

Descripción de la prueba técnica: Diseño e Implementación de una API REST escalable

Contexto:

Como **Tech Leader** de una empresa tecnológica que gestiona un servicio de e-commerce, tu misión es diseñar e implementar una arquitectura de API REST eficiente y escalable que ofrezca soporte para gestionar una cesta de compras y la carga de inventarios desde un archivo CSV. Aparte de la implementación funcional de la API, se espera que propongas decisiones arquitectónicas clave, con un enfoque en la optimización, escalabilidad, seguridad y mantenibilidad del sistema.

Tu solución será revisada para evaluar cómo abordan el diseño arquitectónico, la separación de responsabilidades, la escalabilidad del sistema y cómo utilizas herramientas modernas de la plataforma Java.

Objetivo del proyecto:

Se requiere desarrollar una **API REST** con dos funcionalidades básicas para gestionar artículos de un sistema de e-commerce. El proyecto deberá estar gestionado bajo control de versiones con **Git**, permitiendo su consulta, descarga y ejecución por parte del equipo evaluador.

La prueba técnica se centrará no solo en la funcionalidad básica, sino en cómo organizas la arquitectura general, el manejo de errores, la modularidad del código, la seguridad (gestión de credenciales) y la optimización de las operaciones.

1. **Primera Funcionalidad - Cálculo de la Cesta de Compras:** La API debe exponer un método que reciba una lista de artículos (definidos por id, name y value), calcular el total acumulado de los artículos agregados a una "cesta de compras" y devolver dicho total junto con el valor calculado del **IVA**. Debe considerarse un enfoque eficiente y extensible para gestionar diferentes tipos de impuestos y potenciales variaciones de reglas fiscales.
2. **Segunda Funcionalidad - Importación de Artículos desde un CSV:** El segundo método de la API debe ser capaz de leer un archivo CSV que contiene la misma colección de artículos (con los campos id, name, value) y almacenar este archivo en una carpeta específica del sistema de archivos. Tras la carga, la API debe devolver el número de artículos en el CSV y la suma total de los valores.

Requisitos adicionales:

- Eficiencia en la lectura y escritura de archivos, asegurando que la operación pueda ser escalable en caso de manejar grandes volúmenes de datos.
- Manejo eficiente de errores (¿Qué sucede si el archivo tiene un formato incorrecto? ¿Cómo manejar fallos de E/S?).
- Modularización de la lógica para la importación y validación de archivos.
- **Escalabilidad y rendimiento:** Considera cómo tu API puede ser diseñada para escalar horizontalmente bajo condiciones de alta carga. ¿Se puede desacoplar alguna de las operaciones? ¿Es recomendable usar un patrón de eventos para la funcionalidad de carga de CSV?
- **Seguridad:** Se debe demostrar cómo gestionarías el almacenamiento seguro de credenciales dentro de la API. ¿Qué prácticas recomiendas para proteger los endpoints y evitar vulnerabilidades comunes (por ejemplo, autenticación, autorización, cifrado de datos sensibles)?
- **Documentación:** Debes incluir un archivo README.md en el repositorio donde se describa cómo configurar, ejecutar, y escalar el proyecto. También describe las decisiones arquitectónicas y su justificación.