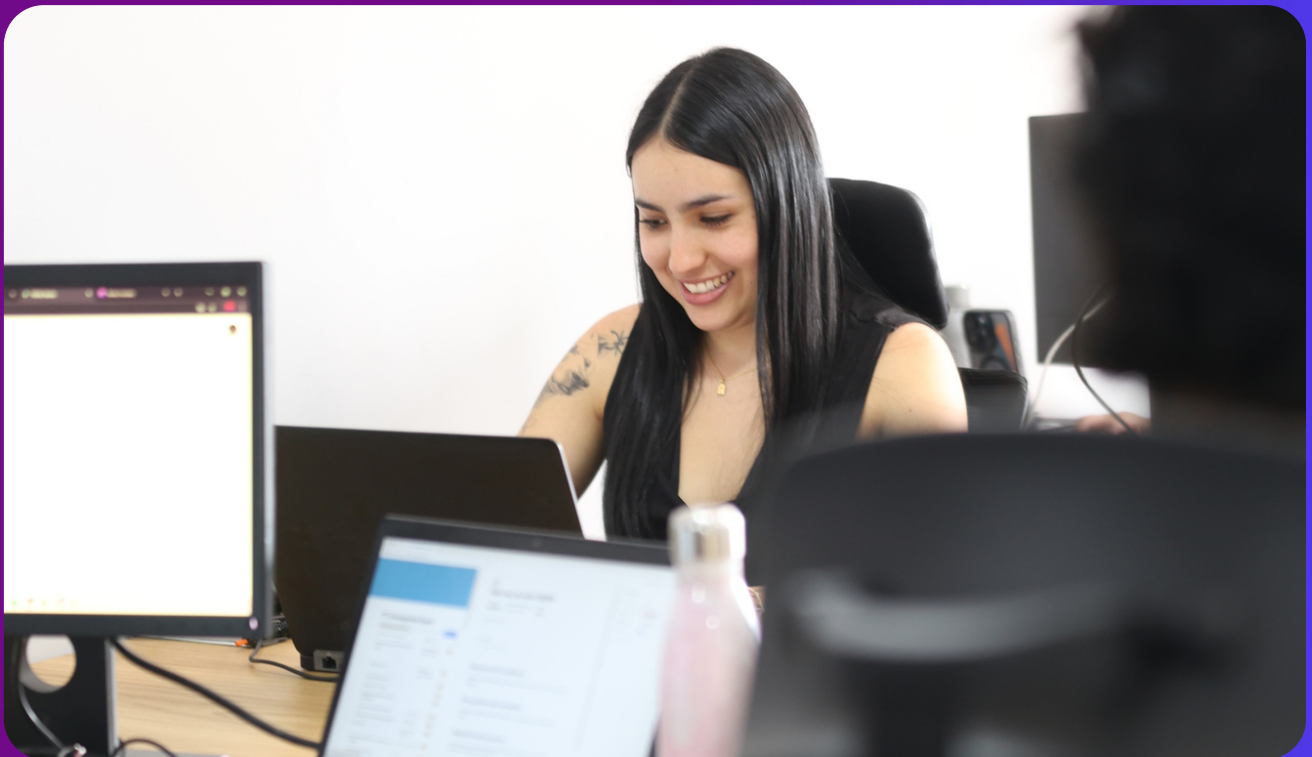


Formación en Java Backend Developer

PROGRAMA 2026



Este programa tiene como objetivo formar desarrolladores backend expertos, con habilidades avanzadas en Java, arquitectura de software, bases de datos SQL y NoSQL, pruebas automatizadas, programación funcional, despliegue en la nube con AWS y capacidad para aplicar inteligencia artificial en el ciclo de desarrollo y en soluciones backend reales, integrando buenas prácticas de seguridad, calidad, documentación y automatización.

1

Duración

160h con una intensidad horaria de 9 horas semanales durante 4.5 meses.

2

Formato

100 % virtual y en vivo

3

Metodología

Aprendizaje teórico acompañado de un proyecto práctico transversal y progresivo, integrando IA como apoyo al análisis, diseño, construcción, pruebas, documentación y despliegue de soluciones backend.

4

Admisión

Mínimo de 290 / 480 puntos en la prueba de admisión.

