



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN SOFTWARE



APLICACIONES WEB AVANZADAS (ISWD813)

SOA VS Microservicios

Juan Pablo Tovar

PERIODO: 2021 B

Fecha:

Lunes 7 de marzo de 2021

SOA

La arquitectura orientada a servicios que se enfoca en la reutilización de elementos que se obtienen a partir de servicios, para que quede claro, un servicio es una unidad de software autónoma que se encarga de realizar una tarea en específico. Al tratarse de una unidad autónoma, cuenta con su propio código y por lo general desempeña actividades elementales que en su conjunto forma un complejo sistema empresarial.

Anteriormente era complicado lograr una integración entre una aplicación y servicios externos o de terceros, pues era necesaria una configuración punto a punto y definir protocolos de red, puertos, direcciones Ip, etc. Este proceso se debía repetir desde 0 para cada uno de los servicios que se van a utilizar, el problema de este tipo de soluciones era que todo formaba parte de una sola aplicación y por lo tanto el proceso de corrección de errores y mantenimiento afectaba de forma directa a todos los servicios involucrados.

SOA utiliza protocolos de red para exponer los datos y gestionar la información a través del manejo de peticiones HTTP, lo cual hace los servicios mantengan un bajo acoplamiento con la plataforma principal y cuando se desee realizar algún cambio solo se vea afectado el propio servicio.

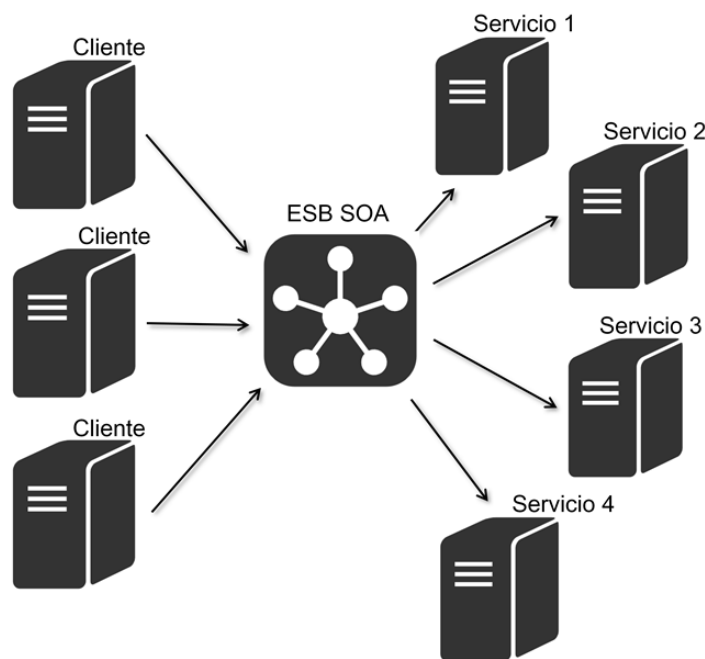


Ilustración 1. Arquitectura orientada a servicios

Beneficios de SOA

- Comercialización más rápida y mayor flexibilidad
- Uso de la infraestructura heredada en los mercados nuevos:
- Reducción de los costos gracias a una mayor agilidad y un desarrollo más eficiente
- Mantenimiento sencillo
- Escalabilidad
- Mayor confiabilidad
- Gran disponibilidad

Microservicios

Los microservicios se tratan de un tipo de diseño de software que se encuentra compuesta por varios servicios independientes que se comunican a través de una API definida, existe un equipo detrás del desarrollo de cada uno de estos servicios. Algunas de las ventajas que ofrece una arquitectura de microservicios es la facilidad para escalar y la velocidad de desarrollo, haciendo más eficiente el uso de recursos dentro de una compañía.

Realizando una breve comparación de una arquitectura monolito convencional con una de microservicios se puede considerar que la primera es como un gran servicio haciendo que tareas como la actualización y el mantenimiento afecten directamente a todo el sistema, haciendo que el nivel de complejidad del código aumente y dificulte el proceso de desarrollo.

