Cumplir con el plazo máximo de entrega establecido, limitando la entrega del proyecto a un período de 4 días. Es fundamental gestionar eficientemente el tiempo para garantizar la prontitud y eficacia en la entrega del trabajo.

Estos requisitos asegurarán una presentación completa y profesional del proyecto, facilitando la comprensión, implementación y prueba por parte de los usuarios y colaboradores.