

¿Cómo dividir una aplicación monolítica en microservicios?

Alfredo Barrón Rodríguez

Centro de Investigación y de Estudios Avanzados del IPN
Unidad Tamaulipas

13 de octubre de 2021



Contenido

- 1 **Introducción**
 - Aplicaciones monolíticas
 - Aplicaciones distribuidas
- 2 **¿Qué son los microservicios?**
- 3 **Microservicios**
 - Beneficios principales
 - Integración
 - Despliegue
 - Desafíos
 - Resumen
- 4 **Dividir el monolito**
- 5 **Despliegue de microservicios**



Introducción

Las aplicaciones que carecen de una arquitectura formal suelen estar fuertemente acopladas, frágiles, difíciles de cambiar y sin una visión o dirección clara.

Las preguntas básicas sobre el despliegue y el mantenimiento son difíciles de responder:

- ¿Que grado de escalabilidad permite la aplicación?
- ¿Cuáles son las características de rendimiento de la aplicación?
- ¿Con que facilidad responde la aplicación a los cambios?
- ¿Cuales son las características de despliegue de la aplicación?



Aplicaciones monolíticas

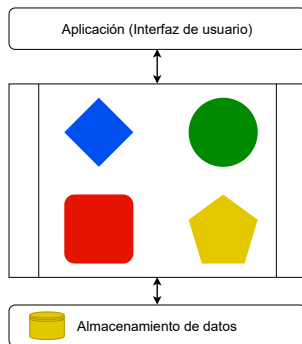


Figura 1: Su enfoque considera todos los componentes que pueden implementarse están contenidos en un solo elemento.

Ventajas

- Única base de código.
- Facilidad de desarrollo.
- Desempeño y comunicación.

Desventajas

- Punto único de falla.
- Complejo de actualizar y mantener.
- Dificultad para adoptar nuevas tecnologías.

Aplicaciones distribuidas

Ventajas

- Reutilización de servicios.
- Mejor capacidad de mantenimiento.
- Servicios mas fáciles de depurar.

Desventajas

- Gestión compleja (naturaleza distribuida).
- Sobrecarga adicional (validación de servicios).

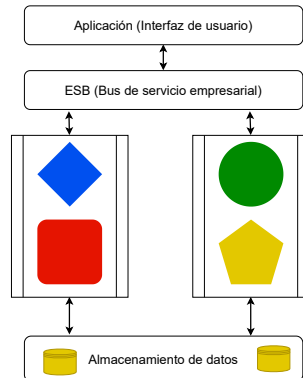


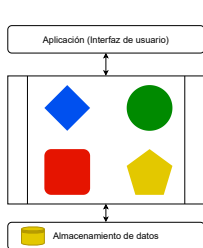
Figura 2: Su enfoque considera múltiples servicios que colaboran y se comunican a través de una red.



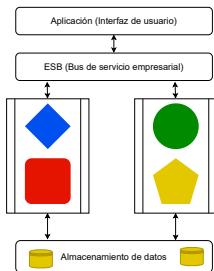
¿Qué son los microservicios?

Los microservicios evolucionaron a partir de dos fuentes principales:

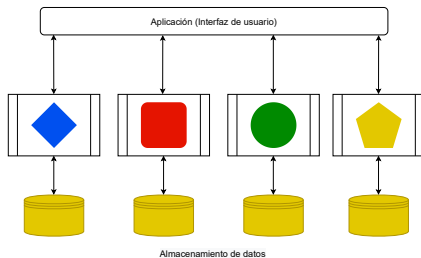
- **Aplicaciones monolíticas** desarrolladas utilizando el patrón de arquitectura por capas.
- **Aplicaciones distribuidas** desarrolladas mediante el patrón de arquitectura orientada a servicios.



Monolítico (Capas)
Una sola unidad



Orientado a Servicios (SOA)
Granularidad gruesa



Microservicios
Granularidad fina



¿Qué son los microservicios?

- Servicios pequeños y centrados en hacer una cosa bien.
- Autónomos que funcionan juntos.
- Componentes desacoplados.

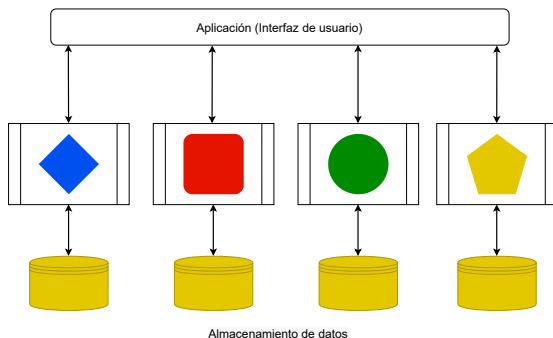
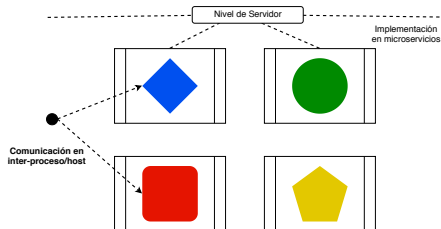


Figura 3: Arquitectura de microservicios.



