

# Estilos de Arquitectura

## 1. Cliente-Servidor

- Definición: Modelo donde los clientes solicitan servicios a un servidor.
- Características
  - Centralización de recursos
  - Interacción cliente-servidor
  - Escalabilidad
- Ventajas:
  - Fácil gestión de recursos
  - Mantenimiento centralizado
- Desventajas
  - Puntos únicos de falla
  - Dependencia del servidor

## 2. Microservicios

- Definición: Descomposición de aplicaciones en servicios pequeños e independientes.
- Características
  - Despliegue independiente
  - Comunicación a través de APIs
- Ventajas
  - Escalabilidad horizontal
  - Flexibilidad en el desarrollo
- Desventajas
  - Complejidad en la gestión
  - Necesidad de orquestación

## 3. Arquitectura en Capas

- Definición: Divide la aplicación en capas que manejan diferentes aspectos.
- Características
  - Modularidad
  - Separación de preocupaciones (presentación, lógica, datos)
- Ventajas
  - Mantenimiento simplificado
  - Reutilización de componentes
- Desventajas
  - Posible sobrecarga en la comunicación entre capas

## 4. Arquitectura Orientada a Servicios (SOA)

- Definición: Comunicación entre componentes distribuidos a través de servicios.
- Características
  - Interoperabilidad
  - Uso de estándares abiertos (SOAP, REST)
- Ventajas
  - Reutilización de servicios existentes
  - Flexibilidad y adaptabilidad
- Desventajas
  - Complejidad en la integración
  - Latencia en las llamadas a servicios

## 8. Serverless

- Definición: Despliegue de aplicaciones sin gestionar servidores físicos; ejecución bajo demanda.
- Características
  - Escalabilidad automática según demanda
  - Pago por uso (costo eficiente)
- Ventajas
  - Reducción significativa de costos operativos
  - Enfoque en el desarrollo del código sin preocuparse por la infraestructura
- Desventajas
  - Aumento potencial en latencia
  - Limitaciones en el uso de bibliotecas y dependencias

## 7. Microkernel

- Definición: Sistema central básico al que se le añaden plug-ins para funcionalidades adicionales.
- Características
  - Extensibilidad mediante módulos
  - Independencia de componentes principales y plug-ins
- Ventajas
  - Flexibilidad para añadir nuevas funcionalidades
  - Mantenimiento simplificado de componentes centrales
- Desventajas
  - Dependencia entre plug-ins puede complicar el sistema

## 6. Arquitectura Basada en Espacio

- Definición: Utiliza espacios de objetos donde se almacenan y acceden a datos.
- Características
  - Alto rendimiento y eficiencia
  - Acceso a datos en memoria
- Ventajas
  - Escalabilidad horizontal
  - Reducción de latencia en accesos a datos
- Desventajas
  - Complejidad en la gestión del espacio

## 5. Arquitectura Dirigida a Eventos

- Definición: Basada en la reacción a eventos para activar funciones o procesos.
- Características
  - Desacoplamiento entre componentes
  - Manejo eficiente de eventos en tiempo real
- Ventajas
  - Alta reactividad
  - Escalabilidad dinámica
- Desventajas
  - Complejidad en el manejo del flujo de eventos