Diplomado de Formación de Arquitectos de software

Objetivo

Generar competencias técnicas que le permitan al egresado:

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

Temario por Módulo

Módulo 1: Digitalización de Procesos

1. **Análisis de procesos de negocio para contextualización de requerimientos**
    a. Análisis de actores
    b. Levantamiento de procesos de negocio
    c. Instrumentos de Análisis
2. **Alineación de Módulos al 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
    b. Análisis de Flujos de trabajo e identificación de mejoras vía TI
4. **Innovación y digitalización de procesos**
    a. Diagnóstico de procesos
    b. Diseño funcional de procesos digitales
    c. Diseño no funcional de procesos digitales
    d. Análisis de implicaciones logísticas y económicas
5. **Diseño detallado de procesos digitales**
    a. Definición de casos de uso
    b. Diseño técnico de casos de uso

Duración: 30 horas

Módulo 2: Diseño de aplicaciones.

1. **Conceptos de Arquitectura**
    a. Qué es Arquitectura
    b. Alcances de la Arquitectura
    c. Responsabilidades clave
    d. Cómo definir la Arquitectura
2. **Fundamentos 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
    e. Introspección
    f. Inyección de dependencias
3. **Diseño 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. Principio de empaque de lo común
    f. Principio de empaque por uso
    g. Estrategia de manejo de errores
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. Estrategias de colaboración: Orquestación vs Coreografía
    g. Principio de equivalencia uso-release

Duración: 30 horas

Módulo 3: 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. Uso de bases de datos relacionales con Java
   b. Uso de bases de datos NO SQL con Java

Duración: 30 horas

Módulo 4: Arquitectura Web e Interoperabilidad

1. **El protocolo HTTP**
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 Web APIs
   c. Implementación de Clientes de Web APIs
   d. Manipulación de JSon en Java
   e. Estandarización: OPEN API Specification
   f. Swagger y documentación de APIs
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 5: Fundamentos de Seguridad Web

1. **Principios de Seguridad OWASP**
2. **Encriptado**
    a. Encriptado simétrico
    b. Encriptado asimétrico
    c. Aplicaciones comunes
3. **Autenticación**
    a. Conceptos clave
    b. Modelos
    c. Implementación basada en Spring Security
4. **Autorización**
    a. Conceptos clave
    b. Modelos
        i. Autorización basada en roles
        ii. Autorización basada en atributos
    c. Implementación basada en Spring Security
5. **Implementación de Tokens JWT**
    a. Autenticación y Autorización JWT
    b. Codificación base 64
    c. Estructura de los tokens JWT
    d. Autenticación y autorización con tokens en APIs Rest
    e. API jjwt
    f. Implementación de la Autenticación
    g. Implementación de la Autorización
6. **OAuth 2.0**
    a. Qué es el OAuth 2.0 Authorization Framework
    b. Login OAuth de Spring Security

Duración: 30 horas

Módulo 6: Diseño de Microservicios

7. **Principios arquitectónicos de los microservicios**
    a. Sistemas distribuidos
    b. Equilibrio entre redundancia e integridad
    c. Orquestación vs Coreografía
    d. Event Driver Architechure
    e. Resiliencia
8. **Orquestación de servicios**
    a. Implementación de servicios RESTful
    b. Manipulación de objetos JSon

   c. Desarrollo de consumidores de APIs Web
   d. Diseño del manejo de errores
9. **Integración de servicios usando Coreografía**
   a. Conceptos de Message Brokers
   b. Desarrollo de publicadores en Message Broker
   c. Desarrollo de suscriptores en Message Broker
10. **Implementación de Resiliencia**
   a. Registro y auto-reconocimiento de servicios
   b. Balanceo de carga en cliente
   c. Uso de Corta-circuitos

Duración: 30 horas

Módulo 7: Gestión de código

1. **Sistematización de Pruebas**
   a. Tipo de pruebas
   b. Diseño de pruebas sistematizadas
   c. Uso de motores de pruebas (Junit, librerías de asserts, etc.)
   d. Test-Driven Development (TDD)
2. **Control de código a nivel Proyecto**
   a. Ciclos de Vida
   b. Sistematización del ciclo de compilación-pruebas-empacado
3. **Control de Versiones**
   a. Conceptos centrales (versionamiento semántico, snapshots y releases)
   b. Git y su arquitectura general
   c. Uso a nivel individuo
   d. Uso a nivel equipo de trabajo
4. **Integración continua y DevOps**
   a. Conceptos centrales

Duración: 30 horas

Prerrequisitos

El alumno debe:

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