

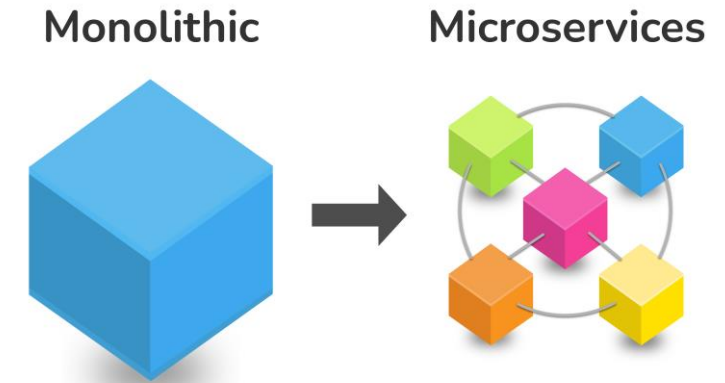


MICROSERVICIOS

Natalia Reyes Altamirano

¿QUÉ SON LOS MICROSERVICIOS?

- Es un enfoque arquitectónico y organizativo para el desarrollo de software donde el software está compuesto por pequeños servicios independientes que se comunican a través de API bien definidas. Los propietarios de estos servicios son equipos pequeños independientes.
- Las arquitecturas de microservicios hacen que las aplicaciones sean más fáciles de escalar y más rápidas de desarrollar. Esto permite la innovación y acelera el tiempo de comercialización de las nuevas características.



BENEFICIOS DE LOS MICROSERVICIOS



- **Agilidad:** Los equipos pueden trabajar de forma independiente y moverse rápidamente, lo que acorta el ciclo de desarrollo.



- **Escalado flexible:** Permiten que cada servicio se escale de forma independiente para satisfacer la demanda de la característica de la aplicación que respalda.



- **Implementación sencilla:** Los microservicios permiten la integración y la entrega continuas, lo que facilita probar nuevas ideas y revertirlas si algo no funciona.



- **Libertad tecnológica:** Los equipos tienen la libertad de elegir la mejor herramienta.



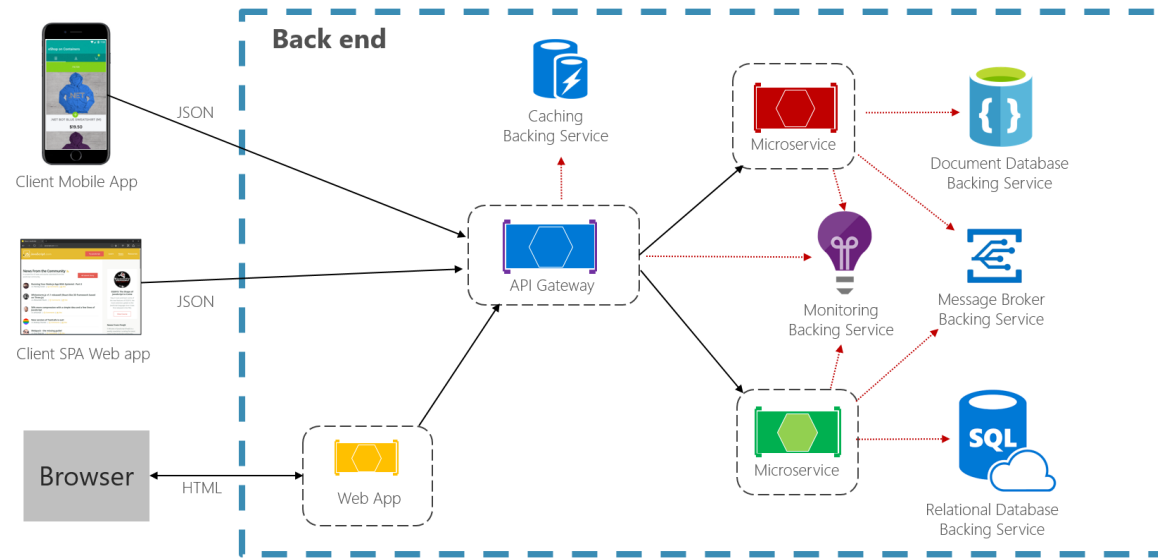
- **Código reutilizable:** Un servicio escrito para una determinada función se puede usar como un componente básico para otra característica.



