

Tipos de Patrones.

Arquitectura de SW

Introducción y conceptos

Patrón de Capas

Organiza el software en capas, donde cada capa tiene una responsabilidad específica.

Ejemplo: Aplicaciones web con capas de presentación, lógica y datos.

Ventajas: Simplifica el mantenimiento y la separación de preocupaciones.

Patrón Cliente-Servidor

Divide el sistema en cliente y servidor. El cliente solicita servicios y el servidor responde.

Ejemplo: Navegador web y servidor.

Ventajas: Escalabilidad y centralización de datos.

Patrón Maestro-Escavo

El maestro distribuye tareas a esclavos que las ejecutan. El maestro recolecta los resultados.

Ejemplo: Bases de datos replicadas.

Ventajas: Paralelización y mayor rendimiento.

Patrón de Filtro de Tubería

Procesa datos en etapas, donde cada etapa es un filtro y los datos fluyen entre filtros.

Ejemplo: Pipelines de procesamiento de audio.

Ventajas: Modularidad y flexibilidad.

Patrón de Intermediario

- Introduce un intermediario que facilita la comunicación entre componentes desacoplados.
- **Ejemplo:** RabbitMQ, Kafka.
- **Ventajas:** Desacoplamiento y escalabilidad.

Patrón de Igual a Igual

Carga de trabajo distribuida entre nodos que pueden ser clientes y servidores.

Ejemplo: Redes P2P como BitTorrent.

Ventajas: Alta escalabilidad y robustez.

Patrón de Bus de Evento

Canal centralizado para la comunicación entre componentes a través de eventos.

Ejemplo: Arquitecturas de microservicios con bus de eventos.

Ventajas: Desacoplamiento y flexibilidad.

Modelo-Vista-Controlador

Separa la aplicación en modelo, vista y controlador.

Ejemplo: Aplicaciones web.

Ventajas: Organización del código y facilidad de pruebas.

Patrón de Pizarra

Múltiples módulos interactúan a través de una pizarra compartida hasta llegar a una solución.

Ejemplo: Sistemas de IA y reconocimiento de voz.

Ventajas: Flexibilidad y colaboración.

Patrón de Intérprete

Define una representación de la gramática de un lenguaje y un intérprete para evaluarla.

Ejemplo: Intérpretes de expresiones matemáticas.

Ventajas: Creación de lenguajes específicos y evaluación simplificada.