



ESCUELA POLITÉCNICA NACIONAL
Facultad de Ingeniería de Sistemas
Aplicaciones Web Avanzadas (ISWD813)

Ensayo
SOA vs MSA

Integrantes:
David E. Morales M.

Docente:
Vicente A. Egüez S.

Fecha:
Lunes 7 de Marzo de 2022

Introducción

En la actualidad, el entorno empresarial es más complejo e impredecible, estas dinámicas de mercado dan a lugar la necesidad de flexibilidad para enfrentar cambios continuos y mantenerse al margen tecnológico. Es así, que el desarrollo de sistemas monolíticos se ha convertido en cosa del pasado, las nuevas tecnologías permiten construir software modular, escalable, efectivo y eficiente. En el presente ensayo se estudiarán dos enfoques de desarrollo de software modular, la Arquitectura Orientada a Servicios y la Arquitectura de Microservicios, arquitecturas en las que las funcionalidades se consideran servicios compartidos y reutilizables, con la finalidad de establecer beneficios y desventajas de ambos enfoques.

Desarrollo

Durante la última década, la globalización, las economías estrictas, procesos comerciales tercerizados y una industria cada vez mas regularizada, obligan a las empresas a transformar las maneras en que brindan sus servicios. En respuesta a estas exigencias aparecen las siguientes arquitecturas que serán detalladas a profundidad a continuación. [1]

SOA

La Arquitectura Orientada a Servicios (SOA) se presenta como un enfoque atractivo para integrar aplicaciones empresariales pues busca dar una mejor interoperabilidad entre componentes y costos reducidos en el desarrollo. Este estilo arquitectónico emergente para el desarrollo e integración de aplicaciones empresariales es un marco de trabajo en el cual los servicios son construidos, desplegados, gestionados y orquestados siguiendo una infraestructura ágil y resiliente.

La necesidad de implementar una arquitectura orientada a servicios es producto de la inversión de aplicaciones de software a gran escala por parte de grandes empresas para manejar sus negocios, estas aplicaciones a menudo son heterogéneo en varias plataformas, sistemas operativos e idiomas.

Con su naturaleza débilmente acoplada, SOA permite a las empresas conectar nuevos servicios o actualizar servicios existentes, brindando oportunidades para aumentar la agilidad empresarial y respondiendo a la demanda.

En este contexto, un servicio es un componente de software que puede ser reutilizado por otro componente o accedido por medio de la red. Un aspecto importante dentro de SOA es la separación de la interfaz del servicio (QUÉ) que brinda la identificación del servicio, de su implementación o contenido (CÓMO) que brinda la lógica del negocio. [2]

Zimmermann sugiere tres niveles de abstracción:

- *Operaciones*: Unidades de funciones que operan recibiendo datos, en interfaces específicas y retornando respuestas estructuradas.
- *Servicios*: Operaciones y agrupamientos lógicos.
- *Procesos de Negocio*: Acciones o actividades para llevar a cabo metas de negocio invocando múltiples servicios.

Se pueden prever tres tipos de servicios:

- *Servicios de Infraestructura*: Para incluir seguridad, monitoreo o administración.
- *Servicios Neutros para el Negocio*: Incluyen manejadores de servicios, notificaciones y flujos de servicios.
- *Servicios de Negocio*: Incluyen servicios basados en el dominio del negocio. (Verificación de direcciones, tarjetas de crédito, inventario, etc.)

Es importante mencionar que SOA emplea un paradigma de encontrar-enlazar-ejecutar, los componentes principales incluyen:

- *Proveedores de Servicios*: Componentes que ejecutan funciones de negocio usando entradas y produciendo salidas, ofrecen los servicios vía intranet o internet.
- *Consumidores de Servicios*: Componentes en la arquitectura que consumen servicios. Estos buscan servicios (basados en algún criterio), cuando lo encuentran se realiza un enlace dinámico.

- *Registro de Servicios*: Repositorio con las descripciones de los servicios para conocer como se debe acceder a los mismos.

A continuación, podemos ver la estructura de la Arquitectura Orientada a Servicios SOA.