5

Enfoque

Integración transversal de IA generativa, asistentes de programación, automatización de pruebas, documentación técnica, consumo de APIs de IA y despliegue seguro de soluciones backend con capacidades inteligentes.

Este programa está dirigido a desarrolladores que ya hayan trabajado con Java en entornos de desarrollo reales o académicos y desean profundizar en arquitectura de software backend moderna, incorporando inteligencia artificial como herramienta de productividad y como capacidad técnica integrada en aplicaciones empresariales.

CONOCIMIENTOS OBLIGATORIOS

- Al menos 1 año de experiencia práctica en desarrollo de software con Java.
- Dominio de la Programación Orientada a Objetos (POO): uso de clases, herencia, polimorfismo, encapsulamiento y diseño basado en interfaces.
- Conocimiento intermedio en bases de datos relacionales: diseño de esquemas, modelos entidad-relación, uso de joins, subconsultas y funciones agregadas.
- Familiaridad con el framework Spring Boot: creación de APIs RESTful con arquitectura MVC, uso de anotaciones y configuración básica.
- Uso práctico de herramientas como Git para control de versiones y comprensión de integración continua básica.
- Conocimientos fundamentales de HTTP, métodos, códigos de respuesta y consumo de APIs desde Postman o Swagger.
- Capacidad para seguir instrucciones técnicas, leer documentación oficial, validar respuestas generadas por herramientas de IA y colaborar en equipos de desarrollo.
- Interés en usar IA generativa de manera responsable para acelerar análisis, diseño, codificación, pruebas, documentación y resolución de problemas técnicos.

PERFIL DE SALIDA

Al finalizar el programa, los participantes habrán desarrollado un perfil de desarrollador backend con competencias avanzadas y capacidad para incorporar inteligencia artificial de forma práctica, segura y alineada con necesidades de negocio, incluyendo:

- ↘ Diseño e implementación de APIs RESTful robustas, aplicando arquitecturas modernas como MVC, Hexagonal y principios SOLID.
- ↘ Uso avanzado de Java y Spring Boot para construir microservicios desacoplados, seguros, escalables y preparados para integrarse con servicios de IA.
- ↘ Uso productivo de asistentes de IA para acelerar tareas de análisis, generación de código base, refactorización, documentación, revisión de errores y creación de pruebas, manteniendo criterio técnico y validación humana.
- ↘ Integración de capacidades de IA en aplicaciones backend mediante consumo de APIs de modelos de lenguaje, prompts estructurados, manejo de respuestas, validaciones, errores, límites de uso y trazabilidad.
- ↘ Manejo profesional de bases de datos relacionales y no relacionales (SQL, DynamoDB, DocumentDB), incluyendo integración, consultas eficientes, modelos híbridos y preparación de datos para casos de analítica e IA.
- ↘ Programación funcional con Java, uso de streams, lambdas y patrones de flujo de datos asíncronos con WebFlux.

PERFIL DE SALIDA

- ↘ Implementación de pruebas automatizadas con JUnit, Mockito y Karate, logrando una cobertura mínima del 80% con JaCoCo, incorporando IA como apoyo para generar escenarios, datos de prueba y casos borde, y ejecutando pruebas de performance con JMeter.
- ↘ Construcción de casos básicos de IA aplicada al backend, como clasificación, recomendación, extracción de información, búsqueda semántica o predicción simple sobre datos del dominio del negocio.
- ↘ Despliegue de soluciones sobre infraestructura en la nube (AWS) con EC2, RDS, S3, IAM, Cognito y Secrets Manager; orquestación local con Docker Compose y comprensión teórica de Kubernetes, incluyendo pods, deployments, services e ingress.
- ↘ Automatización del ciclo de vida del software mediante integración continua y despliegue continuo (CI/CD) con Jenkins, Docker y Docker Compose, incorporando observabilidad con OpenTelemetry y Grafana, complementada con servicios nativos como CloudWatch.
- ↘ Dominio de herramientas de calidad, seguridad y documentación: SonarLint, OpenAPI, Spring Security, gestión de secretos, manejo seguro de tokens y control de exposición de datos sensibles.

