Reto de Codificación

# Posición: Backend Java Developer (Spring Boot + MySQL)

## Título: Sistema de Gestión de Candidatos en Proceso de Selección y Contratación

### Contexto:

## Desarrollar el backend de una aplicación que gestiona información de candidatos que postulan a proceso de selección y contratación utilizando MySQL como base de datos.

## La aplicación debe proporcionar una API REST utilizando Spring Boot con microservicios para realizar operaciones CRUD en la base de datos .

### Requerimientos:

#### Utiliza Spring Boot para iniciar el proyecto.

#### Configura Flyway para la gestión de la base de datos. Crea scripts SQL para la creación de la tabla candidates y carga de datos de prueba iniciales con al menos 5 registros.

#### Crea un modelo Candidates con al menos los siguientes campos: ID; name, email, gender, salary expected y otros campos relevantes.

#### Configura una API REST para realizar operaciones CRUD en el modelo Candidates.

### Puntos adicionales:

## Define una arquitectura hexagonal que incluya modelo de Dominio, capa Repository, capa Service

* + Define un modelo de dominio simple para representar a un candidato en el proceso de reclutamiento.
  + Implementa la capa de repositorio utilizando JPA para acceder a la base de datos.
  + Implementa la lógica de negocio en la capa de servicio, utilizando interfaces y clases concretas (Service01Impl)
* Controladores y Seguridad:
  + Implementa controladores REST protegidos con token JWT para las operaciones de creación y obtención de candidatos.
  + Utiliza Spring Security para gestionar la seguridad.
* Flyway y Datos de Prueba:
  + Configura Flyway para ejecutar los scripts SQL.
  + Asegúrate de que los datos de prueba iniciales estén presentes en la base de datos.
* Pruebas Unitarias e Integración:
  + Escribe pruebas unitarias e integración para las capas de repositorio y servicio.
  + Utiliza herramientas como JUnit y Mockito.

### Recomendaciones:

## Utiliza Java, Spring Boot, Spring Data JPA, Spring Security y Flyway.

## Usa un sistema de construcción como Maven o Gradle.

* Implementa la autenticación con tokens JWT.

## Incluye instrucciones claras sobre cómo configurar y ejecutar la aplicación.

## Asegúrate de que el código sea legible y siga las mejores prácticas de desarrollo.

## Considera la modularidad y reutilización del código.

## Proporciona comentarios en el código donde sea necesario.

### Duración del Reto:

## El reto debe ser entregado máximo 48 horas luego de haber recibido el reto por email.

### Entregables:

## Repositorio de GitHub con el código fuente.

## URL de la API desplegada en Heroku o plataformas similares, brindando acceso como colaboradores para revisar código desplegado.

* Documentación de los endpoints en Swagger
* Una colección en Postman con las llamadas a cada una de las APIs, incluyendo un par de casos ya grabados (uno con HTTP Status 200, exitoso y uno con 500, para caso de error) por cada endpoint.

Python/Django Dev con experiencia Cloud