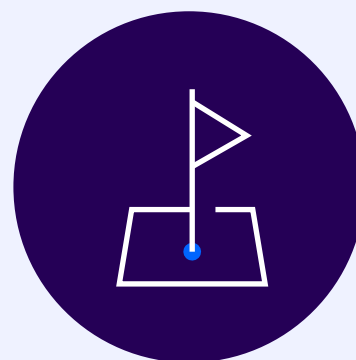


Microservices Architecture: Java, Spring Cloud & GCP-GKE

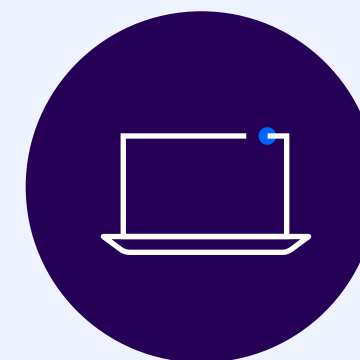
Patterns, Design, Programming, Testing, Build and Cloud Deployment



Inicio:
29 DE SETIEMBRE



Finalización:
26 DE NOVIEMBRE



52 HORAS
académicas



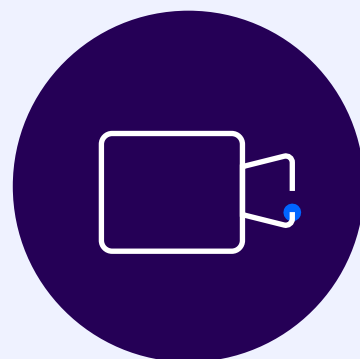
Miércoles y Viernes
DE 19:30H - 22:00H.



Soporte
TÉCNICO



Plataforma
DIGITAL



Aplicativo
ZOOM



**CERTIFICACIÓN
DIGITAL**
Previa aprobación
de examen

Instructor

ARISTEDES NOVOA

Ingeniero de Sistemas Colegiado de la Universidad Nacional del Callao (UNAC), con maestría en Computación y Sistemas de la Universidad de San Martín de Porres (USMP) y Doctorado en Ingeniería Industrial de la Universidad Nacional Mayor de San Marcos (UNMSM). Especialista en optimización, sistematización y automatización de procesos e implementación de soluciones empresariales utilizando Java; con más de 15 años de experiencia.

Área de Especialización: JAVA

Principales Tecnologías

- Java Core
- Java SE
- Java EE
- Microservicios
- Spring Framework
- Docker y Kubernetes



Certificaciones Internacionales

- Java SE (Oracle)
- Agile (SDC, SMC, SPOC, SSMC, SSPOC, STC)
- DevOps y CI/DI (DEPC, Jenkins)
- Big Data y Data Science (Hadoop, Spark, IA, ML, DL)



Líneas de Investigación

- IA, ML, DL
- Blockchain
- Sistemas Reactivos



anovoa@galaxy.edu.pe



Linkedin: <https://bit.ly/3cMcfv3>

RESUMEN

Conceptualización, diseño, programación, pruebas y despliegue de microservicios utilizando Java con Spring Boot y Spring Cloud. Implementación de principales patrones a través de un caso práctico evolutivo utilizando tecnologías exclusivas. Además de conocer las principales recomendaciones y buenas prácticas para implementar arquitecturas de microservicios a nivel empresarial.

OBJETIVO

Contar con los conocimientos y entrenamiento práctico en el uso de Java para desarrollar e implementar microservicios a nivel empresarial utilizando Java, Spring Boot y Spring Cloud; estar preparado para llevar cursos de nivel avanzado en Arquitectura de Microservicios implementando patrones avanzados y tecnologías Cloud (AWS, GCP, Azure, Oracle Cloud, IBM y Red Hat).

METODOLOGÍA

- Exposición de aspectos teóricos de los temas.
- Desarrollo de casos prácticos progresivos (instructor y alumno).
- Experiencias del instructor y alumnos del curso.
- Desarrollo de proyecto de curso.

REQUISITOS ACADÉMICOS

- Fundamentos de arquitectura de aplicaciones en capas o similares.
- Conocimientos avanzados de programación orientada a objetos y funcional de preferencia.
- Conocimiento de bases de datos relacionales y no relacionales de preferencia.
- Conocimientos avanzados de Servicios RESTful con Spring Boot.