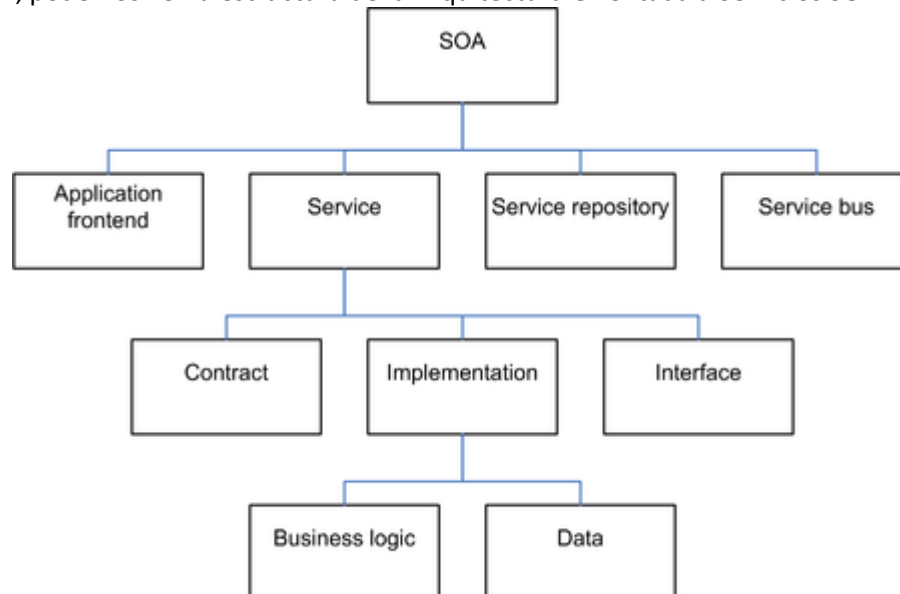


Ilustración 1 Elementos de una arquitectura SOA [3]

SOA es un paradigma para el diseño de una arquitectura de software que activa una interacción entre proveedores con capacidades para abordar problemas comerciales y consumidores para quienes los efectos en el mundo real del uso de esas capacidades brindan las soluciones necesarias. [4]

MSA

Se puede entender a los microservicios como una extensión o evolución de SOA, permitiendo estructurar una aplicación basada en servicios como una colección de servicios de software muy pequeños acoplados libremente. La Arquitectura de Microservicios (MSA) puede ser vista como un nuevo paradigma de programación que se basa en la composición de pequeños servicios, cada uno procesando y comunicándose por medio de mecanismos ligeros, los microservicios son considerados micro por su contribución a la aplicación, mas no por su tamaño en líneas de código. [5]

Si realizamos una comparación con la arquitectura monolítica, en la que todos los procesos están relacionados y se ejecutan como un único servicio, por lo cual incorporar nuevas características se convertirá en una tarea compleja, dificultando la implementación de nuevas ideas. Las arquitecturas monolíticas aumentan el riesgo de la disponibilidad de la aplicación porque muchos procesos dependientes y estrechamente vinculados aumentan el impacto del error de un proceso. [6]

Por otra parte, una arquitectura de microservicios, como la de la Ilustración 2, nos permite crear componentes independientes que ejecutan cada proceso de la aplicación como un servicio. Los servicios al ser independientes se comunican a través de una interfaz definida mediante APIs ligeras, cada uno de estos servicios desempeña una sola función y debido a que se ejecutan de manera independiente, cada servicio se puede actualizar, escalar o implementar para satisfacer demandas específicas en la aplicación.

MSA tiene dos características principales:

- **Autonomía**: Cada servicio se puede desarrollar, implementar, operar y escalar sin afectar el funcionamiento de otros servicios. Su comunicación es a través de APIs ligeras.
- **Especialización**: Cada servicio está diseñado para un conjunto de capacidades y se enfoca en resolver un problema específico.

