

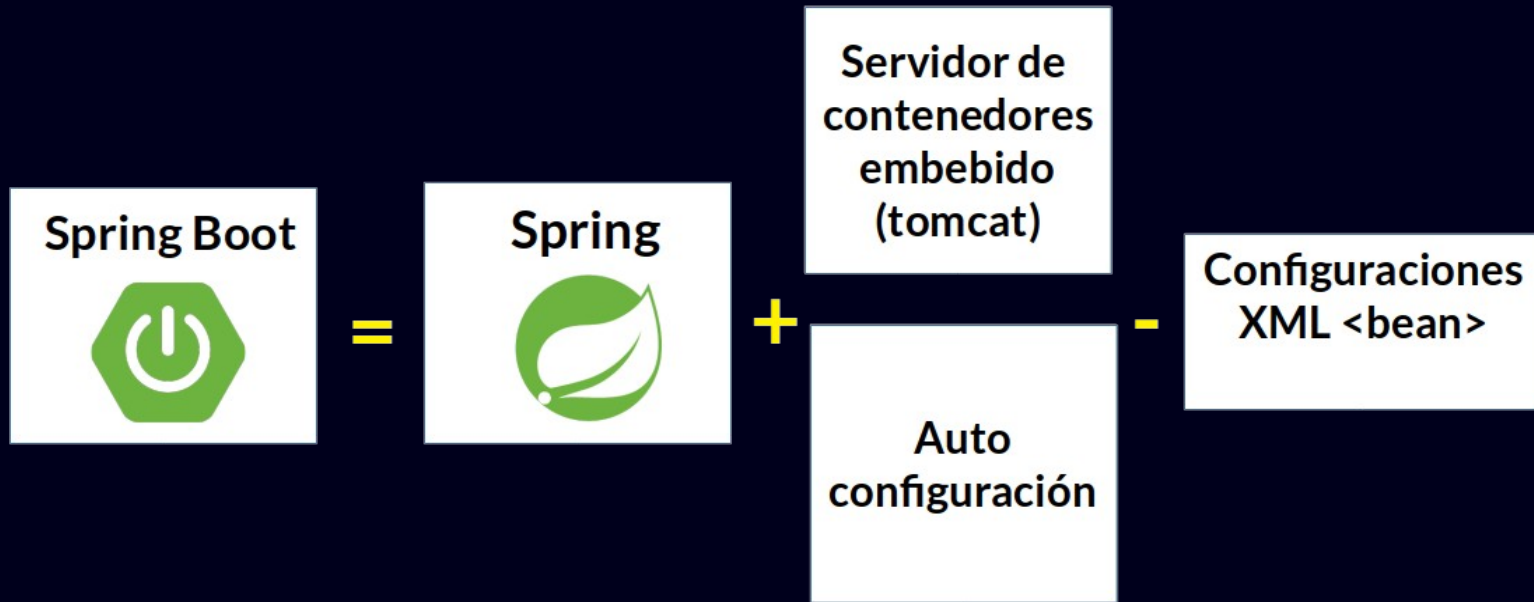
# Top 20 preguntas

En entrevistas para desarrollador  
sobre **Spring Boot**



# ¿Cuál es la diferencia fundamental entre Spring & Spring Boot?

# 01



# ¿Qué es un bean?

# 02

- **Es una pieza de software que representa un objeto dentro de Spring.**
- Son manejados por el contenedor de Spring.
- **Tienen un ciclo de vida.**
- Son reusables.

# ¿Qué es el ámbito de un bean ?

# 03

- Define el ciclo de vida y visibilidad del bean en el contexto.
- Se define con la anotación @Scope

- **singleton**
- prototype
- request
- session
- application
- websocket

→ Por defecto y rara vez será necesario usar uno diferente.

# ¿Qué son los starters y cuál es su objetivo ? 04

- Un starter es una utilidad que facilita la creación y configuración de aplicaciones.
- Agregan dependencias necesarias para un objetivo y parámetros de configuración.