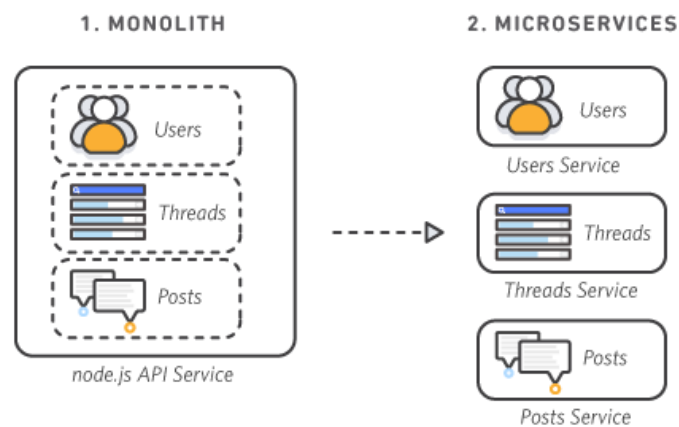


Ilustración 2. Arquitectura de microservicios

Por otro lado, una arquitectura de microservicios se encarga de realizar una sola función distribuyendo la carga computacional, es decir, cada microservicio se encarga de un proceso de la empresa. Todos estos microservicios se conectan con la aplicación principal a través de una API que se definió anteriormente.

Beneficios de microservicios

- Escalabilidad
- Implementación sencilla
- El bajo costo de los errores permite experimentar, facilita la actualización del código y acelera el tiempo de comercialización de las nuevas características
- Código reutilizable
- Agilidad en cambios
- Aplicación independiente
- Menor riesgo

SOA vs Microservicios

En base a las definiciones presentadas anteriormente es evidente que ambas arquitecturas son bastante similares, pero es importante tener claro que no son iguales, si bien comparten una gran cantidad de características como la autonomía de sus módulos las principales diferencias se muestran a continuación:

- **Compartición de componentes:** si MSA minimiza el uso compartido de componentes a través de un contexto limitado, SOA le saca todo el partido, algo que, por otra parte, aumenta las latencias, haciendo que los sistemas basados en este tipo de arquitectura sean más lentos.
- **Granularidad de servicio:** SOA ofrece servicios más versátiles en cuanto a su funcionalidad empresarial, mientras que microservicios se compone de servicios de propósito único y muy especializados.
- **Coordinación:** los Microservicios generalmente carecen de coordinación entre sí, o si la presentan es mínima. Por el contrario, en SOA, es necesario coordinar con varios grupos para atender a las solicitudes de negocio.
- **Middleware:** esta opción de mensajería es la empleada por SOA, que ofrece en base a ella una serie de capacidades adicionales que no se encuentran en MSA, incluyendo mediación y enrutamiento, mejora de mensajes, mensajes y transformación de protocolos. Por su parte, los microservicios trabajan con una capa de API creada entre los servicios y los consumidores de servicios.
- **Interoperabilidad heterogénea:** SOA promueve la propagación de múltiples protocolos heterogéneos a través de su componente middleware de mensajería, por eso, esta opción debe tenerse en cuenta en los casos en que el objetivo sea lograr la integración de varios sistemas utilizando diferentes protocolos en un entorno heterogéneo. Sin embargo, será preferible recurrir a microservicios cuando todos los servicios puedan quedar expuestos y se deba usar el mismo protocolo de acceso remoto, ya que MSA intenta simplificar el patrón de arquitectura al reducir el número de opciones de integración.

Spotify: Arquitectura de microservicios

Spotify es una de las empresas más grandes dedicadas al Streaming de música y cuenta con su aplicación en distintas plataformas como dispositivos móviles, página web, aplicación de escritorio, etc. Esto ha hecho que se tenga que enfrentar a una gran cantidad de obstáculos entre los cuales se encuentran:

- Escalabilidad para millones de usuarios
- Soporte para usuarios en múltiples plataformas
- Manejar reglas de negocio complicadas
- Competir en un mercado que se encuentra en constante movimiento
 - Reaccionar rápidamente
 - Innovar

Para lograr crear un producto que supere todas estas barreras Spotify ha optado por una arquitectura de microservicios, esto le permite poder manejar dichos requerimientos mientras mantiene una velocidad de reacción y gestión de cambios alta.

