



UNIVERSIDAD AUTÓNOMA DE CHIAPAS  
FACULTAD DE CONTADURÍA Y  
ADMINISTRACIÓN, CAMPUS I  
LICENCIATURA EN INGENIERÍA Y  
DESARROLLO DE TECNOLOGÍAS DE  
SOFTWARE



# ARQUITECTURA ORIENTADA A SERVICIOS CONCEPTOS

COMPILADORES

6° M A220001-GEORGINA ANDREA  
MÉNDEZ HERNÁNDEZ

DOCENTE  
DR.

GUTIERREZ ALFARO LUIS

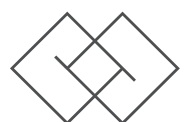
TUXTLA GUTIÉRREZ.

05 DE OCTUBRE DE 2023.

# ÍNDICE

---

➤	RESUMEN	01
➤	1.1 ARQUITECTURA DE MICROSERVICIOS	01
➤	1.2 CARACTERÍSTICAS	01
➤	1.3 BENEFICIOS	01
➤	1.4 TÉCNICAS DE INTEGRACIÓN	01
➤	2. BALANCEO DE CARGA	01
➤	2.1 ARQUITECTURA MONOLÍTICA VS NO MONOLÍTICA	01
➤	2.2 BUENAS PRÁCTICAS PARA DISEÑAR	02
➤	CONCLUSIONES	02



# RESUMEN

Esta investigación explora en profundidad la Arquitectura Basada en Microservicios y la Arquitectura Orientada a Servicios (SOA), destacando características distintivas, beneficios, técnicas de integración, despliegue, y comparando la arquitectura monolítica con la arquitectura de microservicios. Además, se presentan buenas prácticas para diseñar arquitecturas de microservicios, abordando aspectos como la aplicación del cliente, servicios, directorio de servicio y bus de servicio.

## 1.1 ARQUITECTURA DE MICROSERVICIOS

La Arquitectura de Microservicios es un enfoque para desarrollar aplicaciones como un conjunto de servicios independientes y autónomos, cada uno enfocado en una función específica. A diferencia de las arquitecturas monolíticas, los microservicios permiten la escalabilidad y la implementación independiente de cada servicio.

## 1.2 CARACTERÍSTICAS

- **Desacoplamiento:** Los microservicios son unidades independientes que pueden desarrollarse y desplegarse por separado.
- **Escalabilidad:** Los servicios pueden escalarse individualmente según las necesidades.
- **Resiliencia:** La arquitectura permite la tolerancia a fallos y la recuperación rápida.

## 1.3 BENEFICIOS

- **Desarrollo Ágil:** Facilita la entrega continua y rápida iteración en el desarrollo.
- **Escalabilidad Eficiente:** Los servicios se pueden escalar según la carga de trabajo específica.
- **Mantenimiento Simplificado:** Actualizaciones y correcciones se pueden realizar en servicios individuales.

.

## 1.4 TÉCNICAS DE INTEGRACIÓN

- **API REST:** Comunicación a través de interfaces RESTful.
- **Mensajería Asíncrona:** Uso de colas de mensajes para la comunicación entre servicios.

## 2. BALANCEO DE CARGA

- **Round Robin:** Distribución equitativa de las solicitudes entre los servicios.
- **Balanceo Basado en la Carga:** Asignación de solicitudes en función de la carga actual de los servicios.

## 2.1 ARQUITECTURA MONOLÍTICA VS NO MONOLÍTICA

- **Monolítica:** Una aplicación única y grande que contiene todas las funcionalidades.
- **Microservicios:** División en servicios independientes, cada uno con su propia funcionalidad.

## 2.2 BUENAS PRÁCTICAS PARA DISEÑAR

- **Aplicación con el Cliente:** Utilizar interfaces limpias y bien documentadas.
- **Servicios:** Definir límites de contexto claro y funcionalidades específicas.
- **Directorio de Servicio:** Mantener un registro centralizado de servicios para facilitar la descubribilidad.
- **Bus de Servicio:** Implementar un sistema de mensajes eficiente para la comunicación entre servicios.

## CONCLUSIONES

La elección entre arquitecturas monolíticas y de microservicios depende de las necesidades y metas específicas de desarrollo. Las buenas prácticas en el diseño de arquitecturas de microservicios son esenciales para maximizar los beneficios de esta arquitectura moderna y escalable.