

# **Conceptos Teóricos sobre Arquitectura de Software**

# 1. Definición y Propósito

- La arquitectura de software es la estructura de un sistema que comprende componentes, sus propiedades visibles externamente y las relaciones entre ellos.
- Sirve como un plano para guiar el desarrollo del sistema y asegurar que los requisitos se cumplan eficientemente.

## 2. Componentes Principales

- **Componentes:** Unidades funcionales como clases, módulos o servicios.
- **Conectores:** Mecanismos de interacción como APIs o protocolos.
- **Configuración:** Disposición de componentes y conectores.

# 3. Atributos de Calidad

- **Escalabilidad:** Capacidad de manejar mayor carga.
- **Mantenibilidad:** Facilidad para modificar el sistema.
- **Seguridad:** Protección contra accesos no autorizados.
- **Rendimiento:** Respuesta y procesamiento eficiente.
- **Disponibilidad:** Tiempo en que el sistema está operativo.

# 4. Estilos Arquitectónicos

- Arquitectura en capas.
- Microservicios.
- Cliente-servidor.
- Pipes and Filters.

# 5. Decisiones Arquitectónicas

Decisiones fundamentales sobre la estructura del sistema que impactan a largo plazo:

- Selección de componentes.
- Tecnologías a utilizar.
- Lenguajes de programación y marcos de trabajo.

# 6. Diagramas y Vistas

- **Vista lógica:** Componentes funcionales.
- **Vista de procesos:** Interacción entre procesos en tiempo de ejecución.
- **Vista física:** Despliegue del software en hardware.

## **7. Arquitectura Orientada a Servicios (SOA)**

En SOA, los servicios son las unidades fundamentales de funcionalidad y son accesibles a través de una red.



## **8. Documentación de la Arquitectura**

Es crucial documentar la arquitectura para que todos los involucrados comprendan las decisiones arquitectónicas clave.

# 9. Principios SOLID

- S: Principio de responsabilidad única.
- O: Principio de abierto/cerrado.
- L: Principio de sustitución de Liskov.
- I: Principio de segregación de interfaces.
- D: Principio de inversión de dependencia.