



DOUBLE V
PARTNERS



NYX
A Double V Partners company

DOUBLE V PARTNERS NYX

11/2025

Desarrollador Back-End

Prueba Técnica

Unconventional People.
Disruptive Tech.

Estimado(a) candidato(a).

Nos enorgullece saber que Double V Partners NYX tiene el potencial de ser parte de tu proyecto de vida. Sabemos que las pruebas técnicas son desgastantes y algo pesadas, pero creemos que un buen proceso de selección es el que nos permite ver el potencial de una persona y también le permite al candidato(a) saber si este va a ser un trabajo retador, divertido, apasionante y acorde a tus expectativas. Date la oportunidad de hacer esto a conciencia no para pasar una prueba, sino para entender si este es el lugar adecuado para tu crecimiento profesional.

Instrucciones

- ❖ La siguiente es una prueba técnica donde tendrás completa libertad para proponer, idear, planear y plasmar tu filosofía de trabajo.
- ❖ La duración estimada para completar la prueba es de 2 días para terminarla a partir de la hora en la que la recibas.
- ❖ Archivo README.md con instrucciones para levantar el proyecto localmente.
- ❖ Incluir ejemplos de uso de los endpoints.

Adjunto los ejercicios de la prueba Técnica:

1. Operaciones CRUD Usuarios:

- Crear usuario
- Actualizar usuario.
- Obtener listado de usuarios.
- Obtener usuario por ID.

2. Operaciones CRUD Tickets:

- Crear un ticket.
- Editar un ticket.
- Eliminar un ticket.
- Obtener un ticket específico por su ID.
- Obtener una lista paginada de tickets.

3. Filtrado:

- Filtrar tickets por estatus (abierto/cerrado).
- Filtrar tickets por usuario.
- Permitir la combinación de filtros (estatus + usuario).

4. Modelo de Datos Usuario:

- id: Identificador único (UUID)
- nombres
- apellidos
- Fecha_Creacion: LocalDateTime.
- Fecha de Actualización: LocalDateTime.

5. Modelo de Datos Ticket:

- id: Identificador único (UUID).
- descripción (máx 500 caracteres).
- Usuario: Identificador del usuario que generó el ticket. O fecha Creación: LocalDateTime.
- Fecha de Actualización: LocalDateTime.
- status: Enum (ABIERTO, CERRADO).

6. Validaciones:

- Todos los campos obligatorios deben validarse.
- Fecha de creación y actualización deben tener un formato coherente.

7. Seguridad:

- Autenticación y autorización (JWT preferiblemente).
- Protección de endpoints sensibles.

8. Documentación:

- Documentar la API usando OpenAPI/Swagger.

Requerimientos No Funcionales

• Tecnologías a usar:

- Java Spring Boot
- NestJS
- Docker

• Código Limpio:

- ➔ Seguir principios SOLID.
- ➔ Manejo adecuado de excepciones.
- ➔ Respuestas HTTP.
- ➔ Uso de DTOs para las respuestas y solicitudes.
- ➔ Interceptores.

• Pruebas:

- ➔ Semi-Senior: Pruebas unitarias con JUnit.
- ➔ Senior: Pruebas unitarias y de integración.

- **Persistencia:**

- Usar Spring Data JPA con una base de datos en memoria (H2) para simplificar la prueba.

- **Caché:**

- Utilice algún proveedor de caché que permita guardar temporalmente los tickets de un usuario.

- **Buenas Prácticas:**

- Semi-Senior: Correcto uso de anotaciones (@RestController, @Service, @Repository, etc.).
 - Senior: Manejo de transacciones, caché y optimización de consultas. **!!!**

La prueba la tendrás que enviar al correo: welcome@doublevpartners.com con el asunto "Prueba Técnica + tu nombre".

!!!Mucho suerte y gracias por el tiempo que nos dedicas!!!