Ejemplos:

`spring-boot-starter-web`

`spring-boot-starter-data-jpa`

# ¿Cuáles son los principales verbos HTTP y cómo se pueden implementar con Spring Boot ?

# 05

- GET → • @GetMapping
- POST → • @PostMapping
- PUT → • @PutMapping
- DELETE → • @DeleteMapping
- PATCH → • @PatchMapping

# ¿ Qué diferencias existen entre JPA y Hibernate ?

06

- Java Persistence API define una especificación o interfaces para manejo de persistencia.
- Hibernate es una implementación particular de JPA (y es la usada por defecto en Spring Boot)

# ¿ Qué es Spring Initializr ?

# 07

- Es una plataforma online para generar un proyecto base se Spring Boot.
- Se puede usar por la web directamente o desde los principales IDEs como STS o IntelliJ.



# ¿ Qué diferencias existen entre @Repository, @Component, @Service & @Controller ?

# 08

- No existen diferencias funcionales, solo semánticas.
- Son anotaciones para marcar “clases” en Spring y que se crean beans a partir de ellas.

# ¿ Explique la anotación @RestController ?

# 09

- Facilita la creación de API REST usando Spring MVC.
- Combina @Controller + @ResponseBody
- Permita serializar cualquier respuesta de un controlador Spring MVC directamente a json o xml.

# ¿Cuál es la función de Spring AOP ?

# 10

- AOP significa programación orientada a aspectos.
- Permite crear “aspectos” en nuestras aplicaciones para ser usados de forma transversal.

## Ejemplo:

Crear una anotación que permita auditar la realización de determinadas acciones.

# ¿ La anotación @Profile que función tiene y porque es importante ?

# 11

- @Profile permite usar beans particular en dependencia del perfil activo en una aplicación Spring Boot.
- Los perfiles permiten “instanciar” de forma particular una aplicación en dependencia de un escenario, por ejemplo: dev, qa, prod, etc.

# ¿ Qué hace la anotación `@SpringBootApplication` internamente ?

# 12

- Anotación principal de Spring Boot, ubicada por defecto en la clase principal de un proyecto Spring Boot.
- Combina las anotaciones `@Configuration`, `@EnableAutoConfiguration` and `@ComponentScan`.

¿ Qué patrones están presentes en Spring /  
Spring Boot, agregue algún ejemplo ?

13

- Creacional: Singleton
- Estructural: Proxy
- Comportamiento: Cache, Template

### Ejemplo:

Al crear un bean anotando una clase con  
`@Service` y scope por defecto usamos Singleton.

# ¿ Para que se usa @Transactional ?

# 14

- En BBDD una operación es “transactional” se mantiene el estado en un sistema consistente.
- @Transactional permite ejecutar operaciones en Spring (métodos) en forma “todo o nada”.
- Si algo sale mal durante la ejecución de una operación @Transaccional el sistema se \*deja en su estado inicial.

# ¿ Qué es la inyección de dependencias y como la implementa Spring ?

# 15

- DI es un patrón para separar responsabilidades, evitando que una clase sea responsable de instanciar objetos de otra. Promueve el bajo acoplamiento.
- Al arrancar Spring se crean los objetos (beans), y luego se “inyectan” donde son necesarios usando @Autowired.



¿ Cuáles formas de inyección de dependencias que existen en Spring y cuál es la más adecuada ?

16

- Inyección por atributo.
- Inyección por setter.
- Inyección por constructor ←

# ¿ Para que se usa @Value y que lenguaje usa la anotación para su función ?

# 17

- @Value permite leer configuraciones y colocarlas en la propiedad anotada.
- Usa el lenguaje SpEL.

Ejemplo:

```
@Value("${value.from.file}")  
private String valueFromFile;
```

# ¿Cuál es el puerto por defecto de una aplicación Spring Boot y cómo podemos cambiarlo ?

# 18

- Por defecto el puerto es 8080
- Se cambia en la propiedad *server.port* del fichero de configuración.

# ¿ Qué es Spring Actuator ?

# 19

- Librería que proporciona herramientas de monitoreo y administración para un API REST.
- Habilita varios endpoints para ser usados, ej: /beans, /health, /shutdown.
- Permite ser extendido.

# ¿Qué es Spring Cloud?

# 20

- Spring Cloud es un proyecto basado en Spring Boot.
- Implementa soluciones “production-ready” para el desarrollo de microservicios.
- Algunos componentes son:
  - Spring Cloud Config.
  - Spring Cloud Gateway-
- Esta influenciado por Netflix OSS.

**¡ YA ESTAS LIST@ !**

¿ Quieres aprender más ?

En un par de semanas estrenaremos curso práctico gratuito de **Spring Boot**.

