

Universidad Autónoma de Chiapas | Campus 01 | Facultad de contaduría y administración.

Docente: Dr. Luis Gutiérrez Alfaro.

Materia: Taller de desarrollo 4.

Nombre del alumno (s):

- Alegría de la Cruz Brayan Fermín | A210519.

Semestre: 6° | Grupo: "M".

Tema: Arquitectura Orientada a Servicios.

Unidad: 1 | Taller de desarrollo IV.

Número de actividad: 1.2 | Arquitectura orientada a Servicios.

Tuxtla Gutiérrez, Chiapas; a 25 de enero de 2024.

## Contenido

INTRODUCCIÓN.....	3
ARQUITECTURA ORIENTADA A SERVICIOS (SOA). ....	4
¿Qué es SOA? .....	4
Características. ....	4
Beneficios SOA. ....	5
ARQUITECTURA DE MICROSERVICIOS. ....	5
¿Qué es la arquitectura de microservicios?.....	5
Características. ....	6
Beneficios de arquitectura de microservicios. ....	6
Técnicas de implementación de microservicios. ....	7
Balanceo de carga en microservicios. ....	8
Despliegue. ....	8
Arquitectura monolítica vs de microservicios.....	8
Ventajas. ....	8
Desventajas.....	9
Ventajas.....	9
Desventajas.....	9
¿Cuándo usar uno u otro? .....	10
Buenas prácticas para diseñar arquitecturas de microservicio.....	10
REFERENCIAS. ....	11

## INTRODUCCIÓN.

El desarrollo de software con el paso del tiempo se ha convertido en una disciplina que requiere una gestión y productividad altamente demandante. De forma ambigua, durante muchos años los sistemas o aplicaciones han utilizado metodologías bastante simples las cuáles únicamente permiten enfatizar el término “funcionalidad” sin embargo, se deslindan de términos como “interoperabilidad”, “independencia”, entre otros.

En la actualidad se han creado modelos arquitectónicos de software, los cuales, permiten que el desarrollo de un proyecto se lleve a cabo de forma independiente entre componentes; como si de un rompecabezas estuviésemos hablando.

Desde principios de los años 2000's se introdujo un nuevo paradigma que permitía dicha modularidad en el desarrollo de software, sin embargo, no fue aproximadamente hasta una década más tarda en que este tomó terreno y se volvió una tecnología que innovara la forma de desarrollar software.

La arquitectura de microservicios, abrió un nuevo panorama a la hora de desarrollar aplicaciones, la cual, permite la independencia de mecanismos funcionales, autónomos, los cuales, no requieren estar forzosamente ligados a los demás, basta con únicamente mantenerlos bajo comunicación para que trabajen de forma correcta.

Hoy en día la arquitectura de microservicios ha dejado de lado la arquitectura monolítica, la cual, representa la forma común y tradicional de desarrollar una aplicación. Teniendo un todo, dependiente de si mismo y que complica demasiado su escalabilidad; dado que, para escalar, se requiere modificar todo el producto para poder realizar una implementación o innovación.

## ARQUITECTURA ORIENTADA A SERVICIOS (SOA).

### ¿Qué es SOA?

SOA es un paradigma arquitectónico que se centra en la creación de servicios independientes y reutilizables, promoviendo conceptos tales como: flexibilidad, interoperabilidad, capacidad de respuesta a cambios, entre otros. Promoviendo la creación de servicios independientes y autónomos, que puedan ser reutilizados y combinados para satisfacer los requisitos empresariales de un rubro.

### Características.

- **Servicios independientes:** En SOA los servicios son unidades lógicas que operan de forma independiente. Cada servicio es autónomo y puede ser desarrollado, desplegado y actualizado de forma individual.
- **Reusabilidad:** Dado que es una arquitectura que fomenta la modularidad, SOA abre paso a la reutilización de servicios, mejorando la eficiencia y evitando la duplicación de estos mismos.
- **Interoperabilidad:** SOA permite la interoperabilidad, esto, dado que los servicios están diseñados para comunicarse de manera efectiva haciendo uso de estándares abiertos y protocolos bien definidos.
- **Granularidad:** En SOA, los servicios están diseñados para tener el alcance y tamaño adecuado.
- **Seguridad:** SOA implementa medidas de seguridad para proteger la integridad de los datos que se transmiten entre servicios, incluyendo conceptos como: autenticación, autorización y cifrado.

- Flexibilidad y Agilidad: SOA permite a las organizaciones adaptarse fácilmente a cambios, dado que la arquitectura modular facilita la modificación, implementación o eliminación de servicios.