STACK TECNOLÓGICO

CATEGORÍA	HERRAMIENTAS / ALCANCE
Lenguaje de Programación	Java
Frameworks	Spring Boot, Spring Cloud, WebFlux, Spring AI
Bases de Datos	SQL, DynamoDB, DocumentDB, Redis y fundamentos de almacenamiento orientado a búsqueda semántica
Pruebas	JUnit, Mockito, JaCoCo, Karate, JMeter y generación asistida de escenarios con IA
DevOps	Git, Jenkins, Docker, Docker Compose, CI/CD y SonarLint
Cloud	AWS (EC2, RDS, Cognito, CloudWatch, Secrets Manager, KMS, IAM, S3 y fundamentos de servicios de IA en la nube) y fundamentos teóricos de Kubernetes
Documentación	OpenAPI, documentación técnica asistida por IA y validación de contratos de API
Herramientas de IA	GitHub Copilot o asistentes equivalentes, ChatGPT, APIs de modelos de lenguaje, embeddings, prompts estructurados y fundamentos de RAG
Otras herramientas	Lombok, MapStruct, Log4j, Postman, Swagger, OpenTelemetry y Grafana

HABILIDADES ADQUIRIDAS AL FINALIZAR EL PROGRAMA

- ↘ Diseñar y construir APIs RESTful profesionales usando Java y Spring Boot con arquitecturas modernas como MVC y Hexagonal.
- ↘ Aplicar principios de Clean Code y SOLID, desarrollando software limpio, mantenible y escalable, apoyándose en IA para análisis de código, refactorización y detección temprana de problemas.
- ↘ Utilizar programación funcional y reactiva en Java mediante el uso de lambdas, streams y Spring WebFlux.
- ↘ Integrar bases de datos SQL y NoSQL, incluyendo PostgreSQL, DynamoDB y DocumentDB, garantizando eficiencia, atomicidad, consistencia y preparación de datos para consumo analítico o inteligente.
- ↘ Automatizar pruebas de software con JUnit, Mockito y Karate, lograr una cobertura mínima del 80% usando JaCoCo, diseñar escenarios con apoyo de IA y ejecutar pruebas de carga y performance con JMeter.
- ↘ Aplicar modelos simples de Inteligencia Artificial en entornos backend, como predicción de demanda, clasificación de productos, extracción de información, recomendación o búsqueda semántica.
- ↘ Consumir servicios de IA desde aplicaciones Java, gestionando prompts, respuestas estructuradas, errores, límites, costos, seguridad y trazabilidad de las interacciones.

HABILIDADES ADQUIRIDAS AL FINALIZAR EL PROGRAMA

- Usar herramientas de IA generativa para mejorar productividad en análisis, desarrollo, documentación, debugging, revisión de código y diseño de APIs.
- Contenerizar aplicaciones con Docker y orquestar entornos locales o de integración con Docker Compose, garantizando configuraciones reproducibles y consistentes.
- Implementar CI/CD con Jenkins, creando pipelines para integración continua, pruebas, análisis de calidad, seguridad y despliegue automatizado, e instrumentar aplicaciones con OpenTelemetry para visualizar métricas y trazas en Grafana, complementando el monitoreo con CloudWatch.
- Desplegar soluciones backend en AWS utilizando EC2, RDS, S3, IAM, Cognito y Secrets Manager, aplicando criterios de seguridad, escalabilidad y gestión de costos; además, comprender conceptualmente Kubernetes y sus componentes principales sin requerir la creación de clústeres EKS.
- Garantizar la seguridad de las aplicaciones con Spring Security, incluyendo manejo de autenticación, autorización, JWT, OAuth2 básico y protección de información sensible al integrar IA.
- Documentar APIs con OpenAPI y documentación asistida por IA para facilitar la colaboración entre equipos y proveedores externos.
- Colaborar en proyectos de software utilizando Git, prácticas de versionamiento, revisión de código y trabajo en equipo técnico.

MÓDULOS DEL PROGRAMA

MÓDULO 1

Programación Backend básica con fundamentos de IA



OBJETIVO

Construir APIs REST con arquitectura MVC que persistan información en bases de datos relacionales (SQL) y no relacionales (NoSQL), aplicando buenas prácticas de desarrollo, principios SOLID, pruebas unitarias con JUnit y Mockito, y uso de herramientas como SonarLint. Se introduce el uso de Clean Code, asistentes de IA para productividad del desarrollador y fundamentos de IA aplicada en entornos backend a través de casos simples en Java.