MICROSERVICES PATTERNS

- Decompose by business capability
- Database per Service and Shared database
- Externalized configuration
- Service Registry
- Querys(CQRS and API Composition)
- Circuit Breaker (Reliability)
- API Gateway
- Data management (SAGA)
- Service Component Test
- Distributed Tracing and Monitoring
- Observability(Application Metrics and Health check)
- Access Token
- Service instance per Container
- UI patterns (Client-side UI composition)

STACK TECNOLÓGICO

- Java 11 y 16
- Eclipse, Spring Tool Suite, IntelliJ IDEA y Visual Studio Code
- Gradle y Git (Maven opcionalmente)
- Spring ecosystem(Spring Boot, Spring Data, Spring Cloud, Spring Security, Resilience4j, Lombok, JWT, OAuth2).
- Oracle, MySQL, PostgreSQL, Redis, MongoDB y Apache Kafka
- Grafana, Prometheus, ELK/EFA, Jaeger
- Junit y Mockito
- JSONLint, Postman y Swagger
- Node.js, Angular 12, TypeScript y Bootstrap
- Docker y Kubernetes
- Cloud: Google Kubernetes Engine –GKE)

CASO PRÁCTICO DESARROLLADO POR EL INSTRUCTOR

Arquitectura de microservicios de una plataforma de **ventas online** que permita gestionar los principales componentes y procesos de negocio, considerando: productos, clientes, pedidos (maestro-detalle), pagos, despacho, inventarios y estadísticas.

CASO PRÁCTICO DESARROLLADO POR EL ALUMNO

Arquitectura de microservicios de una plataforma de gestión de servicios de **autorizaciones** que permita gestionar los principales componentes y procesos de negocio, considerando: usuarios(clientes), autorizaciones, requisitos, pagos, seguimiento, evaluaciones, observaciones, subsanaciones, encuestas y estadísticas

NOTA: El alumno puede proponer y desarrollar un caso diferente en coordinación con el instructor.

SEMANA 01

INTRODUCCIÓN A MICROSERVICIOS

- ➔ Introducción a arquitectura de aplicaciones
- ➔ Microservicios vs aplicaciones monolíticas (pros y contras)
- ➔ Principales patrones y su implementación en el curso
- ➔ Conceptualización y modelamiento del caso a desarrollar en el curso.
- ➔ Proyectos de referencia y recomendaciones para el desarrollo del proyecto del curso



CREACIÓN DE ARQUITECTURA BASE

- ➔ Mi primer microservicio (gradle e yml).
- ➔ Creación de la arquitectura base (Entity, DTO, Repository, RestController, interfaces y excepciones genéricas).
- ➔ Desarrollo de un **microservicio de referencia**.
- ➔ Pruebas del microservicio con Postman y creación de colecciones (importación y exportación).
- ➔ Creación de estructuras de colección de los proyectos de microservicios de negocio, configuración, administración, seguridad, monitoreo y otros.

SEMANA 02

SPRING CLOUD CONFIG

- ➔ Introducción a cross-cutting concerns patterns.
- ➔ Introducción a Config Server y Config Client.
- ➔ Creando un microservicio de configuración(**ConfigService**) y datos de configuración(**config-data**).
- ➔ Implementando configuración nativa(local) y en nube(git) - asegurando con Spring Security.
- ➔ Configuración de los microservicios de referencia.



SPRING CLOUD DISCOVERY

- ➔ Introducción a service discovery patterns.
- ➔ Introducción a Eureka Server y Eureka Client.
- ➔ Creando un microservicio de registro(**RegistryService**).
- ➔ Configurando los microservicios de referencia.
- ➔ Métricas y monitoreo (Eureka Dashboard y Spring Boot Admin).

SEMANA 03

DATABASE PER SERVICE, API COMPOSITE Y API GATEWAY

- ➔ Implementación de Database per Service patterns(2 RDBMS).
- ➔ Implementación de API Composite (REST Template y Service Discovery).
- ➔ Introducción a external API patterns.
- ➔ Implementación de API Gateway patterns - Spring Cloud Gateway(**GatewayService**).
- ➔ Configuración de los microservicios de referencia.



RESILENCE

- ➔ Introducción a circuit breaker patterns.
- ➔ Introducción a librerías y frameworks disponibles (Hystrix, Resilience4j y Spring Retry).
- ➔ Ciclo de resiliencia(open, close y half-open).
- ➔ Implementación resiliencia de microservicios con Resilience4j.
- ➔ Realización de pruebas de performance y disponibilidad con Curl.

SEMANA 04

APACHE KAFKA

- ➔ Introducción a Event Driven Architecture (EDA's).
- ➔ Arquitectura general de Apache Kafka(Topics, Partitions, Producer y Consumer).
- ➔ Introducción publish / subscribe pattern.
- ➔ Instalación y configuración de Apache Kafka.
- ➔ Desarrollo de un caso práctico de aplicación – Producer y Consumer.



GESTIÓN DE DATOS CACHE

- ➔ Introducción a la gestión de datos en cache.
- ➔ Usos, aplicaciones y principales tecnologías(Redis, Memcached y Spring Boot Cache).
- ➔ Introducción a Redis.
- ➔ Instalación y configuración de Redis.
- ➔ Desarrollo de un caso práctico de aplicación – optimización de consultas y registros.

