

Prueba Técnica - Desarrollador Junior Java Spring Boot

Contexto

El candidato recibe un proyecto Java 17 con Spring Boot que expone una API para manejar entidades relacionadas con un colegio, el cual presenta las siguientes características:

- Entidades Materias y Alumnos ya implementadas (CRUD funcional).
- Entidad Notas definida en base de datos, pero no implementada en el proyecto.
- Existen bugs, por lo cual el proyecto no iniciara, realice los cambios necesarios hasta lograr que este funcione.

El candidato tendrá acceso al proyecto en un repositorio local y podrá abrirlo con VSCode, con conexión a la base de datos incluida.

Parte 1 - Detección y solución de bug (20 puntos)

- Detectar los errores que impiden que el proyecto inicie.
- Ejecutar el proyecto y demostrar que los CRUD implementados funcionan correctamente.

Parte 2 - Implementación CRUD para notas (50 puntos)

- Crear la entidad Notas con las anotaciones correctas para JPA.
- Crear repositorio NotaRepository.
- Implementar DTOs para entrada y salida.
- Implementar servicio NotaService con algun método que seleccione (create, get, update, delete y findAll).
- Crear controlador REST NotaController con el o los endpoints necesarios RESTful.
- Probar funcionalidad con curl o Postman.y si es posible con Swagger (ya implementado).

Parte 3 - Preguntas de Diseño de Sistemas (30 puntos)

Preguntas obligatorias:

1. Explica qué es una arquitectura de capas en una aplicación Spring Boot y responsabilidades de cada capa.
2. Si un cliente pide agregar paginación al listado de alumnos, ¿cómo lo implementarías?
3. ¿Cuál es la diferencia entre @Entity y @Table en JPA?
4. ¿Cómo manejarías el control de errores si un usuario solicita un registro de la entidad notas que no existe?
5. Si tuvieras que permitir que una materia tenga múltiples calificaciones y cada alumno pueda tener múltiples materias, ¿cómo modelarías esa relación en JPA?

Preguntas opcionales:

1. ¿Qué ventajas y desventajas tiene usar DTOs en lugar de devolver directamente la entidad JPA?
2. ¿Cómo organizarías los controladores si en el futuro hay que exponer la misma API para móviles y administración web?
3. ¿Qué es la escalabilidad vertical vs horizontal?
4. ¿Qué es un balanceador de carga y para qué sirve?
5. ¿Cuál es la diferencia entre JWT y Basic Auth?
6. ¿Qué es CORS y por qué es importante?