

## Prueba Técnica desarrollador **Backend For FrontEnd**

### **1. Desarrollo de una API BFF:**

El proyecto se centra en una plataforma de inventarios que tiene un front end web y una API externa que proporciona información sobre productos de la empresa que se encuentran en inventario. Utiliza el lenguaje de programación que te resulte más cómodo, la base de datos es de libre elección relacional o no relacional. La API externa proporciona datos de usuarios ficticios. Tu tarea es desarrollar un Backend for Frontend (BFF) que actúe como intermediario para mejorar la eficiencia y la seguridad.

### **Swagger:**

- **Página detalle de producto:**
  - Desarrollar una interfaz Swagger que contenga el CRUD del producto.

### **Backend:**

- **API BFF:**
  - Implementar un Backend for Frontend (BFF) que actúe como intermediario entre el frontend y la API externa de la plataforma.
  - Desarrollar rutas en el BFF para obtener detalles de producto por ID, categoría y gestionar la autenticación.
  - Crear los endpoint correspondientes para poder eliminar un producto, crear un producto, editar un producto.
  - La API BFF debe manejar la autenticación con un token JWT.
  - Incluye pruebas unitarias para verificar el funcionamiento correcto de la API BFF.
- **Cache**
  - Implementar la lógica de caché para almacenar y recuperar detalles de producto Implementar un sistema de caché en el BFF para almacenar temporalmente información frecuentemente solicitada y reducir la carga en la API externa.

**A continuación, seleccione la respuesta correcta:**

- ¿Qué medidas tomarías para prevenir ataques de inyección SQL en tu código?
  - 1) ☒ Utilizar consultas preparadas o parámetros vinculados para separar los datos de los comandos SQL.
  - 2) Validar manualmente cada entrada de usuario.
  - 3) Desactivar la funcionalidad de entrada de datos en la aplicación.
  - 4) Utilizar cifrado SSL/TLS para proteger contra ataques SQL.
  
- Explica la diferencia entre autenticación y autorización en el contexto de una aplicación web.
  - 1) La autenticación y la autorización son términos intercambiables.
  - 2) ☒ La autenticación se refiere al proceso de verificar la identidad del usuario, mientras que la autorización determina qué acciones tiene permitido realizar un usuario autenticado.
  - 3) La autorización se ocupa de verificar la identidad del usuario.
  - 4) Autenticación significa determinar los permisos de un usuario.

**Responda basado en su experiencia y conocimientos las siguientes preguntas**

- 1) Describe cómo diseñarías una arquitectura escalable para una aplicación web que espera un aumento significativo en el tráfico.

**Respuesta Abierta:**

Utilizando microservicios, que escalen horizontalmente junto a un balanceador de carga. En una arquitectura cloud se puede utilizar EC2 y autoescalamiento o una arquitectura serverless con lambdas.

- 2) Háblame sobre un desafío técnico que hayas enfrentado en un proyecto anterior y cómo lo resolviste.

Tuvimos que actualizar un monorepo de Java 8 a Java 17, tuvimos muchas dificultades con la compatibilidad de librerías, pero iterando y leyendo documentación al respecto logramos la actualización.

**Respuesta Abierta:**

¿Puedes mencionar y explicar brevemente tres patrones de diseño que hayas utilizado en proyectos anteriores?

**Respuesta Abierta:**

MVC para desacoplar responsabilidades entre las capas, también he utilizado el patrón Singleton para obtener instancias únicas, Builder Pattern para inicializar clases complejas, DTO para intercambio de información, entre otros

**Caso de uso:**

La aplicación de validación de horas trabajadas en la compañía NTTDATA, presenta una falla recurrente:

La aplicación presenta caídas constantes, el equipo de soporte realiza la revisión de los tiempos de repuesta de los servicios BFF y observa un tiempo de respuesta de 16 a 20 segundos. Al revisar los logs evidencian un timeout presentado por los servicios **login y consultar horas**, igualmente los servicios llevan corriendo más de 60 días contantes. Dato adicional los servidores contienen un administrador de servicios Kubernetes y cada servicio maneja tres imágenes desplegadas.

¿Qué practica implementaría para solucionar el problema?

¿Qué Script de Kubernetes utilizaría, basado en el nombre de los servicios?

Intentaría programar reinicios periódicos y también un escalamiento automático en caso de estar teniendo delays debido a una saturación en los servicios