MÓDULO 2

Programación Backend Intermedia con integración de IA



OBJETIVO

Diseñar e implementar APIs REST con arquitectura Hexagonal, integrando principios de programación funcional en Java (lambdas, streams, funciones puras) y programación reactiva usando WebFlux. Se aplican prácticas de TDD y se garantiza una cobertura mínima del 80% con herramientas como JaCoCo. Se trabaja con bases de datos SQL y NoSQL (DynamoDB, DocumentDB), Docker, Docker Compose, Jenkins, Redis y consumo de servicios de IA mediante APIs, prompts estructurados, validación de respuestas y casos backend inteligentes.

MÓDULO 3

Programación Backend Complementaria, Cloud, DevOps, observabilidad e IA en producción



OBJETIVO

Implementar soluciones completas sobre servicios de AWS como EC2, RDS, S3, Cognito, IAM, Secrets Manager y CloudWatch. Los estudiantes despliegan aplicaciones contenerizadas mediante Docker y Docker Compose, construyen pipelines CI/CD con Jenkins, instrumentan servicios con OpenTelemetry, visualizan métricas y trazas en Grafana y ejecutan pruebas de performance con JMeter. Kubernetes se aborda a nivel conceptual —pods, deployments, services e ingress— sin requerir la creación de clústeres EKS, priorizando una práctica accesible, segura y orientada al control de costos.

MÓDULO	UNIDAD	NOMBRE	HORAS
I	1	Introducción Proyecto y enfoque IA	3
I	2	SQL - NoSQL para Backend e IA	12
I	3	Spring Boot & Clean con IA asistida	8
I	4	Pruebas Unitarias asistidas por IA	8
I	5	IA aplicada a Java y GenAI Backend	9
I	6	Clean Code - Principios SOLID	4
I	6	Clean Code - Principios SOLID	4
II	7	Arquitectura Hexagonal	14
II	8	Programación Funcional	9
II	10	Docker y Docker Compose	8
II	12	Spring Security	6
III	14	AWS Práctico e integración de servicios IA	20
III	15	DevOps, observabilidad y performance	24

UNIDAD 1

Introducción al Proyecto y enfoque IA



Presentación del reto ARKA, comprensión del caso de negocio, revisión del backlog, arquitectura base y organización del trabajo durante el programa. Se define cómo se utilizará la IA de forma transversal para análisis de requerimientos, generación de hipótesis técnicas, apoyo en documentación, validación de decisiones y aceleración del desarrollo sin reemplazar el criterio del equipo.

UNIDAD 2

SQL y NoSQL para Backend e IA

Modelado de bases de datos relacionales y no relacionales. Consultas SQL intermedias, relaciones entre tablas, principios de normalización e introducción a DynamoDB y DocumentDB. Se incorporan criterios de preparación de datos para casos de IA, calidad de datos, trazabilidad, consultas eficientes y diseño de estructuras que puedan alimentar servicios inteligentes o analítica backend.

UNIDAD 3

Spring Boot & Clean con IA asistida

Creación de proyectos base con Spring Boot. Implementación de arquitectura MVC, controladores, servicios, repositorios y uso de anotaciones clave. Integración con bases de datos. Se utiliza IA generativa como apoyo para scaffolding, explicación de errores, generación de ejemplos, documentación de endpoints y revisión inicial de estructura, manteniendo validación técnica por parte del desarrollador.

UNIDAD 4

Pruebas Unitarias asistidas por IA

Fundamentos de pruebas en Java con JUnit y Mockito. Diseño de pruebas, mocks, asserts y configuración de entornos de prueba. Se incorporan asistentes de IA para identificar casos borde, proponer datos de prueba, generar escenarios iniciales y mejorar la cobertura, asegurando que los resultados sean revisados, ajustados y validados por el equipo técnico.

UNIDAD 5

IA Aplicada a Java y GenAI Backend

Fundamentos de IA aplicada en backend. Introducción a bibliotecas y servicios de IA disponibles para Java, consumo de APIs de modelos de lenguaje, prompts estructurados, respuestas en formato JSON, embeddings, clasificación, predicción o detección de patrones simples. Se desarrollan casos de uso con datos del proyecto, incluyendo automatización de respuestas, extracción de información, recomendación básica o búsqueda semántica.

