



Desarrollo de microservicios con Spring Cloud

ÁREA: DESARROLLO DE SOFTWARE

PRESENTACIÓN

La arquitectura de microservicios se ha convertido en un estándar para el desarrollo de aplicaciones modernas, al permitir construir sistemas escalables, resilientes y de fácil mantenimiento mediante servicios independientes y especializados

En combinación con las aplicaciones nativas de nube (Cloud Native), esta arquitectura facilita despliegues ágiles, alta disponibilidad y mejor aprovechamiento de los servicios en la nube.

Spring Cloud, basado en el ecosistema Spring, proporciona herramientas y soluciones para simplificar el desarrollo de sistemas distribuidos y microservicios, integrando tecnologías de empresas como VMware, HashiCorp y Netflix.

En este curso, los participantes aprenderán a diseñar e implementar microservicios utilizando Spring Cloud, aplicando buenas prácticas y tecnologías actuales orientadas a entornos empresariales y cloud.

PERFIL DE INGRESO

Para cursar este programa, se recomienda que la persona participante cuente con experiencia en el uso de algún framework de persistencia de datos en Java, así como conocimientos básicos de Spring y Spring Boot.

Asimismo, deberá contar con una preparación equivalente a la proporcionada en el curso Lenguaje de programación JAVA.

OBJETIVO

El participante diseñará e implementará APIs REST con Spring Boot, para su aplicación en arquitecturas de microservicios. Además, aplicará Spring Cloud para integrar principios de aplicaciones nativas de la nube y utilizar patrones de sistemas distribuidos en entornos reales.

TEMARIO

1. ARQUITECTURA DE APLICACIONES
 - 1.1. Arquitectura monolítica
 - 1.2. Arquitectura orientada a servicios (SOA)
 - 1.3. Arquitectura de microservicios (MSA)
 - 1.4. Aplicaciones nativas de nube (Cloud Native Applications)



2. DISEÑO DE APIS
 - 2.1. Diseño de interface a partir del código (Code-First)
 - 2.2. Diseño de interface a partir del requerimiento (Contract-First)
 - 2.3. OpenAPI
3. DESARROLLO DE APIS
 - 3.1. Diseño de contrato de API
 - 3.2. Descomposición funcional
 - 3.3. Spring Boot como base de una arquitectura de microservicios
 - 3.4. Implementación de APIs REST mediante Spring Boot
 - 3.5. Pruebas de APIs REST con POSTMAN y curl
4. INTRODUCCIÓN A CONTENEDORES
 - 4.1. Conceptos de contenedores
 - 4.2. Docker
 - 4.3. Comandos de Docker
 - 4.4. Dockerfile
 - 4.5. Creación de contenedores
 - 4.6. Creación de imagen
 - 4.7. DockerHub
5. DISEÑO DE APLICACIONES NATIVAS DE NUBE (12 FACTORES)
 - 5.1. Twelve-Factor App
 - 5.2. Codebase
 - 5.3. Dependencies
 - 5.4. Config
 - 5.5. Backing services
 - 5.6. Build, release, run
 - 5.7. Processes
 - 5.8. Port binding
 - 5.9. Concurrency
 - 5.10. Disposability
 - 5.11. Dev/prod parity
 - 5.12. Logs
 - 5.13. Admin processes
6. GESTIÓN DE LA CONFIGURACIÓN
 - 6.1. Configuración en Spring Boot y variables de entorno
 - 6.2. Configuración externalizada
 - 6.3. Spring Cloud Configuration Server
 - 6.4. Creación de Config Server



