

## Diplomado de Formación de Arquitectos de software

### Objetivo

Generar competencias técnicas que le permitan al egresado:

- Diseñar aplicaciones para que sean simples de mantener
- Diseñar aplicaciones para que observen buen desempeño
- Definir las normas de arquitectura y desarrollo de aplicaciones
- Auditar que las aplicaciones y desarrollos cumplan con las normas de arquitectura y desarrollo

### Temario por Módulo

#### Módulo 1: Diseño de aplicaciones.

- 1. Qué es un Arquitecto de software**
  - a. Qué es Arquitectura
  - b. Alcances de la Arquitectura
  - c. Responsabilidades clave
  - d. Cómo definir la Arquitectura
- 2. Conceptos clave del diseño orientado a objetos**
  - a. Acoplamiento y cohesión
  - b. Polimorfismo y Binding dinámico
  - c. Clases abstractas y servicios para clientes homogéneos
  - d. Interfaces y servicios para clientes heterogéneos
- 3. Diseño y modelado de Capas, componentes y dependencias**
  - a. Principio de responsabilidad única
  - b. Principio de dependencias estables
  - c. Principio de dependencias no cíclicas
  - d. Principio de inversión de dependencia
  - e. Cohesión y Acoplamiento
  - f. Principio de empaque de lo común
  - g. Principio de empaque por uso
  - h. Estrategia de manejo de errores
  - i. Aplicación de patrones de diseño
- 4. Análisis y diseño de casos de uso relevantes para la Arquitectura**
  - a. Principio de segregación de interfaz
  - b. Diseño de clases
  - c. Principio de abierto-cerrado

- d. Principio de abstracciones estables
- e. Principio de sustitución de Liskov
- f. Agregación y especialización a nivel caso de uso
- g. Interfaces de control
- h. Estrategias de colaboración: Orquestación vs Coreografía
- i. Principio de equivalencia uso-release

#### **5. Gestión del acoplamiento**

- a. Uso de Interfaces
- b. Uso de Herencia
- c. Introspección
- d. Inyección de dependencias

Duración: 30 horas

### **Módulo 2: Arquitectura de la persistencia**

#### **1. Diseño de datos para bases de datos SQL y NO-SQL**

- a. Unidades de almacenamiento: clases, tablas, documentos
- b. Llaves primarias
- c. Relaciones
  - i. Cardinalidades
  - ii. Navegación: Llaves foráneas
  - iii. Manejo de relaciones complejas
- d. Composición-Agregación
- e. Especialización y Herencia
- f. Diferencias entre bases de datos relacionales y no-relacionales
- g. Mapeo relacional
- h. Mapeo no-relacional
- i. Papel de los índices

#### **2. Estrategias de manejo de transacciones**

- a. ACID
- b. Enfoque pesimista vs enfoque optimista
- c. BASE

#### **3. Implementación de servicios de datos**

- a. Acceso a bases de datos relacionales desde Java con JDBC
  - i. Operaciones CRUD
  - ii. Búsquedas típicas
  - iii. Manejo Transaccional

- b. Acceso a MongoDB desde Java
  - i. Operaciones CRUD
  - ii. Búsquedas típicas
  - iii. Manejo Transaccional

Duración: 30 horas

### Módulo 3: Arquitectura Web e Interoperabilidad

- 1. El protocolo HTTP y las aplicaciones Web**
- 2. La especificación Java EE**
- 3. Arquitectura de los Componentes Web en Java EE**
  - a. Servlets
  - b. Filtros
  - c. Listeners
- 4. Aspectos relevantes en la arquitectura Web Java EE**
  - a. Ámbitos
  - b. Hilos y Concurrencia
  - c. Configuración
  - d. Estrategias de monitoreo
- 5. Estilo arquitectónico RESTful**
  - a. Principios
  - b. Implementación de Servidores JAX-RS
  - c. Implementación de Clientes JAX-RS
  - d. Manipulación de JSON en Java con API Estándar
- 6. Los clientes de una sola página JavaScript-HTML-CSS**
  - a. JavaScript como ecosistema tecnológico
  - b. Arquitecturas típicas en clientes de una sola página

Duración: 30 horas

### Módulo 4: Digitalización de procesos

- 1. Análisis de procesos para contextualización de requerimientos**
  - a. Análisis de actores
  - b. Relaciones entre actores
  - c. Levantamiento de procesos de negocio basado en responsabilidades
  - d. Instrumentos de Análisis
- 2. Identificación de módulos y componentes de negocio**
  - a. Modularización semántica
  - b. Uso de cadenas de valor

- c. Papel de las dependencias
- 3. Contextualización y Análisis de requerimientos**
  - a. Análisis top-down por sección
  - b. Análisis de Flujos de trabajo e identificación de mejoras vía TI
- 4. Innovación y digitalización de procesos**
  - a. Análisis de oportunidad de aplicación de TI
  - b. Análisis de implicaciones logísticas y económicas
  - c. Digitalización de flujos de trabajo
- 5. Diseño detallado de procesos digitales**
  - a. Definición de casos de uso
  - b. Diseño técnico de casos de uso

Duración: 30 horas

### Módulo 5: Diseño de Microservicios

- 1. Principios arquitectónicos de los microservicios**
  - a. Enfoques de modularización. Ventajas y desventajas
  - b. Técnicas de modularización
  - c. Equilibrio entre redundancia e integridad
  - d. Orquestación vs Coreografía
- 2. Orquestación de servicios**
  - a. Implementación de servicios RESTful con Spring MVC
  - b. Api de manipulación de objetos JSON
  - c. Desarrollo de clientes con RestTemplate
  - d. Diseño del manejo de errores
- 3. Integración de servicios usando Coreografía**
  - a. Patrones observer y cola de tareas
  - b. Rabbitmq Message Broker
  - c. Desarrollo de publicadores de Rabbitmq
  - d. Desarrollo de suscriptores de Rabbitmq
- 4. Registro y auto-reconocimiento de servicios**
  - a. Eureka Server
  - b. Uso de Eureka Server para publicar servicios
  - c. Uso de Eureka Client para descubrir servicios
- 5. Balanceo de carga en cliente**
  - a. Round Robin
  - b. Ribbon integrado con Eureka
  - c. Uso como Api
  - d. Uso de Ribbon integrado con RestTemplate
- 6. Uso de Corta-circuitos**
  - a. Hystrix

- b. Implementación de Circuit breaker con Hystrix
- c. Monitoreo con Hystrix

Duración: 30 horas

## Módulo 6: Gestión de código

### 1. Control de versiones

- a. Conceptos centrales
- b. Git y su arquitectura general
- c. Uso a nivel individuo
- d. Uso a nivel equipo de trabajo
- e. Gestión de los cambios
- f. Uso de Pruebas automatizadas para validar la integración

### 2. Integración continua

- a. Conceptos centrales
- b. Jenkins y su arquitectura general
- c. Uso a nivel individuo
- d. Uso a nivel equipo de trabajo
- e. Gestión de los cambios

Duración: 30 horas

## Prerrequisitos

El alumno debe:

- Tener al menos 2 años de experiencia desarrollando soluciones tecnológicas.
- Saber programar en Java o en C#