

COMO ELEGIR LA ARQUITECTURA DE LA APLICACION SEGUN SU TIPO

D. F. Velásquez Pichilla
7690-16-3882 Universidad Mariano Gálvez
Seminario de Tecnología de Información
dvelasquezp4@miumg.edu.gt

Resumen

La arquitectura de software es el plano estructural sobre el que se construye cualquier aplicación, y su correcta elección incide directamente en el rendimiento, la escalabilidad y la facilidad de mantenimiento del sistema. Este artículo expone criterios y buenas prácticas para seleccionar la arquitectura más adecuada según el tipo de aplicación, considerando factores como el volumen de usuarios, la complejidad del dominio, el tiempo de desarrollo, la disponibilidad de recursos y la experiencia técnica del equipo. Se analizan las características, ventajas y limitaciones de las arquitecturas monolítica, en capas, orientada a servicios, basada en microservicios, sin servidor y orientada a eventos, con ejemplos de contextos reales en los que resultan convenientes. Finalmente, se ofrecen recomendaciones para validar la decisión arquitectónica mediante prototipos y métricas no funcionales antes de implementarla a gran escala.

Palabras claves: arquitectura de software, escalabilidad, microservicios, patrones de diseño, mantenibilidad

Desarrollo del tema

Seleccionar la arquitectura correcta implica comprender no solo el funcionamiento interno del sistema, sino también su entorno operativo y los cambios que experimentará en el tiempo. La decisión debe ser estratégica, considerando tanto las necesidades presentes como la capacidad de evolución futura.

Factores esenciales para la elección:

- **Escalabilidad:** Capacidad para incrementar recursos sin degradar el rendimiento.
- **Mantenibilidad:** Facilidad de adaptación ante nuevas funcionalidades o cambios en los requisitos.
- **Tiempo de desarrollo:** Arquitecturas más complejas requieren mayor planificación y pruebas.
- **Presupuesto:** El costo de infraestructura y mantenimiento varía según el modelo elegido.
- **Experiencia técnica:** La arquitectura debe ser viable para las competencias del equipo.

Principales tipos de arquitecturas y su contexto de uso:

1. **Monolítica:** Todo el sistema se ejecuta como una única unidad de despliegue. Es adecuada para aplicaciones pequeñas o medianas con requisitos estables y un equipo reducido.
2. **En capas:** Organiza la aplicación en niveles separados (presentación, lógica y datos), lo que mejora la organización y facilita la sustitución de componentes.

3. **Orientada a servicios (SOA):** Divide las funcionalidades en servicios independientes con contratos bien definidos, favoreciendo la reutilización en entornos corporativos.
4. **Microservicios:** Conjunto de servicios pequeños y autónomos, cada uno con su propia base de datos, escalables de forma independiente. Útiles en sistemas grandes y con equipos especializados.
5. **Sin servidor (Serverless):** Funciones alojadas en la nube que se ejecutan bajo demanda, eliminando la gestión de servidores. Ideal para cargas variables o eventos esporádicos.
6. **Basada en eventos:** Arquitectura reactiva que responde a sucesos en tiempo real, apropiada para sistemas de monitoreo, IoT o procesamiento en streaming.

Ejemplo de selección: Una aplicación interna para una pequeña empresa puede beneficiarse de un modelo monolítico, evitando sobrecostos. En cambio, una plataforma de comercio electrónico internacional puede requerir microservicios combinados con un enfoque basado en eventos para soportar picos de demanda y garantizar disponibilidad continua.

La decisión debe validarse mediante **prototipos** que simulen la carga real, midan la latencia y evalúen el costo total de propiedad (TCO). También es fundamental anticipar la integración de herramientas de monitoreo y registro desde el inicio.

Observaciones y comentarios

En muchos casos, las fallas no provienen de errores en el código, sino de una elección arquitectónica que no correspondía al tipo de aplicación ni a las capacidades del equipo. Adoptar un enfoque iterativo, comenzando con soluciones simples y evolucionando hacia modelos más complejos, puede evitar sobrecostos y tiempos muertos.

Conclusiones

1. La arquitectura debe responder al tipo de aplicación y a sus requisitos no funcionales.
2. Factores como escalabilidad, costo y experiencia del equipo son determinantes.
3. No existe una arquitectura perfecta; el contexto define la mejor elección.
4. La validación temprana reduce el riesgo de rediseños costosos.

Bibliografía

- Bass, L., Clements, P., & Kazman, R. (2021). *Software Architecture in Practice* (4th ed.). Addison-Wesley.
- Richards, M., & Ford, N. (2020). *Fundamentals of Software Architecture*. O'Reilly Media.
- Newman, S. (2021). *Building Microservices* (2nd ed.). O'Reilly Media.
- Fowler, M. (2015). *Microservices: a definition of this new architectural term*. martinowler.com.