¿Qué son los microservicios?

Lewis and Fowler¹ definen el estilo arquitectónico del microservicio como un enfoque para desarrollar una sola aplicación como un conjunto de pequeños servicios, cada uno ejecutándose en su propio proceso y comunicándose con mecanismos ligeros, a menudo una API² de recursos HTTP³.



¹James Lewis and Martin Fowler. Microservices: a definition of this new architectural term. Mars, 2014

²Interfaz de Programación de Aplicaciones (API, por sus siglas en inglés)

³Protocolo de Transferencia de Hipertexto (HTTP, por sus siglas en inglés)



Beneficios principales

Heterogeneidad tecnológica

- Permite construir utilizando diferentes tecnologías, marcos de trabajo y lenguajes de programación.
- Permite decidir la forma en almacenar los datos para diferentes partes del sistema.

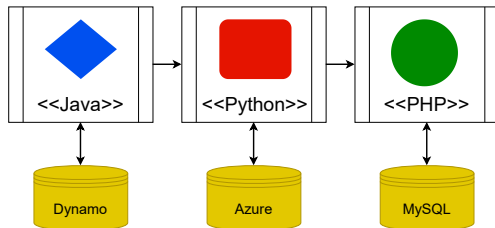


Figura 4: Los microservicios permiten adoptar más fácilmente diferentes tecnologías.



Beneficios principales

Resiliencia

- Si un componente falla, se puede aislar el problema y el resto del sistema puede seguir funcionando.

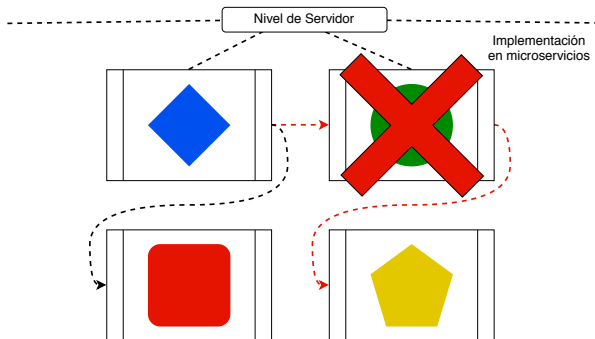


Figura 5: Los microservicios lidian con las mismas fallas que los sistemas distribuidos.



Beneficios principales

Escalado

- Pueden escalarse de manera horizontal (crear mas servicios del mismo grupo) independientemente de los demás.

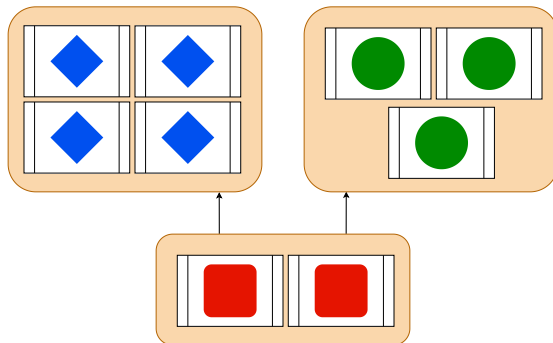


Figura 6: Permite el escalado a sólo a aquellos microservicios que lo necesiten.



Beneficios principales

Facilidad de despliegue

- Pueden construirse, probarse y desplegarse de forma independiente.
- Se reducen los riesgo del despliegue, ya que se pueden desplegar partes del software y no toda la aplicación.

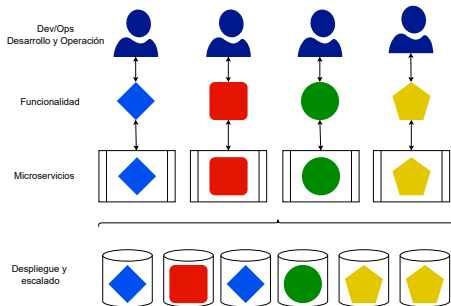


Figura 7: Los microservicios permiten la practicas de integración, entrega y despliegue continuo para el desarrollo de aplicaciones.



Integración

Los microservicios son de naturaleza distribuida, lo que significa que todos los componentes dentro de la arquitectura están totalmente desacoplados unos de otros y se accede a ellos a través de un protocolo de acceso remoto (por ejemplo REST, SOAP, RMI, etc.)