- 6.5. Git como repositorio de configuraciones
- 6.6. Externalización de secretos con variables de entorno
- 6.7. Integración de microservicios con ConfigClient
- 6.8. Spring Boot Actuator
- 6.9. Recarga de la configuración con @RefreshScope
- 7. DESCUBRIMIENTO DE SERVICIOS (SERVICE DISCOVERY)
 - 7.1. Escalamiento vertical y horizontal
 - 7.2. Service discovery en una aplicación nativa de nube
 - 7.3. Creación de Netflix Eureka Server
 - 7.4. Registro de microservicios en Eureka service registry
 - 7.5. Spring Cloud Discovery Client
 - 7.6. Spring Cloud Load Balancer
 - 7.7. Netflix Feign Client
- 8. PATRONES DE RESILIENCIA DEL LADO DEL CLIENTE
 - 8.1. Spring Cloud Load Balancer
 - 8.2. Resilience4j
 - 8.3. Circuit breaker
 - 8.4. Retry
 - 8.5. Client Load Balancing
 - 8.6. Fallback
 - 8.7. Bulkhead
- 9. RUTEO DE SERVICIOS
 - 9.1. ¿Qué es un API Gateway?
 - 9.2. Spring Cloud Gateway
 - 9.3. Creación de API Gateway
 - 9.4. Integración de API Gateway con Eureka Server
 - 9.5. Configuración de rutas
 - 9.6. Filtros globales y personalizados

PERFIL DE EGRESO

Los conocimientos adquiridos permitirán al participante construir una arquitectura de microservicios por medio de APIs REST implementando la resiliencia operativa necesaria para su ejecución exitosa por medio de patrones de sistemas distribuidos a través de Spring Cloud.

DURACIÓN

40 horas



REQUERIMIENTOS INFORMÁTICOS

- Cuenta de correo electrónico.
- Conexión a Internet.

HARDWARE

- Procesador x86 Intel o AMD de 2GHz de doble núcleo o superior
- Memoria RAM instalada de 16 GB RAM
- 20 GB libres en disco duro
- Pantalla con una resolución de 1024x768 o superior.
- Cámara
- Micrófono
- Bocinas

SOFTWARE

- Sistema operativo de 64 bits
 - Windows 10, 11 o 12
 - Linux
 - MacOS
- OpenJDK JDK 21; <https://jdk.java.net/archive/>
- Apache Maven 3.9; <https://maven.apache.org/download.cgi>
- Eclipse IDE for Enterprise Java and Web Developers;
<https://www.eclipse.org/downloads/packages/release/2025-12/r/eclipse-ide-enterprise-java-and-web-developers>
- Plugin Eclipse Spring Tools; <https://marketplace.eclipse.org/content/spring-tools-aka-spring-tool-suite>; Seleccionar y arrastrar el ícono Install en Eclipse para realizar la instalación
- Postman; <https://www.postman.com/downloads/>
- Docker
 - Windows o MacOS; <https://docs.docker.com/desktop/>
 - Linux; <https://docs.docker.com/engine/install>



BIBLIOGRAFÍA

- Binildas, C. A. (Christudas, A.). (2024). *Java microservices and containers in the cloud: With Spring Boot, Kafka, PostgreSQL, Kubernetes, Helm, Terraform and AWS EKS*. Apress.
- Gutierrez, F. (2021). *Spring Cloud Data Flow: Native cloud orchestration services for microservice applications on modern runtimes*. Apress.
- Macero García, M. (2020). *Learn microservices with Spring Boot: A practical approach to RESTful services using an event-driven architecture, cloud-native patterns, and containerization*. Apress.
- Mishra, A. (2022). *Microsoft Azure for Java developers: Deploying Java applications through Azure WebApp, Azure Kubernetes Service, Azure Functions, and Azure Spring Cloud*. Apress Media.
- Newman, S. (2021, agosto). *Building microservices: Designing fine-grained systems* (2nd ed.). O'Reilly Media.
- Nickoloff, J. (2019, noviembre). *Docker in action* (2nd ed.). Manning Publications.
- Scholl, B. (2019, agosto). *Cloud native*. O'Reilly Media.
- Shivakumar, G. (2021, septiembre). *Cloud native architecture and design: A handbook for modern day architecture and design with enterprise-grade examples*. Apress.
- Varanasi, B. (2022). *Spring REST: Building Java microservices and cloud applications*. Apress.