



Nombre: Lesly Tipanluisa

Tema: Ensayo sobre Microservicios vs SOA – Caso de estudio Netflix

En los últimos años, la tecnología ha avanzado de manera considerable por lo que uno de los temas más mencionados es el referente a los microservicios, el cual, ha causado debates entre los equipos de los departamentos de IT acerca de cuán conveniente es utilizar esta arquitectura o la arquitectura SOA. Por lo tanto, es importante explicar en qué consisten cada una de estas arquitecturas, definir sus ventajas, desventajas con el objetivo de tener mayor claridad al momento de tomar una decisión sobre qué arquitectura usar de acuerdo con el modelo de negocio que tenga una organización.

En primer lugar, SOA (Service Oriented Architecture) o arquitectura orientada a servicios especifican una manera de hacer que los componentes de software sean reutilizables e interoperables a través de interfaces de servicio[1]. Cada servicio SOA posee el código y los datos que son necesarios para ejecutar una actividad empresarial, por ejemplo: comprobar el crédito de un cliente, calcular el pago mensual de un préstamos, etc. Aquí se tiene una reducción de las dependencias entre aplicaciones ya que se puede llamar a las interfaces de servicio con poco o ningún conocimiento de cómo está implementado el servicio.

Por otro lado, los microservicios estructuran una aplicación como una colección de servicios, permitiendo de esta forma realizar una entrega rápida, frecuente y confiable de aplicaciones grandes y complejas [2]. Cada uno de los servicios trabajan al mismo tiempo, se ejecutan en su propio proceso y se mantienen en constante comunicación entre sí, por lo que se puede determinar que cada servicio se empaqueta y se despliega de forma independiente, al estar diseñados como servicios altamente autónomos debido a que no dependen de una comunicación sincronizada y centralizada. Entre las funciones que puede cumplir un microservicio está el funcionamiento de un carrito de compras, un motor de búsqueda o administrar usuarios de una web.

Entre las ventajas que ofrecen los microservicios es que se puede realizar la implementación de manera simple, ya que se tienen partes literales lo que permite no afectar a otros servicios. Además, debido a que las funciones están aisladas, el código es



más fácil de entender. Uno de los principios que pertenece a esta arquitectura es la Autonomía, por ejemplo, permite utilizar cualquier lenguaje de programación, Java, Python, etc. siempre que se tenga un criterio de selección coherente, es decir la elección de los lenguajes debe ser de acuerdo con los beneficios que se obtendrán al momento de usarlos.

Luego, se tiene que SOA es una manera de diseñar e implementar los procesos de negocios, con el propósito de obtener gran flexibilidad y evolución. Una de las principales ventajas que presenta SOA es que resuelve los problemas de conectividad, además, de permitir una gran independencia de las plataformas. Por lo tanto, reduce el nivel de acoplamiento, está dirigida para la implementación de aquellos sistemas que requieren de gran conectividad e integración de dispositivos remotos. Ofrece la facilidad de evolucionar a nuevos modelos de negocios e integración de tecnologías.

Tanto en los microservicios como en SOA se va a necesitar más personas debido a que el nivel de conocimiento que se necesita es mayor. En los microservicios el volumen de actividad es más alto ya que todo comienza a crecer lo que genera la necesidad de más personal, más recursos y tanto el diseño como el testing se vuelven más complejos. En SOA la implementación es más costosa porque se necesita más documentación, personal, herramientas para realizar debugs y sobre todo necesita de una supervisión continua, tanto de los logs, al momento de enviar alertas o correos cuando se genere alguna eventualidad resulta en mucho trabajo.

Una vez que se han presentado las ventajas y desventajas que tienen los microservicios y SOA, se procederá a realizar un análisis del caso de estudio: “*Design and analysis of Netflix microservice architecture*”[3].



Netflix ha sido el mejor servicio de transmisión de video por suscripción en línea del mundo, su tráfico representa más del 15% de la capacidad global de ancho de banda de Internet. En 2019, Netflix tenía aproximadamente 167 millones de suscriptores, en promedio cinco millones de suscriptores cada trimestre, cubriendo de esta forma más de 200 países alrededor del mundo. También se conoce que los usuarios de Netflix pasan más de 165 millones de horas al día en más de 4000 películas y 47000 series de televisión. Con esta información, las estadísticas muestran que el sistema de transmisión de video diseñado por el equipo técnico de Netflix tiene alta disponibilidad y escalabilidad siendo capaz de proporcionar servicios para usuarios de todo el mundo. Un dato importante es que el equipo técnico de Netflix se demoró más de ocho años en construir un sistema de TI tan potente en la actualidad.