UNIDAD 6

Clean Code – Principios SOLID

Aplicación de principios SOLID en código Java. Refactorización, buenas prácticas, legibilidad y uso de herramientas como SonarLint para asegurar calidad de código. Se integra IA como apoyo para detectar duplicidad, sugerir mejoras, explicar smells y proponer alternativas de diseño, manteniendo criterios de arquitectura, mantenibilidad y seguridad.

UNIDAD 7

Arquitectura Hexagonal

Diseño de aplicaciones desacopladas con Hexagonal. Implementación de adaptadores de entrada y salida, puertos y separación de lógica de negocio. Se incorporan escenarios donde la IA se consume como un adaptador externo, evitando acoplamientos directos con proveedores, facilitando pruebas, reemplazo de servicios y control de riesgos técnicos.

UNIDAD 8

Programación Funcional

Introducción a la programación funcional con Java: lambdas, streams, funciones puras, inmutabilidad, operaciones terminales e intermedias. Se aplican estos conceptos al procesamiento de datos, transformación de colecciones, preparación de insumos para servicios de IA y construcción de flujos más expresivos, legibles y mantenibles.

UNIDAD 9

Programación Reactiva (WebFlux)

Desarrollo de APIs reactivas con Spring WebFlux. Uso de Mono y Flux, backpressure, patrones de suscripción y asincronismo en flujos de datos. Se exploran integraciones con servicios externos, incluyendo servicios de IA, manejo de latencia, timeouts, resiliencia y procesamiento asíncronico de respuestas.

UNIDAD 10

Docker y Docker Compose

Contenerización de aplicaciones Java, creación y optimización de imágenes Docker y construcción de entornos reproducibles con Docker Compose. Se orquestan localmente servicios backend, bases de datos, caché y dependencias del proyecto, gestionando redes, volúmenes, variables de entorno, health checks, credenciales y configuración segura para desarrollo, pruebas e integración continua.

UNIDAD 11

Spring Cloud

Fundamentos de microservicios con Spring Cloud: configuración distribuida, descubrimiento de servicios, balanceo de carga y comunicación entre servicios. Se revisan patrones para exponer capacidades de IA como servicios internos, centralizar configuración, manejar resiliencia y desacoplar integraciones con proveedores externos.

UNIDAD 12

Spring Security

Seguridad en aplicaciones Java: autenticación y autorización con Spring Security, JWT, OAuth2 básico, manejo de usuarios y roles. Se incorporan buenas prácticas para proteger endpoints que consumen IA, controlar permisos, evitar exposición de datos sensibles, proteger claves de API y validar entradas/salidas generadas por modelos.

UNIDAD 10

**Cloud Computing – AWS y
fundamentos de Kubernetes**

Conceptos de nube, servicios clave de AWS (EC2, RDS, IAM, Cognito y S3), arquitectura orientada a servicios, despliegue inicial de aplicaciones, seguridad, escalabilidad, monitoreo y gestión de costos. Se introducen de forma teórica los fundamentos de Kubernetes —pods, deployments, services, ingress, namespaces y configuración— y el propósito de servicios administrados como EKS, sin aprovisionar clústeres durante la formación.

UNIDAD 14

AWS Práctico e integración de servicios IA

Implementación práctica de infraestructura accesible en AWS: despliegue de aplicaciones contenerizadas en EC2, bases de datos en RDS, almacenamiento en S3, gestión de identidades con IAM y Cognito, manejo de secretos con Secrets Manager y monitoreo básico con CloudWatch. Se integran servicios backend con capacidades de IA, manejo seguro de tokens, variables y secretos, trazabilidad de consumo y criterios de operación y control de costos, sin requerir clústeres EKS.

UNIDAD 15

DevOps, observabilidad y performance

Automatización del ciclo de vida del software con Jenkins: integración y entrega continua, creación de pipelines, testing automatizado, análisis de calidad, versionamiento y despliegue. Se instrumentan aplicaciones con OpenTelemetry para recolectar métricas y trazas, se construyen tableros en Grafana y se complementa la operación con CloudWatch. Además, se diseñan y ejecutan pruebas de carga, concurrencia y performance con JMeter, interpretando resultados para identificar cuellos de botella y validar el comportamiento de las APIs.

enyo*i* by AceleraTI



TELÉFONO
(57) 313 710 7072



PÁGINA WEB
acelerati_latam



INSTAGRAM
acelerati_latam



LINKEDIN
AceleraTI_Latam