Aunque existen docenas de formas de implementar un patrón de arquitectura de microservicios, hay tres topologías principales mas comunes y populares:

- 1 La topología basada en API REST.
- 2 La topología basada en la aplicación REST.
- 3 La topología de mensajería centralizada.



Integración

¿Que es un API?

Es una Interfaz de Programación de Aplicaciones que son un conjunto de definiciones y protocolos para diseñar e integrar el software de las aplicaciones. En otras palabras las API permiten interactuar con una computadora o un sistema para obtener datos o ejecutar una función.

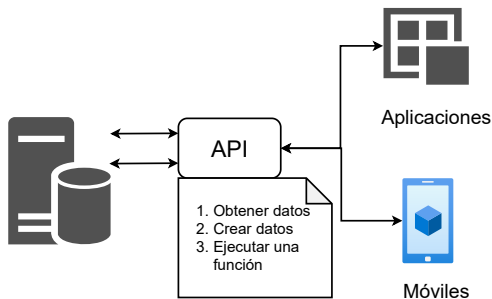


Figura 8: Ejemplo de un API.



Integración

¿Qué es REST?

REpresentational State Transfer⁴ es un conjunto de principios arquitectónicos que se ajusta a las necesidades de los servicios web y las aplicaciones móviles.

Los servicios web REST se comunican a través de la especificación HTTP utilizando el vocabulario HTTP:

- Métodos (GET, POST, etc.)
- Sintaxis HTTP URI (rutas, parámetros, etc.)
- Tipos de medios (XML, JSON, HTML, Texto plano, etc.)
- Códigos de respuesta HTTP.



⁴Transferencia de Estado Representacional

Integración

La topología basada en API REST.

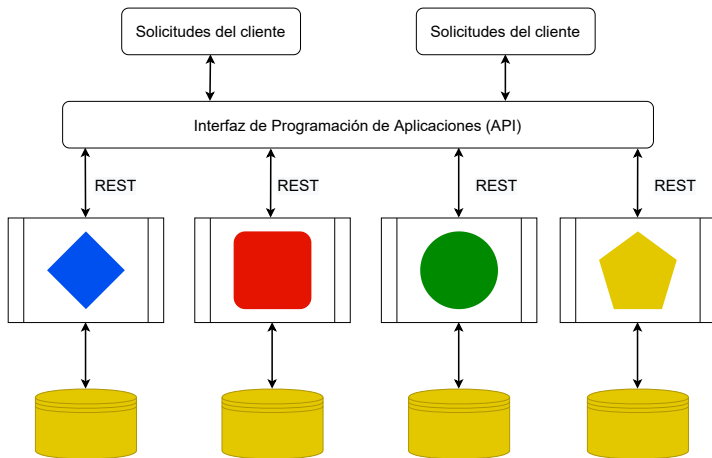


Figura 9: API REST.



Integración

La topología basada en la aplicación REST.

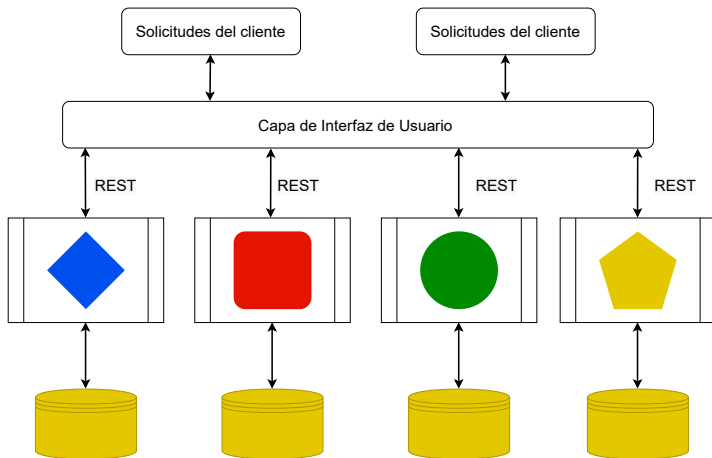


Figura 10: Aplicación REST.



Integración

La topología de mensajería centralizada.

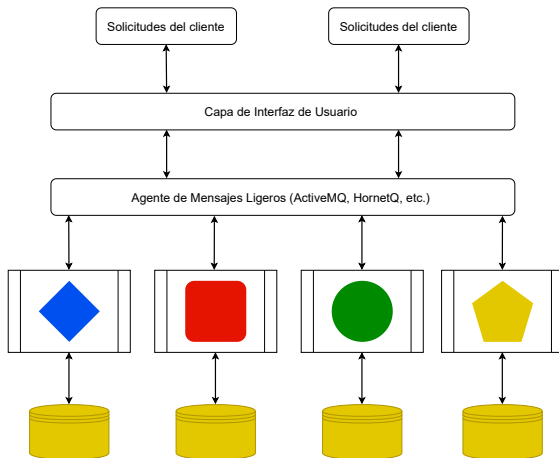


Figura 11: Intermediario de mensajería ligera.