La transformación de la infraestructura de Netflix comenzó en agosto de 2008, cuando el equipo técnico se dio cuenta que necesitaba una infraestructura más confiable sin un solo punto de falla. Por lo tanto, la gerencia del equipo técnico tomó la decisión de migrar la infraestructura de TI de su propio IDC a la nube pública y transformarla en una arquitectura de microservicio, reemplazando los programas individuales con componentes de software manejables más pequeños.

Netflix migró su infraestructura a la nube de AWS porque proporciona bases de datos altamente confiables, almacenamiento a gran escala y muchos centros de datos en todo el mundo, lo que ha eliminado el trabajo pesado y repetitivo de crear su propio centro de datos, enfocándose así en el propósito del negocio, que es brindar una experiencia de transmisión de video de alta calidad. Netflix eligió una arquitectura de microservicios para su sistema basado en la nube con el fin de admitir cargas ligeras y pesadas que se ejecutan en la misma infraestructura subyacente.

A continuación, se realiza un análisis de la disponibilidad y escalabilidad de esta arquitectura, por lo que se debe estudiar cada componente para comprender su rendimiento bajo diferentes cargas. En el lado del cliente, Netflix aún puede controlar el rendimiento del dispositivo a través de su propio SDK, el funcionamiento del componente de arquitectura de cliente es el siguiente: la aplicación cliente divide su conexión con el backend en dos tipos, los que se usan para la detección y reproducción de contenido. Al transmitir el video, si la conexión de red está sobrecargada o se produce un error, la



aplicación cliente reducirá de manera inteligente la calidad del video. El componente de servicio de puerta de enlace API se comunica con el balanceador de carga de AWS para resolver todas las solicitudes del cliente. La API de la aplicación actúa como la capa de proceso empresarial de los microservicios de Netflix, esta API proporciona la lógica para ensamblar llamadas a los microservicios subyacentes en el orden deseado, con datos adicionales de otros almacenamientos de datos para construir las respuestas apropiadas. Las componentes de la API de la aplicación corresponden a las funciones comerciales principales de Netflix. Debido a que se necesita ser escalable y altamente disponible bajo altas solicitudes, la api de aplicaciones se divide en tres categorías: API de registro para solicitudes de no miembros, API de descubrimiento para solicitudes de búsqueda y descubrimiento, la API de reproducción para transmisión de video y visualizaciones de solicitudes de licencia.

Debido a que la API de la aplicación debe procesar una gran cantidad de solicitudes y generar respuestas adecuadas, el procesamiento interno debe ejecutarse en paralelo, por lo que realizaron una combinación de ejecución síncrona y E/S asíncrona.

Las ventajas que han proporcionado los microservicios a Netflix se detallan a continuación, por ejemplo, la API de reproducción no siempre esperará la ejecución del microservicio, ya que usa el comando hystrix para controlar el tiempo de espera antes de obtener resultados. Lo que permite tener un control aceptable con respecto al retraso y de esta forma evitar fallas en cascada que afecten a más servicios. Si el servidor OCA actualmente seleccionado tiene una falla o sobrecarga en la red, el cliente cambiará inmediatamente a otro servidor OCA con la conexión de red más confiable. Por otro lado, la escalabilidad es mejor ya que la API permite la ejecución paralela de tareas en el bucle de eventos de red y el bucle de eventos salientes asíncronos.



Finalmente, cabe mencionar que una diferencia fundamental entre SOA y los microservicios, es que los microservicios no tienen un bus de servicios. En SOA todos se comunican a través del bus de servicios, pero está dividido en capas, en cambio, en los microservicios todo es totalmente independiente por lo que cada uno puede colaborar con otro servicio.

Si una organización ha creado una aplicación con la arquitectura de microservicios, se recomienda realizar una evaluación de los microservicios, con el objetivo de analizar lo construido e identificar lo que se necesita mejorar, de manera que se reduce el riesgo arquitectónico y organizacional, maximizando así los beneficios de esta arquitectura.

Bibliografía

- [1] I. C. Education, “What is SOA (Service-Oriented Architecture)? | IBM,” 2021. [Online]. Available: <https://www.ibm.com/cloud/learn/soa>. [Accessed: 20-Feb-2022].
- [2] C. Richardson, “What are microservices?,” 2021. [Online]. Available: <https://microservices.io/>. [Accessed: 20-Feb-2022].
- [3] C. D. Nguyen, “Design and analysis of Netflix microservice architecture,” 2021. [Online]. Available: <https://developpaper.com/design-and-analysis-of-netflix-microservice-architecture/>. [Accessed: 20-Feb-2022].