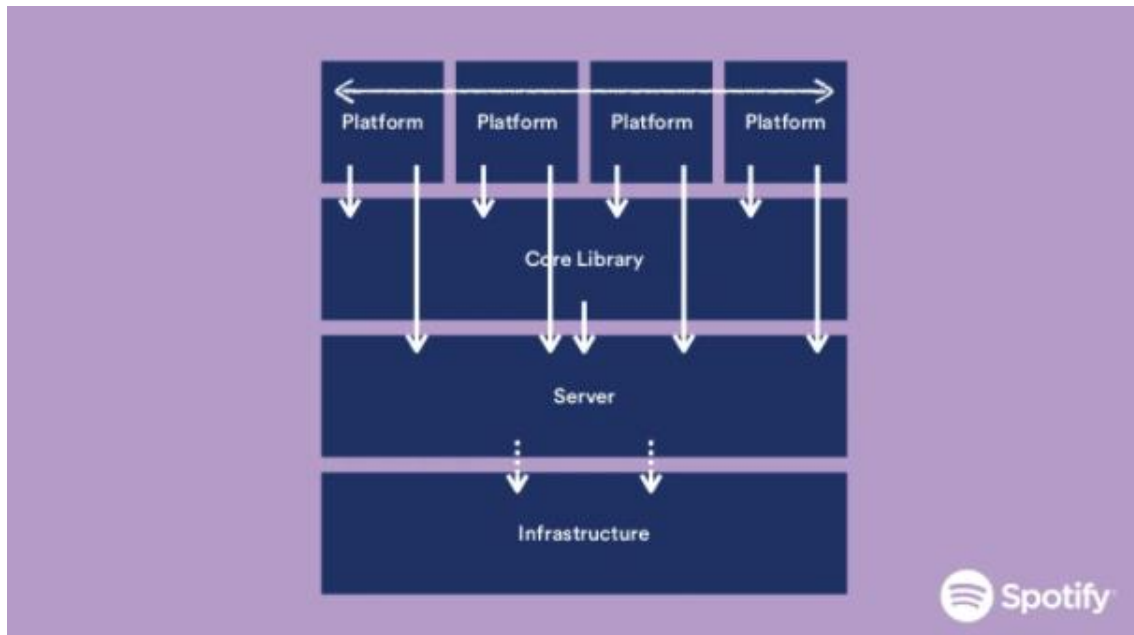


Ilustración 3. Arquitectura de Spotify

Spotify llama a esta solución Equipos autónomos full stack, y se refiere a autónomos como la capacidad de actuar libremente y de manera independiente, para lo cual se ha definido una arquitectura en donde cada plataforma interactúa con un núcleo de librerías y un servidor común, sin embargo, los equipos responsables del desarrollo y mantenimiento de cada uno de los productos de las plataformas trabajan de manera independiente.

Una de las dificultades que presenta esta arquitectura es la sincronización vertical que existe entre la experiencia de usuario (Cualquiera de las plataformas), la implementación del núcleo de librerías, el servidor y la infraestructura, pues existe una dependencia entre estos cuatro elementos y para esto se ha definido un equipo de características que tiene la función de enfocarse únicamente en la implementación de nuevas características en la capa de librerías y en el servidor, para que cada una de las plataformas pueda utilizar estas características.

