



ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS



Nombre: Miguel Angel Muenala Quinga

Grupo: Gr1SW

Fecha: 02/08/2023

Tema: Ventajas, Desventajas, Similitudes y diferencias de Monolitos, SOA, Microservicios.

La arquitectura de software juega un papel fundamental en la configuración del diseño, el desarrollo y la escalabilidad de las aplicaciones modernas. Tres enfoques arquitectónicos destacados, monolitos, arquitectura orientada a servicios (SOA) y microservicios, han surgido como contendientes para estructurar sistemas de software complejos. Cada uno de estos enfoques tiene su propio conjunto de ventajas, desventajas, similitudes y diferencias que justifican un análisis exhaustivo.

En la arquitectura de software moderna, los monolitos, la arquitectura orientada a servicios (SOA) y los microservicios representan enfoques distintos para crear aplicaciones complejas. Los monolitos ofrecen simplicidad y facilidad de implementación, pero pueden volverse difíciles de manejar a medida que crecen las aplicaciones. SOA mejora la modularidad y la interoperabilidad, pero introduce complejidades de comunicación. Los microservicios brindan escalabilidad granular y aislamiento de fallas, pero exigen una mayor sobrecarga de administración.

En cuanto a similitudes, los tres tienen como objetivo mejorar la mantenibilidad y la modularidad y admiten diferentes tecnologías y lenguajes.

La elección de la arquitectura adecuada depende de las características específicas del proyecto, como la naturaleza de la aplicación y las necesidades de escalabilidad. Cada enfoque tiene sus pros y sus contras, moldeando el panorama del software moderno y abordando diversos desafíos de desarrollo.

En conclusión, las opciones de desarrollo y arquitectura entre Monoliths, SOA y Microservices ofrecen un espectro diverso de beneficios y desafíos a considerar. Los monolitos brindan simplicidad en el desarrollo y la implementación, pero pueden sufrir problemas de escalabilidad y dificultad en el mantenimiento. SOA hace hincapié en la reutilización y la interoperabilidad, pero puede introducir complejidad y posibles cuellos de botella en el rendimiento. Por otro lado, los microservicios sobresalen en escalabilidad, aislamiento de fallas y heterogeneidad tecnológica, al tiempo que exigen una mayor sobrecarga operativa. Si bien estos enfoques comparten el objetivo de crear sistemas de software funcionales, sus ventajas y desventajas en términos de facilidad de desarrollo, escalabilidad, mantenimiento e integración del sistema deben evaluarse cuidadosamente en consonancia con los requisitos específicos del proyecto y los objetivos comerciales. Cada arquitectura tiene su propio lugar dentro del panorama del software, y elegir la correcta implica un análisis cuidadoso de las compensaciones y una comprensión profunda del contexto del proyecto.