- **Resistencia:** La independencia del servicio aumenta la resistencia de una aplicación a los errores.

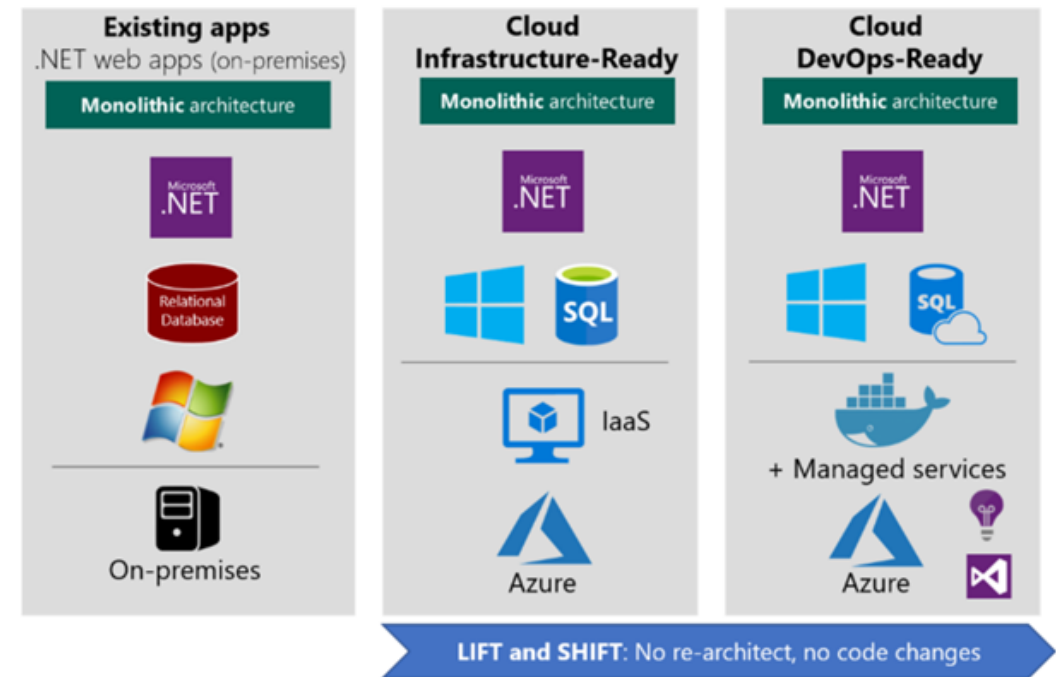
CÓMO FUNCIONA LA ARQUITECTURA DE MICROSERVICIOS

- En una arquitectura de microservicios, una aplicación compleja se desglosa en capacidades discretas que pueden desarrollarse y operarse de forma independiente. Los microservicios individuales se comunican entre sí, a menudo a través de las APIs, y tienen conexiones sueltas, pero cada microservicio puede desarrollarse, implementarse y actualizarse por separado.



IMPLEMENTACIÓN DE MICROSERVICIOS

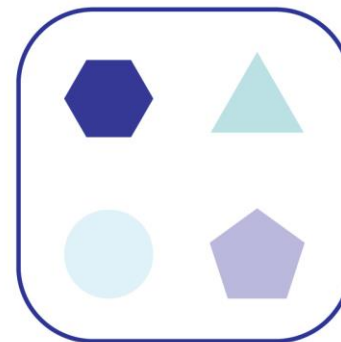
1. **Nativo de la nube.** Algunos servicios y aplicaciones establecidos de alto volumen comienzan como microservicios y permanecen en la nube.
2. **Refactorización y cambio.** Estas implementaciones comienzan en las instalaciones o en un centro de datos en el perímetro y se refactorizan para adaptarse a la arquitectura de microservicios basados en la nube. La refactorización puede incluir reasignar bases de datos y otros recursos asociados con la arquitectura monolítica, de modo que se combinen con los microservicios correspondientes.
3. **Lift and shift.** Algunas organizaciones migran sus aplicaciones a una arquitectura de microservicios, sin refactorización, en una simple transición "lift and shift".