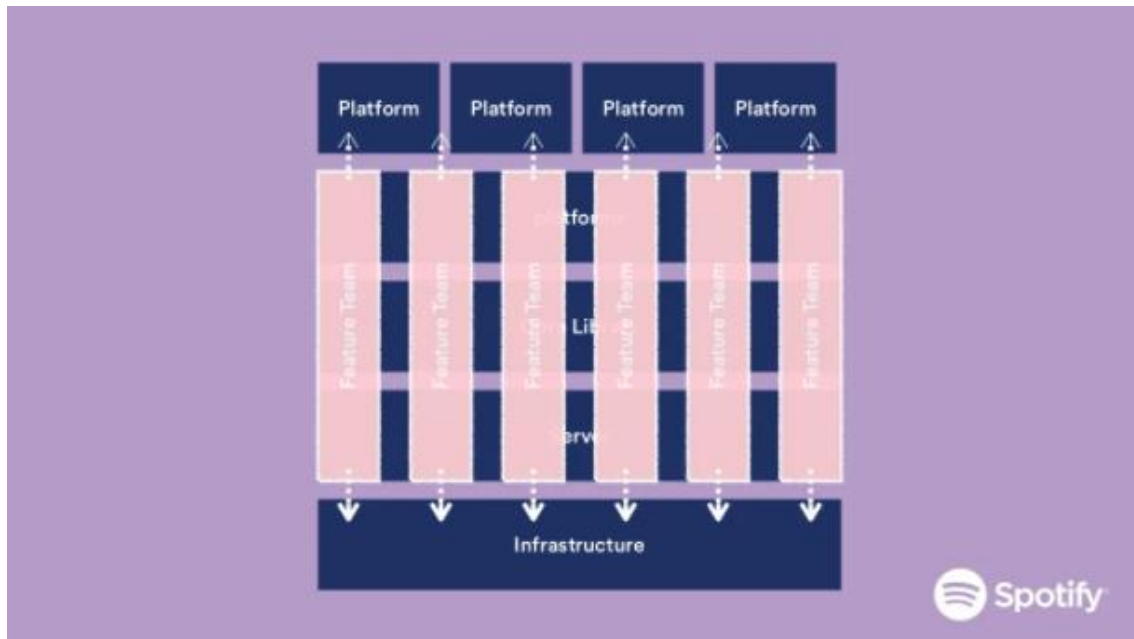


Ilustración 4. Equipos de características

Con la utilización de esta arquitectura Spotify cuenta actualmente con más de 90 equipos de desarrollo, formando un total de más de 600 empleados, 5 departamentos de desarrollo y un solo producto.

Como se mencionó anteriormente el tener un servidor monolito en el cual se encuentren varias bases de datos conectadas conlleva un gran número de desventajas y uno de los requerimientos vitales para este producto era el rendimiento por lo que contar con varios servidores monolitos con un balanceador de carga no es la solución pues solo multiplicaría el número de problemas, mientras que la arquitectura de microservicios mostraba alguna de sus cualidades:

- Fácil de escalar basado en cuellos de botella.
- Fácil de probar.
- Fácil despliegue.
- Fácil de monitorear.
- Puede ser versionado de manera independiente.
- Son menos susceptibles a fallas de software de gran escala.

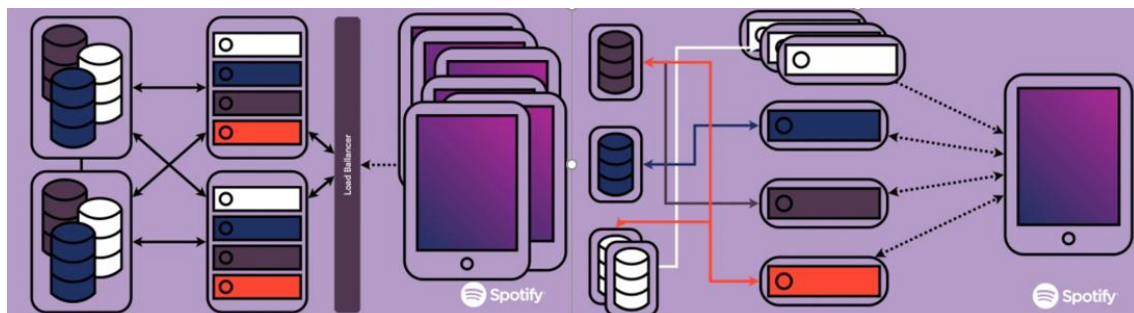


Ilustración 5. Arquitectura monolito vs Microservicios

Con la utilización de esta arquitectura Spotify el día de hoy cuenta con más de 810 servicios activos, de los cuales 10 servicios se encuentran a cargo de un equipo y aproximadamente 1.7 sistemas por persona con acceso a servidores de producción.

Referencias

¿Qué es SOA y cuál es su diferencia con los Microservicios? (4 de agosto de 2017). Obtenido de Powerdata: <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-es-soa-y-cual-es-su-diferencia-con-los-microservicios>

Amazon. (s.f.). Obtenido de ¿Qué son los microservicios?: <https://aws.amazon.com/es/microservices/>

Goldsmith, K. (13 de marzo de 2016). *Microservices at Spotify*. Obtenido de SlideShare: <https://www.slideshare.net/kevingoldsmith/microservices-at-spotify>

Hiberus. (16 de junio de 2022). Obtenido de ¿Cuáles son los beneficios de una arquitectura microservicios?: <https://www.hiberus.com/crecemos-contigo/cuales-son-los-beneficios-de-una-arquitectura-microservicios/>

RedHat. (27 de julio de 2020). Obtenido de ¿Qué es la arquitectura orientada a los servicios (SOA)?: <https://www.redhat.com/es/topics/cloud-native-apps/what-is-service-oriented-architecture>

SPOTIFY APOLLO OPERATIONS MANUAL. (s.f.). Obtenido de Github: <https://spotify.github.io/apollo/>