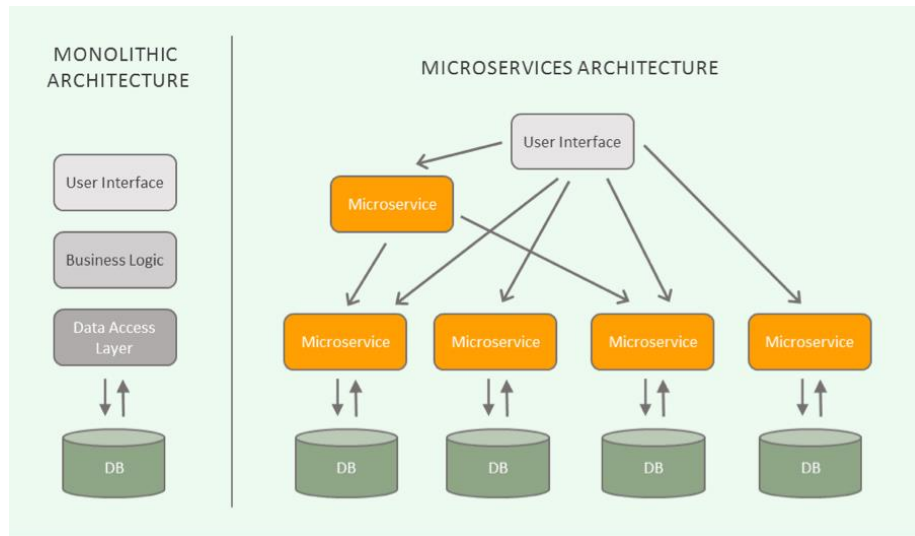


Ilustración 2 Comparación entre Arquitectura Monolítica y MSA [7]

Ventajas y Desventajas de los MSA y SOA

Tras un análisis exhaustivo se ha logrado capturar los beneficios y desventajas de los enfoques anteriormente estudiados y los cuales se mostrarán a continuación.

	Ventajas	Desventajas
SOA	<ul style="list-style-type: none"> • Aplicaciones débilmente acopladas. • Conectividad e interoperabilidad de aplicaciones. • Alineación de TI según las necesidades del negocio. • Arquitectura centrada en procesos. • Desarrollo paralelo e independiente. • Mejor escalabilidad y cambios evolutivos. • Costos reducidos de desarrollo e integración de aplicaciones • Mantenimiento sencillo. 	<ul style="list-style-type: none"> • Conexión débil, ya que el sistema distribuido tenderá a caerse en algún momento. • La integración de servicios puede ser compleja si no existe personal capacitado para trabajar en un entorno basado en SOA. • La integración de servicios en entornos heterogéneos puede convertirse en un problema. • Si las aplicaciones requieren continuamente funcionalidades adicionales, todo el sistema puede volverse inestable. • Requiere una gran inversión inicial en tecnología y desarrollo.

MSA	<ul style="list-style-type: none"> • Fomentan una organización de equipos pequeños e independientes que se apropian de los servicios, aumento de rendimiento de la organización. • Permiten la integración y la entrega continuas, facilitando probar nuevas ideas y revertirlas si algo no funciona. • Cada servicio escala de forma independiente, satisfaciendo la demanda de la aplicación que respalda. • Los equipos tienen la libertad de elegir la mejor herramienta para resolver sus problemas específicos. • Es reutilizable, pues la división del software en módulos pequeños y bien definidos les permite a los equipos usar funciones para diferentes propósitos. • La independencia del servicio aumenta la resistencia de una aplicación a los errores. 	<ul style="list-style-type: none"> • Presentan un alto consumo de memoria, tanto de CPU como RAM. • Con un gran número de microservicios es complejo gestionarlos. • Existencia de problemas de latencia o balanceo de carga. • En un ambiente distribuido, las pruebas se vuelven complicadas. • Las configuraciones iniciales y la creación de la arquitectura necesitarán de una alta inversión de tiempo. • Coste de implementación alto, debido a los costes de infraestructura y las pruebas distribuidas. [8]
-----	--	--

Caso de Estudio de SOA: Amazon

En los años recientes, ha existido una tendencia en la ingeniería de software sobre la computación en la nube, las plataformas en la nube se han popularizado por ser adoptadas como modelo por varias compañías como Amazon, Microsoft e IBM. A continuación, se presenta un caso de estudio de una empresa que emplea la arquitectura de microservicios para innovar en el mercado. [9]

En 2001, Amazon empleaba una arquitectura monolítica, cada uno de los niveles que lo componían tenía componentes, todos con alto acoplamiento y baja cohesión. Todo esto acompañado de un gran equipo de desarrolladores encargados de mantener el sitio web monolítico, cuando se agregaba una nueva funcionalidad o se realizaban correcciones tenían que asegurarse de que el cambio no rompería ninguna otra cosa en ese proyecto.

