

Propuesta de Protocolo

2024

30 de Enero

INSTITUTO TECNOLÓGICO
DE CHILPANCINGO

Alumno: Perea Valdez Leonardo Alberto

Arquitectura de Software

Definición

La arquitectura de software es la disciplina responsable del diseño y la estructuración de sistemas de software complejos. Es un elemento fundamental en el ciclo de vida del desarrollo de software, ya que proporciona una visión global del sistema, guía la toma de decisiones técnicas y facilita la comunicación entre los diferentes actores involucrados. La arquitectura de software se encarga de definir la estructura lógica y física de un sistema, incluyendo sus componentes, sus interfaces, sus interacciones y sus propiedades.

La arquitectura de software se puede dividir en dos partes principales:

- **Arquitectura lógica:** Describe la estructura interna del sistema, sus componentes, interfaces y las relaciones entre ellos. Define cómo se organizan los diferentes módulos de software y cómo interactúan entre sí.
- **Arquitectura física:** Describe la implementación física del sistema, incluyendo su hardware, software de sistema operativo, middleware, infraestructura de red y distribución de componentes. Define cómo se mapea la arquitectura lógica a los recursos físicos y cómo se distribuye el sistema en diferentes entornos.

La importancia de la arquitectura de software radica en los siguientes aspectos:

- **Cumplimiento de requisitos:** La arquitectura de software proporciona un marco para definir y garantizar el cumplimiento de los requisitos funcionales y no funcionales del sistema, como rendimiento, escalabilidad, seguridad y mantenibilidad.
- **Facilita el mantenimiento y la evolución:** Una arquitectura bien diseñada facilita la comprensión del sistema y su mantenimiento a largo plazo, permitiendo su evolución y adaptación a nuevos requisitos.
- **Mejora la calidad del software:** Una arquitectura sólida contribuye a la calidad del software al promover la modularidad, la reutilización de componentes, la capacidad de prueba y la separación de preocupaciones.
- **Comunicación y colaboración:** La arquitectura de software proporciona un lenguaje común y una visión compartida para todos los involucrados en el desarrollo del sistema, facilitando la comunicación y la colaboración efectiva.

Tipos de Arquitectura de Software

Existen varios enfoques y patrones arquitectónicos comúnmente utilizados en el diseño de sistemas de software. Algunos de los tipos más relevantes incluyen:

1. **Arquitectura Cliente-Servidor:** En este modelo, los clientes envían solicitudes a un servidor central, el cual procesa las solicitudes y devuelve las respuestas correspondientes. Es ampliamente utilizado en aplicaciones web y sistemas empresariales.
2. **Arquitectura de Red entre Pares (Peer-to-Peer):** En esta arquitectura, todos los nodos de la red tienen roles equivalentes y pueden actuar tanto como clientes como servidores, permitiendo el intercambio descentralizado de recursos y tareas.
3. **Arquitectura Modelo-Vista-Controlador (MVC):** Este patrón arquitectónico separa la lógica de la aplicación en tres componentes: el modelo (datos y lógica de negocio), la vista (interfaz de usuario) y el controlador (manejo de eventos y flujo de control). Es ampliamente utilizado en el desarrollo de aplicaciones web y de escritorio.
4. **Arquitectura Orientada a Eventos (EDA):** En este enfoque, los componentes del sistema responden a eventos generados de forma asincrónica. Es útil para aplicaciones en las que las acciones del usuario o los cambios en el sistema desencadenan respuestas inmediatas, como sistemas de mensajería, monitoreo en tiempo real y aplicaciones de Internet de las cosas (IoT).
5. **Arquitectura de Microservicios:** Consiste en desarrollar una aplicación como un conjunto de pequeños servicios independientes y desacoplados, cada uno encargado de una tarea específica. Estos servicios se comunican a través de interfaces bien definidas (API) y pueden ser implementados y escalados de forma independiente.
6. **Arquitectura en Capas:** Este enfoque organiza la aplicación en capas lógicas, como la capa de presentación, la capa de lógica de negocio y la capa de acceso a datos. Cada capa se comunica solo con la capa adyacente, promoviendo la modularidad, la reutilización y el mantenimiento.
7. **Arquitectura Hexagonal (Puertos y Adaptadores):** También conocida como "puertos y adaptadores", esta arquitectura separa la lógica de negocio de los detalles de implementación, como interfaces de usuario, bases de datos o servicios externos. Esto facilita la prueba, el mantenimiento y la capacidad de cambiar componentes sin afectar el núcleo del sistema.