

**Materia:**

**DISEÑO ELECTRÓNICO BASADO EN  
SISTEMAS EMBEBIDOS**

**Alumno:**

Posadas Pérez Isaac Sayeg

Paniagua Rico Juan Julian

García Azzúa Jorge Roberto

**Grado y grupo:**

8°G

**Profesor:**

Garcia Ruiz Alejandro Humberto

**Unidad 4 -Tarea 5:**

**MICROSERVICIOS**

# MICROSERVICIOS

## Introducción

En el desarrollo moderno de software, especialmente en entornos distribuidos y escalables, se ha adoptado ampliamente la **arquitectura de microservicios** como alternativa a los modelos monolíticos tradicionales. Esta arquitectura permite construir aplicaciones como un conjunto de servicios pequeños, autónomos y especializados, facilitando su desarrollo, despliegue y mantenimiento. Gracias a su modularidad, los microservicios se han convertido en una pieza clave para organizaciones que buscan agilidad, escalabilidad y eficiencia operativa.

## Desarrollo

La arquitectura de microservicios se basa en la **división de una aplicación en múltiples servicios pequeños**, cada uno enfocado en una única función o dominio del negocio. Estos servicios se comunican entre sí a través de interfaces bien definidas, generalmente mediante protocolos HTTP (REST), RPC (gRPC) o mensajería asíncrona (como RabbitMQ o Kafka).

Cada microservicio es:

- **Independiente:** puede desarrollarse y desplegarse por separado.
- **Especializado:** realiza una función específica (por ejemplo, autenticación, facturación).
- **Autónomo:** puede tener su propia base de datos y lógica de negocio.
- **Escalable individualmente:** puede escalarse de forma aislada según la demanda.

Esta arquitectura permite que distintos equipos trabajen en paralelo en diferentes microservicios, lo cual agiliza el desarrollo y mejora la calidad del software.

Además, los microservicios facilitan el uso de **contenedores** (como Docker) y herramientas de orquestación (como Kubernetes), lo que simplifica su gestión en entornos de producción.

## Ejemplo:

Imaginemos una plataforma de comercio electrónico compuesta por los siguientes microservicios:

- **Servicio de usuarios:** gestiona el registro, inicio de sesión y perfiles.
- **Servicio de productos:** maneja el catálogo, inventario y descripciones.
- **Servicio de carrito de compras:** permite agregar, modificar y eliminar productos.
- **Servicio de pagos:** procesa transacciones mediante diferentes pasarelas.
- **Servicio de notificaciones:** envía correos o mensajes de confirmación.

Cada uno de estos servicios puede escalar de forma independiente y ser actualizado sin necesidad de interrumpir el resto de la aplicación.

Por ejemplo, si durante el Black Friday hay una alta demanda en el servicio de pagos, este puede escalarse horizontalmente sin afectar al servicio de usuarios o productos.

## Conclusión

La arquitectura de microservicios ofrece una solución moderna, escalable y eficiente para el desarrollo de aplicaciones complejas. Al dividir la aplicación en servicios pequeños y autónomos, se mejora la mantenibilidad, se acelera el desarrollo y se facilita el despliegue continuo. Sin embargo, su adopción también implica desafíos técnicos como la orquestación, la comunicación entre servicios y la gestión de fallos, los cuales deben abordarse con herramientas e infraestructuras adecuadas. En conjunto, los microservicios representan un paso esencial hacia sistemas más flexibles, resilientes y preparados para entornos de nube y contenedores.

## Bibliografía

1. Newman, S. (2015). *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media.
2. Dragoni, N., et al. (2017). *Microservices: Yesterday, Today, and Tomorrow*. In *Present and Ulterior Software Engineering*, Springer.
3. Fowler, M., & Lewis, J. (2014). *Microservices - a definition of this new architectural term*. martinowler.com.
4. Richardson, C. (2018). *Microservices Patterns: With examples in Java*. Manning Publications.
5. Bass, L., Weber, I., & Zhu, L. (2015). *DevOps: A Software Architect's Perspective*. Addison-Wesley.