A comienzos de los 2000, Amazon incluso tenía un grupo de ingeniería cuyo único trabajo era tomar nuevas versiones de la aplicación y empujarla manualmente a través del entorno de producción de Amazon.

Las principales razones para adoptar una nueva arquitectura se basan en la frustración por parte de los ingenieros para mantener el sistema, además de verse limitados en la capacidad de innovar con nuevas funcionalidades. El cambio comenzó con el cambio a MSA, Amazon también implementó cambios en cómo operaba su organización. Cambiaron su único equipo de desarrollo de producto, por pequeños «equipos de dos pizzas», equipos tan pequeños que pudieran alimentarlos con solo dos pizzas, aunque en realidad, hay entre 6 y 8 desarrolladores por equipo. Cada uno de estos equipos recibió la propiedad total de uno o unos pocos microservicios. Así que definieron su propia hoja de ruta de características, diseñaron sus características según sus necesidades, las codificaron, probaron, implementaron y operaron.

Después de todos estos cambios, Amazon mejoró dramáticamente su ciclo de vida de desarrollo front-end. En la actualidad, los equipos de productos pueden tomar decisiones rápidamente y generar nuevas características para sus microservicios. Ahora la compañía realiza 50 millones de implementaciones al año, gracias a la arquitectura de microservicios y sus procesos de entrega continua.

Conclusiones

SOA y MSA ofrecen mejores oportunidades para la integración empresarial con costos bajos, fácil mantenimiento, mayor mantenibilidad y escalabilidad mejorada. Sin embargo, aunque SOA y MSA prometen grandes beneficios, ya que se basa en principios sólidos de servicios granulares, poco acoplados, basados en estándares, interoperables y reutilizables, también existen numerosos desafíos como el cambio de mentalidad, una gran inversión inicial, la falta de fiabilidad de los protocolos de Internet, estándares en evolución y la novedad del enfoque.

Referencia

- [1] T. Yoon y B.-K. Jeong, «Service Oriented Architecture (SOA) Implementation: Success Factors and Realized Benefits,» *International Journal of Information Systems in the Service Sector*, vol. 10, nº 2, p. 1, 2018.
- [2] M. Zaigham, «Service Oriented Architecture: Potential Benefits and Challenges,» *11th WSEAS International Conference on COMPUTERS*, pp. 497-502, 2007.
- [3] Wikipedia, «Enterprise SOA,» 2005. [En línea]. Available: https://upload.wikimedia.org/wikipedia/commons/thumb/d/d4/SOA_Elements.png/450px-SOA_Elements.png. [Último acceso: 22 Febrero 2022].
- [4] K. Laskey y K. Laskey, «Service oriented architecture,» *WIREs Computational Statistics*, vol. 1, pp. 101-106, 2009.
- [5] L. De Lauretis, «From Monolithic Architecture to Microservices Architecture,» *IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pp. 93-97, 2019.
- [6] Amazon Web Services, «¿Qué son los microservicios?,» Amazon , [En línea]. Available: <https://aws.amazon.com/es/microservices/>. [Último acceso: 22 Febrero 2022].
- [7] Auribox Training, «¿Qué son los Microservicios?,» Auribox Training, 30 Junio 2017. [En línea]. Available: <https://blog.auriboxtraining.com/desarrollo-web/microservicios/>. [Último acceso: 22 Febrero 2022].
- [8] Decide Soluciones, «Arquitectura de microservicios: qué es, ventajas y desventajas,» 3 Septiembre 2019. [En línea]. Available: <https://decidesoluciones.es/arquitectura-de-microservicios/>. [Último acceso: 22 Febrero 2022].
- [9] Apiumhub, «Los Microservicios,» Apiumhub, [En línea]. Available: <https://apiumhub.com/es/tech-blog-barcelona/los-microservicios/>. [Último acceso: 22 Febrero 2022].