SEMANA 05

QUERYS PATTERNS

- ➔ Introducción a Query patterns.
- ➔ Implementando Querys patterns(CQRS).
- ➔ Implementando Command - Database SQL.
- ➔ Implementando Query - Database NoSQL.
- ➔ Sincronización de data a través de mensajes utilizando Apache Kafka.

SAGA PATTERNS

- ➔ Introducción a SAGA patterns.
- ➔ ACID-2PC vs SAGA patterns (análisis comparativo).
- ➔ Tipos: Choreography vs Orchestration .
- ➔ Principales frameworks(Eventuate y Axon).
- ➔ Implementando SAGA para orquestación de procesos utilizando Apache Kafka.
- ➔ Seguimiento y monitoreo de consistencia de datos.



SEMANA 06

IMPLEMENTACIÓN DE MÉTRICAS

- ➔ Importancia de la Observabilidad de los microservicios.
- ➔ Pilares de observabilidad y tecnologías disponibles.
- ➔ Instalando y configurando Prometheus y Grafana.
- ➔ Habilitando end points y métricas de monitoreo.
- ➔ Creando dashboard de monitoreo (uso de recursos y disponibilidad).



IMPLEMENTACIÓN DE LOGS Y TRACING

- ➔ Introducción a Logs y Tracing.
 - Instalando y configurando el stack ELK/EFK .
 - Creando Logg de los microservicios a monitorear.
 - Instalando y configurando Jaeger.
 - Configurando la trazabilidad de los microservicios.

SEMANA 07

SPRING SECURITY (OAUTH2)

- ➔ Asegurando los microservicios .
- ➔ Creando se microservicio de autenticación distribuida(**authenticationservice**).
- ➔ Delegando autenticación a nivel de microservicios de manera automática (API Compositions).
- ➔ Habilitando endpoints y accesos para observabilidad de los microservicios.

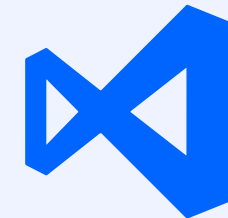
NOTA: La Aplicación solo permite demostrar la implementación de los temas de seguridad y disponibilidad de los microservicios, no contempla el desarrollo de todas las funcionalidades de los diferentes microservicios de la solución. Para ello existen cursos exclusivos de Full-Stack o Front-End.

DESARROLLO DE COMPONENTES UI

- ➔ Introducción a angular.
- ➔ Componentes UI (Client-side UI composition pattern) relacionados con la disponibilidad(Reliability) de los microservicios.
- ➔ Generación de tokens con OAuth2 .
- ➔ Uso en peticiones http(get/post, token).
- ➔ Pruebas de funcionalidad.



Spring
Security



OAuth 2.0

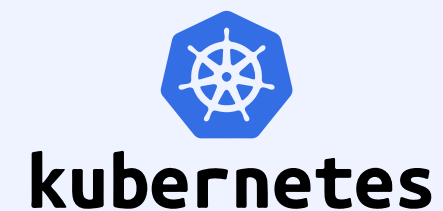
SEMANA 08

BUILD & CLOUD DEPLOYMENT (DOCKER+ KUBERNETES)

- ➔ Introducción a deployment patterns.
- ➔ Configuración de los microservicios para build(plugins) .
- ➔ Contenerizando los microservicios(imágenes).
- ➔ Builds y despliegue de los microservicios(Kubernetes) en cloud (Google Kubernetes Engine –GKE).
- ➔ Realización de pruebas .



OAuth 2.0



CIERRE DEL CURSO

- ➔ Presentación de proyectos por parte de los alumnos.
- ➔ Repaso general de los temas desarrollados.
- ➔ Entrega de certificados.
- ➔ Lecciones aprendidas.
- ➔ Próximos temas a investigar.

CRONOGRAMA

SEMANA	01		02		03		04	
ACTIVIDAD	Microservicios y arquitectura base		Configuración y descubrimiento		Configuración y descubrimiento		Comunicación asíncrona	
FECHA	29 SET	01 OCT	06 OCT	08 OCT	13 OCT	15 OCT	20 OCT	22 OCT

SEMANA	05		06		07		08	
ACTIVIDAD	Consulta y consistencia de datos		Pilares de observabilidad		Seguridad y desarrollo del Front-Angular		Despliegue y evaluación final	
FECHA	03 NOV	05 NOV	10 NOV	12 NOV	17 NOV	19 NOV	24 NOV	26 NOV

Este cronograma puede estar sujeto a cambios por parte de Galaxy Training.