

## Preguntas

1. Diferencia entre poo y p funcional
2. Pilares de POO (ejemplos)
3. Diferencias entre clase abstracta vs interfaz, interfaz vs clase
4. Estructuras de datos java / colecciones (list, map), string
5. Manejo de excepciones
6. ¿Qué es lo nuevo que trae java 8, java 11?
7. ¿Qué paradigmas de programación conoces?
8. ¿Qué es programación funcional? Contar experiencia en ello
9. ¿Qué te permite realizar un api stream?
10. ¿en qué te beneficia el uso de expresiones lambda?
11. Diferencias entre map y flatmap. ejemplo
12. Lambda en java. de qué forma lo has usado.
13. ¿Has trabajado con spring boot? comentar experiencia, versiones usadas
14. ¿qué entiendes por starters?
15. ¿Qué diferencia hay entre spring framework y spring boot?
16. ¿qué tipo de anotaciones/ estereotipos has utilizado en spring boot?
17. ¿Qué es bean en sprintboot?
18. spring cloud: config, eureka, api gateagay
19. ¿Has visto Programación asincrónica y síncrona?
20. ¿Qué entiendes por programación reactiva?
21. ¿Qué framework has usado para programación reactiva?
22. Contar experiencia sobre el uso de p. reactiva en sus proyectos
23. ¿qué me puedes comentar de web flux? mono y flux
24. p. reactiva vs p funcional
25. ¿bajo qué escenario recomiendas usarlo la p reactiva?
26. ¿Con que tipo de arquitectura has trabajado?
27. ¿Has desarrollado microservicios? comentar experiencia
28. ¿qué problema has resuelto usando microservicio?
29. pros y contras de uso de microservicios
30. en un escenario de 3 o más microservicios que complejidad le ves
31. que es taxonomía y semántica de microservicios
32. principio contract first
33. ¿qué es un contrato, como lo hacías?
34. ¿Has usado swagger/open api?
35. ¿cuándo usas get, post, put, patch?
36. Pruebas unitarias, ¿qué herramientas has usado?
37. ¿Qué patrones has usado en pruebas unitarias?
38. Diferencias entre los patrones usados para las PU
39. ¿Has trabajado con junit5 y mockito, cual es la diferencia entre mock, spy?
40. ¿En una prueba unitaria como validabas, que método, valor, y otro que no devuelva nada?
41. ¿Qué es cobertura de código?
42. ¿Has trabajado con bd relacionales/ no relacionales?
43. en que escenario recomiendas usar bd relacional o no relacional
44. ¿podrías usar índices en mongodb?
45. sobre sql server has modelado datos, normalización
46. ¿Has creado store procedure?
47. ¿qué herramientas de CI/CD has usado?
48. ¿has podido crear pipeline en azure devops o Jenkins?
49. ¿has integrado jenkins con sonarqube?
50. ¿para qué usas sonarqube?

51. ¿qué entiendes por cobertura de código?
52. ¿qué hace fortify?
53. ¿qué modelo usaste para control de cambios de código? /gitflow
54. ¿cómo manejas el control de versiones
55. 3 comandos git
56. ¿Has trabajado con PR?
57. ¿qué comandos conoces o has usado de git?
58. ¿Has trabajado con marcos ágiles o waterfull?
59. Lineamientos y prácticas de desarrollo seguro utilizado
60. Casos de uso de SOLID
61. Casos de uso de patrones de diseño de microservicios
62. Patrones de diseño de software: creacionales, estructurales y comportamiento
63. factory method vs abstract factory
64. seguridad en ms: jwt, encriptación, ofuscación, spring security
65. ¿has trabajado con kibana?
66. ¿Has realizado el monitoreo de microservicios?
67. Por qué crees que es importante las pruebas de rendimiento
68. docker vs kubernetes?
69. como arranges un contenedor, usas el docker run o docker up
70. como borras un contenedor?
71. conoces comandos de kubernetes: kubctl
72. has usado Kafka. ¿qué es apache Kafka?
73. ¿cuál es el papel de zookeeper?
74. ¿qué es idempotencia en Kafka?
75. jpa vs hibernate
76. en arquitectura de ms como aseguras la disponibilidad
77. como registrabas los logs con las apis
78. ¿qué niveles de logs conoces?
79. cuando usas un info vs un error
80. ¿has llegado a crear ms desde cero?
81. ¿qué entiendes por orquestación de microservicios?
82. ¿has realizado llamadas en paralelos a microservicios?
83. paralelismo vs concurrencia
84. ¿cómo documentabas las apis?
85. ¿porque es importante documentar el api?
86. ¿sobre p funcional que operadores has usado?
87. ¿cuántos aprobadores tenías de los PR?
88. ¿has hecho pruebas de rendimiento?
89. ¿estas pruebas que me permite validar?
90. ¿cómo funciona el patron observer?
91. ¿has usado lombok, checkstyle?
92. ¿has hecho monitoreo en producción?
93. ¿has trabajado con herramienta para memoria caché?
94. ¿Cómo consideras que fue tu curva de aprendizaje para adaptarte en java?
95. ¿Qué es lo más complicado al ingresar a un nuevo equipo?
96. ¿en qué tipo de cliente has estado?
97. ¿qué complicaciones has tenido con trabajo remoto?
98. ¿cómo te fue la interacción cuando estas en modo remoto, dudas o como lo has manejado?
99. ¿Cómo manejas las dependencias o bloqueantes en los proyectos ágiles?