Despliegue

¿Qué es un contenedor virtual?

Los contenedores permiten empaquetar con facilidad el código de una aplicación, sus configuraciones y sus dependencias en componentes esenciales de uso sencillo que aportan uniformidad de entorno, eficacia operativa, productividad para los desarrolladores y control de versiones.

Plataformas de despliegue

- Linux Containers (LXC)
- Docker
- Kubernetes



Desafíos

Complejidad

Dividir una aplicación en microservicios independientes conlleva mas artefactos que **gestionar**..

- Registro de todos los servicios por su entorno dinámico (Servicio de registro y descubrimiento).
- Monitoreo para identificar posibles problemas en los servicios.
- Consistencia y gestión de los datos, cada servicio tiene su propio almacenamiento de datos.
- Distribución (Latencia, Tolerancia a fallos, Seguridad, etc.).



Resumen

Ventajas y desventajas al utilizar los **microservicios**.

1 Infraestructura

- Infraestructura no dedicada (múltiples nubes).
- Puede cambiar sobre el tiempo (migración de servicios).

2 Plataforma

- Pueden ser reutilizados para ser usados en n soluciones al mismo tiempo.
- Desacoplados de infraestructuras y plataformas de manejo de datos.

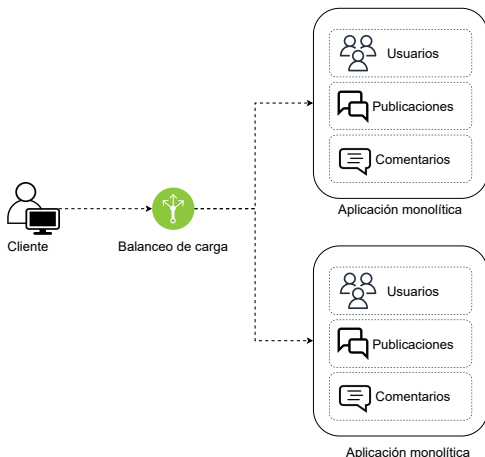
3 Software

- Pueden atender a número indeterminado de clientes (caracterización de carga compleja).



Dividir el monolito

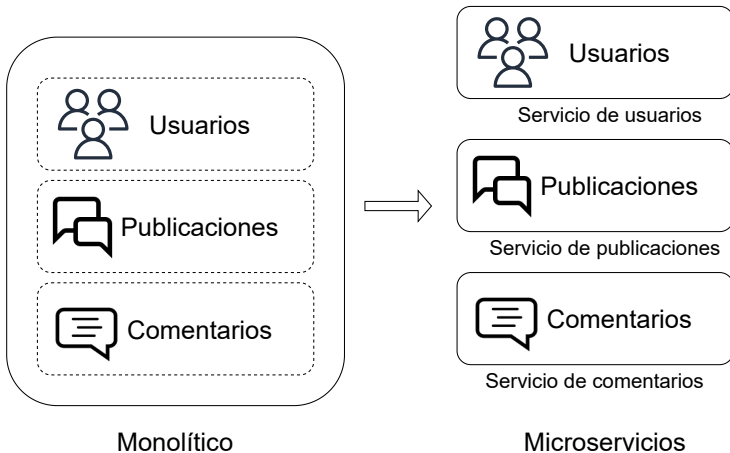
Implementación de una aplicación monolítica



- **Cliente:** realiza una solicitud o petición.
- **Balanceo de carga:** se encarga de distribuir las solicitudes.
- **Aplicación monolítica:** La arquitectura es monolítica porque comparten las mismas características que las demás aplicaciones.

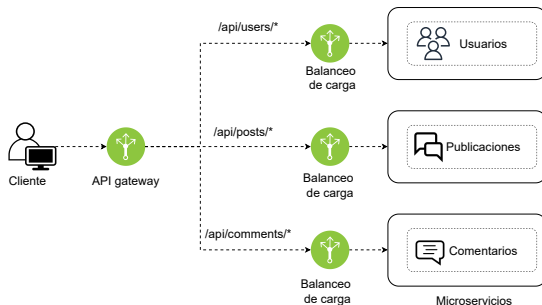


Dividir el monolito



Despliegue de microservicios

Implementación de una aplicación en microservicios



- **Cliente:** realiza una solicitud o petición.
- **API gateway:** actúa como un único punto de entrada a un grupo de microservicios.
- **Balanceo de carga:** se encarga de dirigir las solicitudes hacia el servicio correcto.
- **Microservicios:** Cada servicio gestiona una única característica.



Despliegue de microservicios

Escalabilidad en microservicios

