

CURSO de ESPECIALIZACIÓN en: **TECNOLOGÍAS JAVA**

MÓDULO	JAVA MODERNO	2 SEMANAS
	SPRING BOOT	3 SEMANAS
	SPRING BOOT Avanzado	2 SEMANAS
	APACHE KAFKA	2 SEMANAS
	QUARKUS	3 SEMANAS
Requisitos de conocimientos previos	Conocimientos básicos de programación y fundamentos de programación orientada a objetos. Se recomienda contar con nociones de bases de datos relacionales, SQL.	
Duración	3 MESES Lunes a Viernes	

MÓDULO 1. JAVA MODERNO

Fundamentos y Desarrollo Profesional

1. Fundamentos y Java moderno

- 1.1. Entorno de desarrollo y JDK.
- 1.2. Compilación, ejecución y estructura de proyectos.
- 1.3. Variables, tipos de datos y operadores.
- 1.4. Estructuras condicionales e iterativas.
- 1.5. Métodos.
- 1.6. Clases y objetos.
- 1.7. Encapsulación.
- 1.8. Constructores.
- 1.9. Modificadores de acceso.
- 1.10. Herencia.
- 1.11. Polimorfismo.
- 1.12. Interfaces y clases abstractas.
- 1.13. Static, final y buenas prácticas.

2. Java para desarrollo profesional

- 2.1. Excepciones y manejo de errores.
- 2.2. Colecciones.
- 2.3. Genéricos.
- 2.4. Lambdas y expresiones funcionales.
- 2.5. Streams API.
- 2.6. Optional.
- 2.7. Manejo de archivos.

- 2.8. API moderna de fechas.
- 2.9. Maven o Gradle.
- 2.10. Git y control de versiones.
- 2.11. Principios básicos de código limpio y refactorización.

Producto del módulo: aplicación Java estructurada, versionada y con buenas prácticas básicas.

MÓDULO 2. SPRING BOOT BACKEND, APIs REST Y PERSISTENCIA

1. Spring Boot y APIs REST

- 1.1. Spring Initializr.
- 1.2. Estructura de un proyecto.
- 1.3. IoC e inyección de dependencias.
- 1.4. Controllers.
- 1.5. APIs REST.
- 1.6. DTO.
- 1.7. ResponseEntity.
- 1.8. Parámetros y rutas.
- 1.9. Validación.
- 1.10. Configuración mediante application.properties y YAML.
- 1.11. Documentación de APIs. (OpenAPI 3 / Swagger)

2. Persistencia con JPA y Hibernate

- 2.1. ORM y JPA.
- 2.2. Hibernate.
- 2.3. Spring Data JPA.
- 2.4. Entidades.
- 2.5. Relaciones.
- 2.6. Repositorios.
- 2.7. Consultas personalizadas.
- 2.8. Paginación.
- 2.9. Transacciones.
- 2.10. PostgreSQL.
- 2.11. Migraciones de base de datos.

3. Calidad, pruebas y seguridad

- 3.1. Manejo global de excepciones.
- 3.2. Logging.
- 3.3. JUnit.
- 3.4. Mockito.
- 3.5. Pruebas de servicios.
- 3.6. Pruebas de controladores.
- 3.7. Pruebas de integración.
- 3.8. Introducción a Testcontainers.
- 3.9. Spring Security.
- 3.10. JWT.

- 3.11. Roles y permisos.
- 3.12. CORS.

Producto del módulo: API REST completa con persistencia, validación, pruebas y seguridad.

MÓDULO 3. SPRING BOOT AVANZADO: IA y Microservicios

1. IA aplicada al desarrollo con Java

- 1.1. Desarrollo asistido por IA
 - 1.1.1. Autocompletado inteligente.
 - 1.1.2. Generación de código.
 - 1.1.3. Explicación de código.
 - 1.1.4. Generación de DTO, servicios y estructuras repetitivas.
 - 1.1.5. Generación de pruebas.
 - 1.1.6. Identificación de casos límite.
 - 1.1.7. Refactorización asistida.
 - 1.1.8. Revisión de código.
 - 1.1.9. Generación de documentación.
- 1.2. Buenas prácticas
 - 1.2.1. Ingeniería de contexto.
 - 1.2.2. Proporcionar arquitectura y restricciones.
 - 1.2.3. Convenciones de código.
 - 1.2.4. Solicitud de soluciones verificables.
 - 1.2.5. Comparación entre alternativas.
 - 1.2.6. Verificación humana.
 - 1.2.7. Ejecución de pruebas.
 - 1.2.8. Detección de errores o alucinaciones.
 - 1.2.9. Seguridad y privacidad.
 - 1.2.10. Manejo de secretos y credenciales.
 - 1.2.11. Consideraciones sobre propiedad intelectual y dependencias.