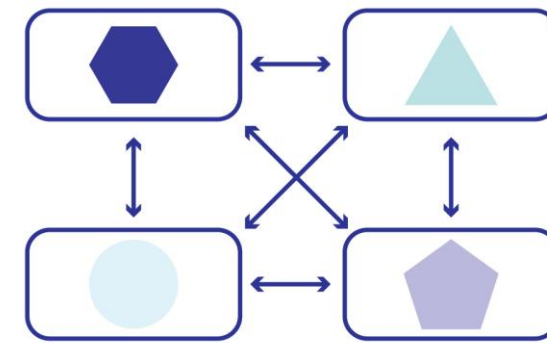
MICROSERVICIOS Y DEVOPS

- La arquitectura de microservicios se describe como optimizada para DevOps y la integración continua/entrega continua (CI/CD), y en el contexto de pequeños servicios que se pueden implementar con frecuencia.
- Dado el aumento de la complejidad y las dependencias de los microservicios, las prácticas de DevOps de automatización de la implementación, la monitorización y el ciclo de vida se consideran una parte integral de las arquitecturas de microservicios. Por eso, los microservicios a menudo se consideran el primer paso para adoptar una cultura de DevOps, que permite:
 - Automatización
 - Mayor escalabilidad
 - Facilidad de gestión
 - Agilidad
 - Entrega e implementación más rápidas

Arquitectura Monolítica



Arquitectura Microservicios

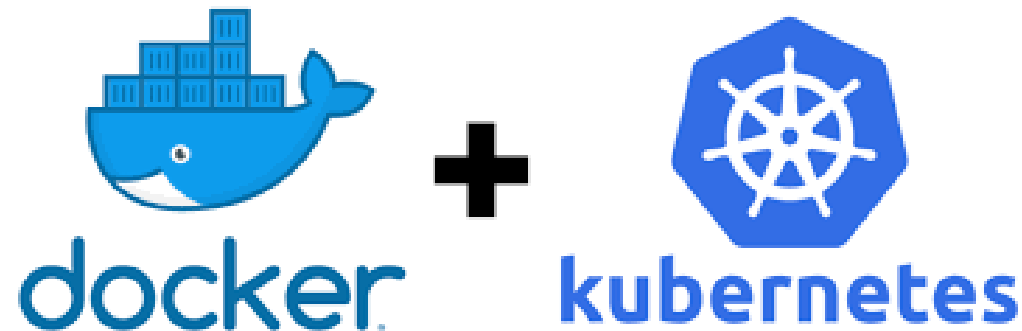


TECNOLOGÍAS Y HERRAMIENTAS

- Contenedores, Docker y Kubernetes:

Un contenedor es simplemente el paquete de una aplicación y todas sus dependencias. Docker es una plataforma de contenedorización y un tiempo de ejecución de contenedores que ayuda a crear, implementar y ejecutar contenedores. Kubernetes y otras soluciones como Docker Swarm, Mesos o HashiCorp Nomad ayudan a llevar a cabo la contenedorización a gran escala.

Docker y Kubernetes empaquetan un servicio en un contenedor completo que se puede implementar y descartar rápidamente.



TECNOLOGÍAS Y HERRAMIENTAS

- Gateways de API:

Una gateway de API actúa como un proxy inverso para los clientes mediante solicitudes de direccionamiento, distribuyendo las solicitudes por varios servicios y proporcionando seguridad y autenticación adicionales.

Existen varias tecnologías que se pueden utilizar para implementar gateways de API, incluidas las plataformas de gestión de API, pero si la arquitectura de microservicios se implementa mediante contenedores y Kubernetes, el gateway se implementa normalmente mediante Ingress o, más recientemente, Istio.



TECNOLOGÍAS Y HERRAMIENTAS

- Mensajería y streaming de eventos:

Es necesario acoplar las llamadas de API que establecen el estado con la mensajería o el streaming de eventos para que los servicios puedan emitir cambios en el estado y otros stakeholders puedan escuchar esos cambios y ajustarse a ellos. Es probable que este trabajo sea el más adecuado para un message broker de uso general, pero hay casos en los que una plataforma de streaming de eventos como, por ejemplo, Apache Kafka, puede ser una buena solución. Y con la combinación de microservicios con la arquitectura impulsada por eventos, los desarrolladores pueden crear sistemas distribuidos, altamente escalables, tolerantes a errores y ampliables que pueden consumir y procesar grandes cantidades de eventos o información en tiempo real.