#### Beneficios SOA.

- Gestión del ciclo de vida de servicios: Dado que SOA aborda este punto; desde el diseño hasta el despliegue de los servicios, facilita el mantenimiento y la evolución continua de estos mismos.
- Seguridad y Gobernanza: SOA incorpora protocolos de seguridad para proteger los datos que se transmiten entre servicios; procurando su integridad y confidencialidad.
- Mejor colaboración empresarial: SOA promueve la colaboración entre áreas de una organización, dado que proporciona un enfoque común y estandarizado para construir sistemas; facilitando la integración y cooperación.
- Reducción de carga operativa de TI: Se disminuye el coste de mantenimiento, administración y esfuerzo de desarrollo gracias a protocolos de normalización y estandarización de los servicios.

## ARQUITECTURA DE MICROSERVICIOS.

### ¿Qué es la arquitectura de microservicios?

La arquitectura de microservicios, es un enfoque arquitectónico el cual aboga por dividir una aplicación en pequeños servicios independientes y autosuficientes; con su propia lógica y base de datos, comunicándose entre sí con ayuda de una API.

La arquitectura de microservicios, además, se considera un reemplazo moderno de la arquitectura monolítica, siendo esta más flexible e independiente.

#### Características.

- Descomposición en servicios: La aplicación se descompone en servicios independientes y autosuficientes, cada uno centrado en una funcionalidad que satisfaga la lógica del negocio.
- Escalabilidad: Los microservicios pueden escalarse de forma independiente según el tráfico de demanda, esto permite asignar recursos a un servicio que requiera mayor capacidad sin afectar a otros.
- Tolerancia a fallos: Dicho enfoque arquitectónico fomenta la construcción de un sistema resistente; la falla de un microservicio no deberá afectar la funcionalidad de los demás.
- Gobernanza descentralizada: Permite que los equipos de desarrollo sean más ágiles y respondan rápidamente a cambios.
- Comunicación a través de protocolos ligeros: Dado que los servicios se comunican por medio de protocolos ligeros (HTTP/REST), la comunicación puede ser asíncrona y estandarizada.

#### Beneficios de arquitectura de microservicios.

- Escalabilidad y flexibilidad: Permite escalar servicios de manera independiente según la demanda.
- Reutilización y mantenimiento: Facilita el desarrollo, despliegue y mantenimiento de cada servicio. Permitiendo así, a los equipos trabajar de forma autónoma sin afectar la aplicación en su totalidad.

- Autonomía: Cada servicio es autónomo, por tanto, esto favorece la descentralización y permite a los equipos tomar decisiones de forma eficiente y rápida.
- Diversidad tecnológica: Permite la elección e implementación de tecnologías para cada microservicio, facilitando su adaptación.
- Innovación: Permite la introducción de nuevas funcionalidades o innovaciones a la aplicación. Por tanto, se puede trabajar en la implementación de nuevas funcionalidades en un microservicio sin interferir con los demás.

#### Técnicas de implementación de microservicios.

- Contenedores: Cada microservicio se empaqueta junto con sus dependencias en un contenedor, lo cual, garantiza su consistencia en entornos de prueba y desarrollo.
- Orquestación de contenedores: Con ayuda de herramientas de automatización, tales como Ansible, Chef o Puppet; se evita la producción de errores.
- API Gateway: Se emplea para proporcionar un punto de entrada a los clientes y gestionar aspectos de seguridad y transmisión de datos.
- Implementación gradual: Implica en migrar de una arquitectura monolítica a una de microservicios.
- Desarrollo basado en eventos: Implica el uso de sistemas de colas o mensajería en eventos para facilitar la comunicación asíncrona entre microservicios.

### Balanceo de carga en microservicios.

El concepto de balanceo de carga, se refiere a la distribución igualitaria del tráfico entre instancias de un servicio para garantizar el uso eficiente de recursos y mejorar la disponibilidad y respuesta del sistema.

### Despliegue.

Bajo el contexto de microservicios, el despliegue refiere al proceso de poner a trabajar y hacer visible una versión o actualización de un servicio de una aplicación. Dicho concepto, implica llevar a cabo cambios de infraestructura, implementación de nuevas versiones de los microservicios y asegurar cambios efectivos, disponibles y escalables.

### Arquitectura monolítica vs de microservicios.

La arquitectura monolítica permite la construcción de un proyecto como una unidad única e indivisible. Algunas de sus características son:

- Comprende una interfaz del lado del cliente
- Una aplicación del lado del servidor
- Una base de datos

En ella, todas las funciones se realizan y se sirven en un mismo lugar. Por tanto, tienen un código muy grande y carecen de modularidad. Se realizan cambios en todo el stack a la vez.

