

Preguntas frecuentes para posición Java

Preguntas Generales

1. ¿Cuéntame de ti?
2. ¿Has trabajado con TDD? ¿Qué es para ti TDD?
3. ¿Para qué sirve la arroba transaccional, cuéntame cómo lo has utilizado en tu experiencia?
4. ¿Que es el garbage connector y en tu experiencia para que es útil?
5. ¿Cual es tu experiencia con el API stream de Java?
6. En tu experiencia, ¿Cual es la diferencia entre un set y un list?
7. ¿Con qué frameworks trabajas en Java? ¿Conoces Spring Boot?
8. ¿Qué es lo principal que usas con spring boot?
9. ¿Qué es lo que más te gusta y menos te gusta de Java y Spring Boot?
10. ¿Cómo sueles manejar las excepciones y errores en Spring Boot?
11. ¿Con cuáles métodos HTTP has trabajado?
12. ¿Qué cliente HTTP recomiendas para spring boot?
13. ¿Has trabajado con scopes?
14. ¿Conoces las pruebas de mutación?
15. ¿Qué le mejorarías a Java?
16. ¿Con cuáles principios SOLID sueles trabajar?
17. ¿Cómo lo has implementado JWT?
18. ¿Tienes experiencia con cross domain?
19. ¿Cómo asegurarias un servicio contra ataques de fuerza bruta?
20. En tu experiencia, ¿Cuando usas bases de datos relacionales o no relacionales?
21. ¿Sabes que es idempotencia? ¿Cómo lo has trabajado?
22. ¿Qué patrones de diseño conoces? ¿Con cuales has trabajado?
23. ¿Has escuchado de Clean Architecture? ¿Cómo lo has implementado (si lo has hecho)?
24. ¿Cual es tu experiencia con implementar métodos en interfaces?
25. ¿Qué principios de clean code aplicas en tu trabajo?
26. ¿Qué ventajas y desventajas tiene Git según tu experiencia?
27. ¿Has trabajado con el servicio Rest?
28. ¿Con cuáles tipos de relaciones has trabajado en JPA?
29. En tu experiencia, ¿Cuál es la diferencia entre JPA y Hibernate?
30. ¿Con qué métodos seguros trabajas usualmente?
31. ¿Con qué tipos de excepciones tienes experiencia?
32. ¿Has trabajado con arquitectura hexagonal? Si es así, ¿cuál desventaja y ventaja has visto de usarla?
33. ¿Cómo utilizas Valid y Validated?
34. ¿Cual es tu experiencia con polimorfismo?
35. En tu experiencia, ¿Cuál es la diferencia entre spring y spring framework?
36. ¿Cuál ha sido tu mayor éxito profesional?

37. ¿Cómo te mantienes actualizado en las nuevas versiones de Java?
38. ¿Cómo enfrentas desafíos técnicos?
39. ¿Cómo te aseguras que tus compañeros entiendan tus necesidades técnicas?
¿Cómo las transmites a los demás?
40. ¿Has tenido conflictos con compañeros? ¿Y cómo lo resolviste?
41. ¿Cómo crees que se puede generar un ambiente de desarrollo y mejoramiento continuo?
42. ¿Cuál consideras que es tu principal fortaleza y debilidad en habilidades blandas?
43. ¿Cómo motivas a los demás a seguir una buena idea que tengas?
44. ¿En qué país estás? ¿Te adaptarías a nuestro horario? (Horario de Ecuador aproximadamente de 8:30AM-5:30PM).

Sobre Seniority:

*Normalmente se pide un seniority entre Semi Senior o Senior, por lo que es importante demostrar confianza y en caso de no serlo, demostrar que estás aprendiendo y trabajando para llegar a un mayor seniority pronto.

1. ¿Cómo te autoevaluarías tú? Junior, Semi-Senior o Senior?
2. ¿Qué consideras que te faltaría para ser Senior?
3. ¿Has tenido la oportunidad de orientar a otros desarrolladores?
4. Dame un ejemplo en el que influiste positivamente en la dirección de un proyecto.

Preguntas sobre el ejercicio práctico:

Al compartir este se pedirá explicar que se creó desde una perspectiva no técnica y luego compartir la parte técnica. Además se probará que funcione todo lo requerido.

1. ¿Qué arquitectura implementaste?
2. ¿Qué consultas tienes?
3. Muestrame una prueba unitaria y explicame que hiciste.
4. ¿Cuál es la diferencia entre una prueba unitaria o una prueba de integración? ¿Cómo las implementaste?
5. Muestrame donde hiciste un interfaz repositorio donde hayas implementado JPQL (si lo hiciste).
6. Muestra donde mapeaste un Entity Bean.
7. ¿Qué le mejorarías a tu código?
8. ¿Qué validaciones hiciste?
9. ¿Usaste componentes pre fabricados? Si es así, ¿por qué?
10. ¿Cómo aseguraste endpoints en tu ejercicio práctico (si lo hiciste)?
11. ¿Cómo implementaste las entidades de bases de datos?
12. ¿Podrías mostrar una interfaz del repositorio?
13. ¿Manejas transaccionalidad en tus servicios? Si no lo hiciste, ¿cómo lo implementarías?

14. ¿Cual crees que fue la mejor parte de tu código? ¿Lo que consideras que hiciste mejor?
15. ¿Aplicaste validaciones de entrada?