

Spring Boot es un framework de código abierto que facilita el desarrollo de aplicaciones Java con una configuración mínima. Se construye sobre Spring Framework, proporcionando un enfoque simplificado para la creación de aplicaciones independientes y de producción listas para ejecutarse. Spring Boot permite a los desarrolladores enfocarse en la lógica del negocio sin preocuparse por la configuración y la infraestructura subyacente.

Spring Boot ofrece una serie de características que facilitan el desarrollo de aplicaciones. Proporciona una configuración automática que detecta las bibliotecas presentes en el classpath y configura automáticamente los beans y las propiedades necesarias. Esto reduce significativamente la cantidad de configuración manual que los desarrolladores deben realizar. Además, Spring Boot incluye un servidor web embebido, como Tomcat o Jetty, lo que permite ejecutar aplicaciones como servicios independientes sin necesidad de un contenedor de aplicaciones externo. Esto facilita el desarrollo, la prueba y el despliegue de aplicaciones.

El uso de Spring Boot también incluye un soporte robusto para perfiles de configuración, permitiendo definir configuraciones específicas para diferentes entornos (desarrollo, prueba, producción) y cambiar fácilmente entre ellos. Además, proporciona herramientas de monitoreo y gestión integradas a través de Actuator, permitiendo a los desarrolladores monitorear el estado de sus aplicaciones, ver métricas, inspeccionar configuraciones y realizar tareas administrativas. Spring Boot también se integra fácilmente con una amplia gama de bases de datos, sistemas de mensajería y otros servicios a través de dependencias de “starter”, que son conjuntos preconfigurados de dependencias y configuraciones.

Spring Boot es ideal para la creación de microservicios, servicios RESTful, aplicaciones web y aplicaciones empresariales. Su capacidad para simplificar el desarrollo y despliegue de aplicaciones lo convierte en una opción popular para proyectos de todos los tamaños. Además, su integración con Spring Cloud facilita la construcción de sistemas distribuidos y microservicios.

Spring Cloud: Descripción y Usos

Spring Cloud es un conjunto de herramientas y frameworks que extienden las capacidades del Spring Framework para facilitar el desarrollo de sistemas distribuidos y microservicios. Se enfoca en proporcionar soluciones a los desafíos comunes que enfrentan las arquitecturas de microservicios, como la configuración distribuida, el descubrimiento de servicios, el balanceo de carga, la tolerancia a fallos, la gestión de sesiones y la comunicación entre servicios.

Spring Cloud es esencial para la creación de sistemas distribuidos robustos y escalables. Facilita la implementación de patrones de microservicios y proporciona herramientas para abordar los desafíos comunes en estas arquitecturas, como la gestión de configuración, descubrimiento de servicios, balanceo de carga, resiliencia y seguridad. Al integrar Spring Cloud con Spring Boot, los desarrolladores pueden construir y gestionar sistemas complejos de manera más eficiente y efectiva.

Objetivos del Taller

Este Taller enseña a desarrollar aplicaciones Java escalables con Spring Boot y Spring Cloud, enfocándose en configuración mínima, automatización y despliegue eficiente. Los participantes aprenderán a crear servicios RESTful, microservicios y aplicaciones web, aprovechando herramientas de monitoreo y gestión. También abordarán desafíos de sistemas distribuidos, como configuración distribuida, descubrimiento de servicios y tolerancia a fallos. Con un enfoque práctico, el curso prepara a los desarrolladores para construir y gestionar aplicaciones empresariales complejas.

2

Contenidos

Introducción Spring Boot y APIs REST

- ¿Cómo funciona la Web? Arquitectura cliente-servidor
- Protocolo HTTP: Métodos GET, POST, DELETE y PUT
- Librerías vs Frameworks
- Introducción a Maven

Introducción a Spring Boot

- Introducción a Spring Framework
- Primer proyecto con Spring Boot
- Patrón MVC
- Introducción a las APIs Rest

APIs con Spring Boot

- Creando una API con Spring Boot
- @GetMapping + Parámetros: @PathVariable
- @GetMapping + Parámetros: @RequestParam
- Introducción a JSON
- @PostMapping + @RequestBody
- Postman
- Responses: @ResponseBody + ResponseEntity

APIs con Spring Boot – Parte II

- Patrón DTO (Data Transfer Object)
- Arquitectura Multicapas
- @Repository y @Service
- Inversión de Control
- Inyección de dependencias
- @Autowired

CRUD con JPA + Hibernate

- Introducción a ORM + JPA
- Motor JPA: Hibernate
- Configurando un proyecto con JPA + Hibernate
- Creando un CRUD con JPA + Hibernate
- Relaciones con JPA + Hibernate: @OneToOne + @OneToMany + @ManyToMany

Taller Spring Boot y Spring Cloud



Introducción a los Microservicios

Monolitos vs Microservicios
¿Cómo crear Microservicios? – Parte 1
Nuevas Annotations con Lombok
¿Cómo crear Microservicios? – Parte 2
Clases y Annotations a utilizar con Microservicios
¿Cómo comunicar Microservicios? RestTemplate
@ElementCollection y FetchType
Ejercicio Práctico

Spring Cloud y Feign

¿Qué vamos a ver en este módulo?
Introducción a Spring Cloud
¿Qué es FEIGN?
Ejercicios Prácticos

Patrones de diseño en Microservicios

Patrones: Service Registry y Service Discovery
Service Registry y Discovery con Eureka Server
Patrón: Load Balancing (Balanceo de Carga)
Load Balancing con SpringCloud Load Balancer
Patrón: Circuit Breaker
Aplicando Circuit Breaker con Resilience4J
Patrón: API Gateway
API Gateway con Spring Cloud Gateway
Patrón: Config Server (Central Configuration)
Aplicando Config Server

Docker y Microservicios

¿Qué es Docker?
¿Cómo Instalar Docker?
Docker y Microservicios

Inteligencia Artificial

- Acompañamiento con IA para la resolución específica de temas del curso

Conclusiones

Metodología

- Talleres prácticos (80% Práctico 20% Teórico).
- Los participantes deberán aprender los conceptos propios de la herramienta para ponerlos en práctica con diversos ejemplos que se consideran en el curso.

Taller Spring Boot y Spring Cloud

Dirigido a

Desarrolladores de aplicaciones

Requisitos Alumnos

Conocimientos en Java o en algún lenguaje de programación

Generalidades

- Se aplicará un cuestionario previo al inicio del taller a cada participante para evaluar su nivel de conocimiento. Esto permitirá identificar la necesidad de nivelación y/o realizar ajustes en el temario si es necesario.
- Duración 36 horas cronológicas
- Curso cerrado.