

Módulo 1. Fundamentos de SOA, servicios y microservicios (4:00 horas) 65% práctica (2:30 hrs)

- 1.1 Motivadores de decisión de negocio y de tecnología respecto a SOA, servicios y microservicios.
- 1.2 Objetivos y beneficios estratégicos de la computación orientada a servicios.
- 1.3 Introducción a servicios y microservicios.
- 1.4 Características fundamentales de la arquitectura SOA.
- 1.5 Entendiendo el paradigma de diseño de la orientación a servicios incluyendo los cuatro pilares de adopción.
- 1.6 Introducción a capas de servicios, modelos de servicios y composición de servicios.
- 1.7 Inventario de servicios, gobierno y gestión de servicios y apis.
- 1.8 Introducción a tecnologías comunes de servicios, incluyendo api gateways, virtualización y contenerización.
- 1.9 Impactos de adopción, incluyendo consideraciones de gobierno, infraestructura, rendimiento y estandarización.

Módulo 2. Conceptos de tecnología de servicios (4:30 horas) 70% práctica (3:30 hrs)

- 2.1 Comparando los tipos de implementación de servicios.
- 2.2 Roles y agentes de servicios.
- 2.3 Patrones de intercambio de mensajes y actividad del servicio.
- 2.4 Conceptos básicos de XML, XML schema, JSON y JSON schema.
- 2.5 Cabeceras, códigos de respuesta y métodos del protocolo http.
- 2.6 Propiedades y reglas de diseño de REST.
- 2.7 Servicios REST, contratos, recursos y mensajería.
- 2.8 Hypermedia y vinculación tardía.
- 2.9 Conceptos básicos de wsdl y SOAP.
- 2.10 Web services, contratos, mensajería y registro.
- 2.11 Conceptos de la computación en la nube.
- 2.12 Escalamiento vertical y horizontal.
- 2.13 Multiarrendador, elasticidad y resiliencia.
- 2.14 Uso por demanda, acceso ubicuo y medida de uso.
- 2.15 Nubes públicas, privadas e híbridas.

Módulo 3. Fundamentos de la arquitectura de microservicios y contenerización (5:30 horas) 70% práctica (3:50 hrs)

- 3.1 Introducción a la arquitectura de microservicios
- 3.2 Desafíos comunes de diseño de los microservicios
- 3.3 Diseño de granularidad de los microservicios
- 3.4 Principios de diseño para los microservicios
- 3.5 Introducción a la contenerización
- 3.6 Contenerización vs. Virtualización
- 3.6 Elementos fundamentales de la arquitectura de contenedores
- 3.7 Motor de contenedores, archivos de compilación, imágenes y redes
- 3.8 Automatización de microservicios, trazabilidad y monitoreo
- 3.9 Registro de instancias del microservicio
- 3.10 Tipos, mecanismos y tecnologías de escalamiento
- 3.11 Motivadores de decisión de tecnología para el despliegue de microservicios en la nube
- 3.12 Patrones de diseño micro task abstraction y micro task segregation
- 3.13 Patrones de diseño rich container y logical pod container
- 3.14 Prácticas y beneficios de DevOps

Módulo 4. Arquitectura avanzada de microservicios y contenerización (6:30 horas) 80% práctica (5:15 hrs)

- 4.1 Patrones de diseño para la composición de microservicios
 - 4.1.1 Patrón de diseño “Autonomous proxy service”
 - 4.1.2 Patrón de diseño “Shared isolated database”
- 4.2 Patrones de diseño de capas y niveles de aislamiento de microservicios
 - 4.2.1 Patrón de diseño “Pre-defined data view”
 - 4.2.2 Patrón de diseño “Microservice instance registration”
- 4.3 Patrones de diseño “Workload distribution” y “service load balancing”
 - 4.3.1 Patrón de diseño “Synchronized cross-instance events”
 - 4.3.2 Patrón de diseño “Event-driven messaging”
- 4.4 Patrones de diseño “Atomic event processing” y “Appended events”
 - 4.4.1 Patrón de diseño “Centralized isolated state database”
 - 4.4.2 Patrón de diseño “Container chain”
 - 4.4.3 Patrón de diseño “Single-node multi-containers”
 - 4.4.4 Patrón de diseño “Multi-container isolation control”
 - 4.4.5 Patrón de diseño “Volatile configuration”
 - 4.4.6 Patrón de diseño “Serverless microservice deployment”
 - 4.4.7 Patrón de diseño “Dynamic scalability”
 - 4.4.8 Patrón de diseño “Micro scatter-gather”
 - 4.4.9 Patrón de diseño “Leader node election”
 - 4.4.10 Patrón de diseño “Redundant implementation”
 - 4.4.11 Patrón de diseño “Composition autonomy”
 - 4.4.12 Patrón de diseño “Container sidecar”
 - 4.4.13 Patrón de diseño “Microservice ambassador”
 - 4.4.14 Patrón de diseño “Log aggregator”
 - 4.4.15 Patrón de diseño “Distributed diagnostics”

Módulo 5. Taller práctico (7:30 horas) 85% práctica (6:20 hrs)

Como culminación del curso se llevará a cabo un taller práctico para aplicar patrones, principios, prácticas y procesos de diseño aprendidos en el curso.

Se utilizarán las siguientes tecnologías y herramientas de apoyo:

- Docker Desktop
- .NET
- Kubernetes
- Apache Kafka
- MongoDB
- Elastic
- Consul