### Ventajas.

- Fácil de testear y debuguear.
- Fácil de desplegar: únicamente necesita un directorio o servidor.
- Fácil de trabajar: dado que es la arquitectura tradicional.



### Desventajas.

- Código enorme "todo en un lugar".
- Carece de acoplamiento al estar todo en un mismo lugar; afectando a otras por sólo una.
- Escalabilidad costosa: al querer escalar un módulo, se requiere escalar todo lo demás.
- Dificil incorporación de nuevas tecnologías: empieza el proyecto en una tecnología, surgiendo otras, imposibles de agregar; terminando en la reescritura de todo el proyecto.

La arquitectura de microservicios, Divide la aplicación en unidades independientes más pequeñas; con su propia lógica y base de datos. Comunicándose entre sí con ayuda de una API.

### Ventajas.

- Fácil actualización de forma individual, un error en uno, no afecta a otros. Los cambios son fáciles de realizar.
- Fácil entendimiento de la app, dado que todo está dividido en microservicios.
- Escalabilidad: cada microservicio permite ser escalado de forma independiente ahorrando tiempo y dinero.
- Múltiples tecnologías: permite que las empresas posean la filosofía de "choose your own language" permite que los desarrolladores programen en el lenguaje que les sienta más cómodo.

### Desventajas.

- Complejidad: conlleva un sistema distribuido, diseñado meticulosamente; se deberá poseer una infraestructura desarrollada y analizada de forma compleja.

- Complejo de testeo: el testeo unitario es fácil, pero el test de integración es complicado dado que se debe coordinar múltiples microservicios.

### ¿Cuándo usar uno u otro?

- Poseyéndose: Una App sencilla y rápida; arquitectura monolítica.
- Poseyéndose: Una App grande y escalable; arquitectura de microservicios.
- Se requiere buen hardware y conocimiento, además un equipo de desarrollo grande.

### Buenas prácticas para diseñar arquitecturas de microservicio.

- Aplicación con el cliente: El término aplicación con el cliente refiere a la interfaz de usuario (UI); siendo ésta aquella con la cual, el usuario final interactúa directamente. De igual forma, el término enfatiza el cómo la interfaz de usuario interactúa con los servicios en el backend para obtener datos y obtener funcionalidad.
- Servicios: Bajo el contexto de la arquitectura de microservicios, este término refiere a las unidades de desarrollo que encapsulan la funcionalidad de una aplicación. Siendo estos componentes autónomos que se diseñan para cumplir una tarea específica.
- Directorio de servicios: Refiere a un componente que proporciona información sobre la ubicación o características de los diferentes servicios disponibles.
- Bus de servicio: Refiere al patrón arquitectónico que facilita la comunicación entre diferentes microservicios en una arquitectura distribuida.

## REFERENCIAS.

IBM. (s.f). ¿Qué es la arquitectura orientada a servicios (SOA)? <https://www.ibm.com/mx-es/topics/soa>

BeServices. (diciembre, 2021). ¿Qué es la arquitectura orientada a servicios (SOA)? ¿Cuáles son sus beneficios en la empresa? <https://blog.beservices.es/blog/que-es-la-arquitectura-orientada-servicios-soa-cuales-son-los-beneficios-en-la-empresa#:~:text=Un%20enfoco%20SOA%20permite%20crear,Incrementa%20la%20agilidad%20organizacional>.

AWS Amazon. (s.f). ¿Qué es la arquitectura orientada a servicios (SOA)? <https://aws.amazon.com/es/what-is/service-oriented-architecture/>

Intel. (junio, 2021). Microservicios y arquitectura de microservicios. <https://www.intel.la/content/www/xl/es/cloud-computing/microservices.html>

Atlassian. (s.f). Arquitectura de microservicios. <https://www.atlassian.com/es/microservices/microservices-architecture>

RootStack. (diciembre 23, 2022). Tecnologías con las que se pueden implementar los microservicios. <https://rootstack.com/es/blog/tecnologias-con-las-que-se-pueden-implementar-los-microservicios>

AppMaster. (agosto 03, 2023). Equilibrio de carga en microservicios con NGINX: guía detallada. <https://appmaster.io/es/blog/equilibrio-de-carga-microservicios-nginx>

Aws Amazon. (s.f). ¿Cuál es la diferencia entre la arquitectura monolítica y la de microservicios? <https://aws.amazon.com/es/compare/the-difference-between-monolithic-and-microservices-architecture/#:~:text=Modificar%20la%20arquitectura%20monol%C3%ADtica%20es,peque%C3%B1os%20componentes%20o%20servicios%20independientes>.