

LA MEJOR ARQUITECTURA SEGUN LA APLICACION A DESARROLLAR

*C. A. Ramírez García
7690-21-10603 Universidad Mariano Gálvez
Seminario de Tecnologías de Información
cramirezg19@miumg.edu.gt*

Repositorio

<https://github.com/CarlosRamirez2003/FORO-2.git>

Resumen

La selección de la arquitectura de una aplicación es un paso clave en el desarrollo de software, ya que influye directamente en su rendimiento, escalabilidad, facilidad de mantenimiento y capacidad de adaptación futura. Entre las arquitecturas más comunes se encuentran la monolítica, en capas (N-Tiers), de microservicios y serverless o basada en eventos, cada una con ventajas y limitaciones que deben evaluarse de acuerdo con las necesidades específicas del proyecto. Factores como el tipo de aplicación (móvil, bancaria, web, de escritorio, etc.), la escalabilidad requerida, el desempeño esperado, la mantenibilidad, la complejidad del dominio, el presupuesto disponible y el grado de integración con otros sistemas son determinantes en la elección. Una arquitectura mal seleccionada puede ocasionar sobrecostos y problemas técnicos, mientras que una elección correcta optimiza recursos y asegura la estabilidad a largo plazo. Por ello, es esencial realizar un análisis previo y elegir la solución que satisfaga las necesidades presentes y permita crecer en el futuro.

Palabras clave: arquitectura, escalabilidad, usuarios , datos , micro servicios

Desarrollo del tema

Existen diferentes tipos de arquitecturas de las cuales las principales son: Arquitectura Monolítica, Arquitectura en Capas (N-Tiers o Three-Tier), Arquitectura de Microservicios y Arquitectura Serverless o basada en eventos.

para saber que arquitectura se acompla tambien dependerá de el tipo de aplicacion que se vaya a desarrollar ya sean mobiles , bancarias , web ejecutables etc, se tienen que analizar puntos clave como: Escalabilidad: ¿Necesitará crecer en usuarios y datos rápidamente?, Desempeño: ¿Requiere baja latencia o tiempo real?, Mantenibilidad: ¿Será mantenida por un equipo grande o pequeño?, Complejidad del dominio: ¿El negocio tiene muchas reglas y flujos?, Presupuesto y tiempo: ¿Es un prototipo rápido o un sistema crítico? y Integraciones: ¿Dependerá de otros sistemas o APIs externas?

la arquitectura monolítica es una de las más antiguas y utilizadas en proyectos pequeños y medianos. Consiste en desarrollar toda la aplicación como una sola unidad, lo que facilita el despliegue y el mantenimiento inicial, ya que todo el código está en un único lugar. Por ejemplo, un sistema de gestión para una pequeña tienda podría construirse de forma monolítica para reducir costos y tiempos. Sin embargo, a medida que crece el número de usuarios y funcionalidades, esta arquitectura puede volverse difícil de escalar y mantener.

La arquitectura en capas o N-Tiers es muy común en entornos corporativos y académicos. Divide la aplicación en capas como presentación, lógica de negocio y acceso a datos. Esto permite separar responsabilidades, de manera que un cambio en la interfaz no afecte la lógica interna. Por ejemplo, un sistema bancario podría utilizar esta arquitectura para aislar la capa de datos de las operaciones de usuario, garantizando mayor seguridad y facilidad de mantenimiento.

En cambio, la arquitectura de microservicios es ideal para proyectos grandes o con múltiples módulos. Cada servicio es independiente y se comunica con los demás a través de APIs, lo que permite actualizaciones y despliegues sin afectar a todo el sistema. Grandes empresas como Netflix o Amazon usan esta arquitectura para manejar millones de transacciones simultáneas. La desventaja es que requiere una infraestructura más avanzada, un equipo capacitado y una buena estrategia de comunicación entre servicios.

Por último, la arquitectura serverless o basada en eventos está ganando popularidad. En este modelo, el código se ejecuta bajo demanda en plataformas en la nube, pagando solo por el tiempo de ejecución. Es útil para aplicaciones que no requieren un servidor activo todo el tiempo, como procesamiento de imágenes, notificaciones push o análisis de datos en tiempo real. Por ejemplo, un sistema que procese información meteorológica cada hora podría beneficiarse de esta arquitectura para reducir costos operativos.

Observaciones y comentarios

Durante el desarrollo de este trabajo se pudo identificar la importancia de analizar previamente las necesidades de un proyecto antes de seleccionar la arquitectura. Se observó que una mala elección puede generar sobrecostos y problemas de mantenimiento, mientras que una decisión correcta facilita la escalabilidad y el rendimiento. También se evidenció que la experiencia del equipo de trabajo influye directamente en la factibilidad de implementar arquitecturas más complejas como microservicios o serverless. En futuros proyectos, se recomienda realizar pruebas piloto con distintas arquitecturas para evaluar su viabilidad antes de implementarlas de forma definitiva.

Conclusiones

La elección de la arquitectura de una aplicación debe basarse en un análisis detallado de factores como escalabilidad, rendimiento, mantenibilidad, complejidad del dominio, presupuesto y nivel de integración requerido.

No existe una arquitectura universalmente superior; la idoneidad de cada una depende del tipo de aplicación y del contexto en el que será implementada.

Las arquitecturas monolíticas resultan adecuadas para proyectos pequeños o con recursos limitados, mientras que microservicios y serverless se recomiendan para sistemas de gran escala o con alta demanda de disponibilidad y flexibilidad.

Adoptar una arquitectura sin una planificación adecuada puede generar sobrecostos, problemas de rendimiento y dificultades en el mantenimiento a largo plazo.

Es recomendable iniciar con una arquitectura sencilla que cubra las necesidades actuales, pero que permita escalar y adaptarse a medida que crezcan las demandas del negocio o la aplicación.

E-Grafia

<https://dev.to/yokwejuste/choosing-the-best-architecture-for-your-software-23h9>

<https://www.digiteum.com/web-application-architecture/>

<https://appmaster.io/es/blog/como-elegir-la-arquitectura-de-software>