

Taller de Investigación

Fundamentos necesarios antes de implementar una API con Spring Boot

● BLOQUE 1 – Fundamentos Web y APIs

1. ¿Qué es una API y cuál es su propósito dentro de una arquitectura de software?
 2. ¿Qué diferencia existe entre una API REST y una aplicación web tradicional que devuelve HTML?
 3. ¿Qué es un protocolo de comunicación?
Investigue al menos dos ejemplos y explique cuál se usa en las APIs web.
 4. ¿Qué es HTTP y cuál es su función en el modelo Cliente-Servidor?
 5. ¿Quién transporta los datos en una comunicación web?
Explique el papel del protocolo HTTP y del formato de datos (por ejemplo, JSON).
 6. ¿Qué es una petición HTTP (HTTP Request) y cuáles son sus partes?
 7. ¿Qué es una respuesta HTTP (HTTP Response) y qué información contiene?
 8. Explique los métodos HTTP más usados en APIs REST:
GET, POST, PUT y DELETE.
¿Qué tipo de operación representa cada uno?
 9. ¿Qué son los códigos de estado HTTP?
Investigue y explique los siguientes: 200, 201, 400, 404 y 500.
 10. ¿Qué es JSON y por qué es el formato más utilizado en APIs modernas?
-

● BLOQUE 2 – Fundamentos de Spring y Spring Boot

11. ¿Qué es el Spring Framework y qué problema intenta resolver en el desarrollo empresarial con Java?
12. ¿Qué es Spring Boot y en qué se diferencia del Spring tradicional?
13. ¿Qué es un servidor embebido y cuál utiliza Spring Boot por defecto?
14. ¿Qué es el contenedor de Spring (ApplicationContext) y qué es un Bean?
15. Explique qué es Inversión de Control (IoC) y cómo se aplica en Spring Boot.
16. ¿Qué es Inyección de Dependencias (DI) y qué ventajas ofrece frente a crear objetos con `new`?

● BLOQUE 3 – Anotaciones Fundamentales en Spring Boot

17. Investigue y explique el propósito de las siguientes anotaciones:

- @SpringBootApplication
- @RestController
- @RequestMapping
- @GetMapping
- @PostMapping
- @PathVariable
- @RequestBody

18. ¿Cuál es la diferencia entre @Component, @Service y @Repository?
¿Por qué existen si técnicamente hacen algo similar?

● BLOQUE 4 – Arquitectura en Capas

19. ¿Qué es la arquitectura en capas (Layered Architecture)?
Explique el rol de las siguientes capas en una API:

- Controller
- Service
- Repository
- Model

20. ¿Por qué no es buena práctica colocar toda la lógica dentro del Controller?
Explique los problemas que se generan cuando no se separan responsabilidades.