



Prueba Técnica para Desarrollador Backend – NestJS

Objetivo:

Desarrollar una API REST utilizando **NestJS** que simule la parte backend de una tienda virtual. El sistema debe gestionar productos, usuarios y transacciones, integrando una base de datos, autenticación y seguridad.

Requisitos Técnicos:

1. Back-End:

- Crear una API REST utilizando **NestJS**.
- Implementar controladores, servicios y módulos de manera organizada.
- La API deberá gestionar:
 - **Productos:** Crear, leer, actualizar y eliminar productos (CRUD).
 - **Usuarios:** Registro y autenticación de usuarios, usando **JWT**.
 - **Transacciones:** Registro de compras realizadas por los usuarios.
- La base de datos deberá ser **PostgreSQL** o **MongoDB**.

2. Autenticación y Seguridad:

- Implementar autenticación y autorización utilizando **JWT**.
- Proteger las rutas de la API con **guards** para que solo usuarios autenticados puedan acceder a determinadas funcionalidades (por ejemplo, compra de productos).
- Implementar políticas de seguridad: cifrado de contraseñas con **bcrypt**.

3. Procesamiento de Pagos:

- Integrar con **Mercado Libre**, **Stripe** u otro servicio de procesamiento de pagos.
- Validar el estado de las transacciones y notificar a los usuarios.

4. Pruebas:

- Escribir pruebas unitarias y de integración usando **Jest**.
- Incluir pruebas para los servicios de productos, usuarios y transacciones.

5. Extras (Opcionales pero valorados):

- Implementar funcionalidades en tiempo real usando **WebSockets** (por ejemplo, notificaciones de transacciones en tiempo real).
- Contenerizar la aplicación con **Docker**.
- Usar **Swagger** para documentar la API.



- Diseñar un frontend sencillo donde se puede ejecutar un proceso de pago de un producto

Entrega:

- Sube el proyecto a un repositorio en BitBucket.
- Incluye un archivo README con instrucciones claras sobre cómo instalar y ejecutar el proyecto.

Plazo:

- Tienes 2 días para completar la prueba.
- La entrega anticipada se considerará positivamente.

Evaluación:

Se evaluarán los siguientes aspectos:

- Correcta implementación y funcionalidad de la API.
- Estructura, claridad y calidad del código.(Clean Code, SOLID)
- Uso adecuado de las mejores prácticas en NestJS.
- Pruebas y documentación del proyecto.
- Creatividad y habilidades para resolver problemas adicionales, como la integración de pagos o uso de WebSockets.