2. Arquitectura avanzada y microservicios

- 2.1. Monolito modular vs microservicios.
- 2.2. Cuándo utilizar microservicios.
- 2.3. Comunicación síncrona.
- 2.4. REST.
- 2.5. OpenFeign o WebClient.
- 2.6. API Gateway.
- 2.7. Configuración distribuida.
- 2.8. Service Discovery cuando corresponda.
- 2.9. Resilience4j.
- 2.10. Circuit Breaker.
- 2.11. Retry.
- 2.12. Rate limiting.
- 2.13. Observabilidad básica.

Producto del módulo: Arquitectura de microservicios con Spring Boot, resiliencia y asistencia de IA aplicada al desarrollo.

MÓDULO 4. APACHE KAFKA:

Arquitectura Orientada a Eventos

1. Fundamentos de Apache Kafka

- 1.1. Arquitectura orientada a eventos.
- 1.2. Eventos y mensajería.
- 1.3. Kafka moderno.
- 1.4. Brokers.
- 1.5. Topics.
- 1.6. Particiones.
- 1.7. Replicación.
- 1.8. Producers.
- 1.9. Consumers.
- 1.10. Consumer Groups.
- 1.11. Offsets.
- 1.12. Serialización.
- 1.13. Configuración básica.
- 1.14. Administración.

2. Kafka aplicado con Java

- 2.1. Productores Java.
- 2.2. Consumidores Java.
- 2.3. Integración con Spring Boot.
- 2.4. Comunicación asíncrona.
- 2.5. Manejo de errores.
- 2.6. Reintentos.
- 2.7. Dead Letter Topics.
- 2.8. Garantías de entrega.
- 2.9. Idempotencia.
- 2.10. Diseño básico de eventos.
- 2.11. Monitoreo y pruebas.

Producto del módulo: integración basada en eventos entre componentes del sistema.

MÓDULO 5. QUARKUS :

Desarrollo Cloud-Native

1. Fundamentos de Quarkus

- 1.1. ¿Qué es Quarkus?
- 1.2. Diferencias conceptuales con Spring Boot.
- 1.3. Arquitectura basada en extensiones.
- 1.4. Quarkus CLI.
- 1.5. Creación de proyectos.

- 1.6. Dev Mode.
- 1.7. Live Coding.
- 1.8. REST con Jakarta REST.
- 1.9. JSON.
- 1.10. Configuración.
- 1.11. Perfiles.
- 1.12. Dev Services.
- 2. Persistencia, pruebas y mensajería con Quarkus**
 - 2.1. Hibernate ORM.
 - 2.2. Panache.
 - 2.3. Repository Pattern.
 - 2.4. Active Record Pattern.
 - 2.5. PostgreSQL.
 - 2.6. Transacciones.
 - 2.7. REST + persistencia.
 - 2.8. Testing con Quarkus.
 - 2.9. @QuarkusTest.
 - 2.10. Mocking.
 - 2.11. Integración con Kafka.
 - 2.12. Introducción al enfoque reactivo.
- 3. Quarkus Cloud-Native y proyecto integrador**
 - 3.1. Docker y contenedores.
 - 3.2. Construcción de imágenes.
 - 3.3. Native Image.
 - 3.4. GraalVM o Mandrel.
 - 3.5. Health Checks.
 - 3.6. Configuración externa.
 - 3.7. Observabilidad básica.
 - 3.8. Kubernetes.
 - 3.9. Comparación final Spring Boot vs Quarkus.
 - 3.10. Integración con el proyecto desarrollado.

Proyecto final:

Desarrollar inicialmente un backend con Spring Boot y, posteriormente, implementa un servicio complementario con Quarkus.