

MICROSERVICIOS



Microservicios

- ▶ Los microservicios son una arquitectura de software que tiene como objetivo dividir una aplicación en pequeños servicios independientes y altamente escalables. Cada servicio se encarga de una tarea específica y funciona de manera autónoma, lo que permite que se puedan desarrollar, probar y desplegar de forma independiente
- ▶ La arquitectura de microservicios mantiene un sistema similar a un gobierno descentralizado, donde cada módulo contará por ejemplo con su propia base de datos, en lugar de acudir todos a la misma sobrecargándola así de solicitudes y arriesgándonos a que si falla ésta, toda la aplicación caiga.

Características

- ▶ Algunas de las características de los microservicios son:
 - Modularidad: los servicios se pueden desarrollar, probar y desplegar de forma independiente.
 - Escalabilidad: cada servicio se puede escalar de forma independiente según la carga que reciba.
 - Tolerancia a fallos: si un servicio falla, no afecta al funcionamiento de los demás servicios.
 - Tecnología diversa: cada servicio puede utilizar la tecnología que mejor se adapte a sus necesidades.

Modularidad

- ▶ la modularidad en los microservicios permite una mayor eficiencia en el desarrollo y en la implementación de los servicios, al permitir trabajar de forma aislada en cada uno de ellos y desplegarlos
- ▶ cada servicio se puede desarrollar y mantener de forma separada, sin afectar el funcionamiento de los demás servicios.



Escalabilidad

permite una mejor utilización de los recursos de hardware y una mayor capacidad de respuesta del sistema en general.

Tolerancia a Fallos

- ▶ la capacidad del sistema para continuar funcionando a pesar de que uno o varios servicios fallen. Esto gracias a que están diseñados para ser independientes y autónomos. Cada servicio se ejecuta en su propio proceso y puede ser desplegado en un servidor diferente.

Tecnología diversa

- ▶ La ventaja de utilizar microservicios es que cada servicio puede utilizar la tecnología que mejor se adapte a sus necesidades, sin que esto afecte a los demás servicios.
- ▶ Por ejemplo, un servicio que requiere de un alto rendimiento puede utilizar tecnologías específicas para mejorar su velocidad de respuesta, mientras que otro servicio que requiere de una mayor seguridad puede utilizar tecnologías que le permitan garantizar la protección de los datos.
- ▶ Deviene en eficiencia y flexibilidad

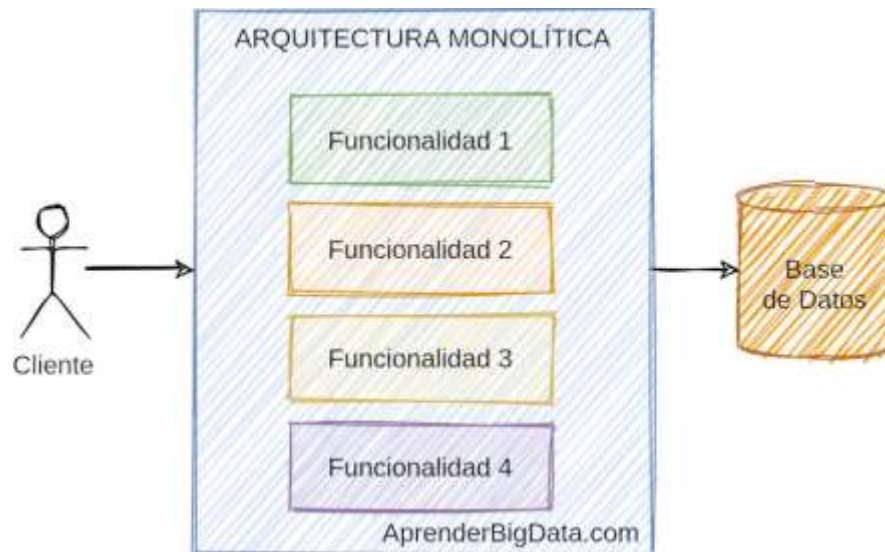
Comparaciones con el enfoque de la arquitectura Monolítica

- Enfoque de desarrollo de software en el que todas las funcionalidades de una aplicación se implementan en una sola unidad de código que se ejecuta en un solo proceso o contenedor. En una aplicación monolítica, todas las funciones están fuertemente acopladas, lo que significa que un cambio en una parte de la aplicación puede tener un efecto en cascada en otras partes.

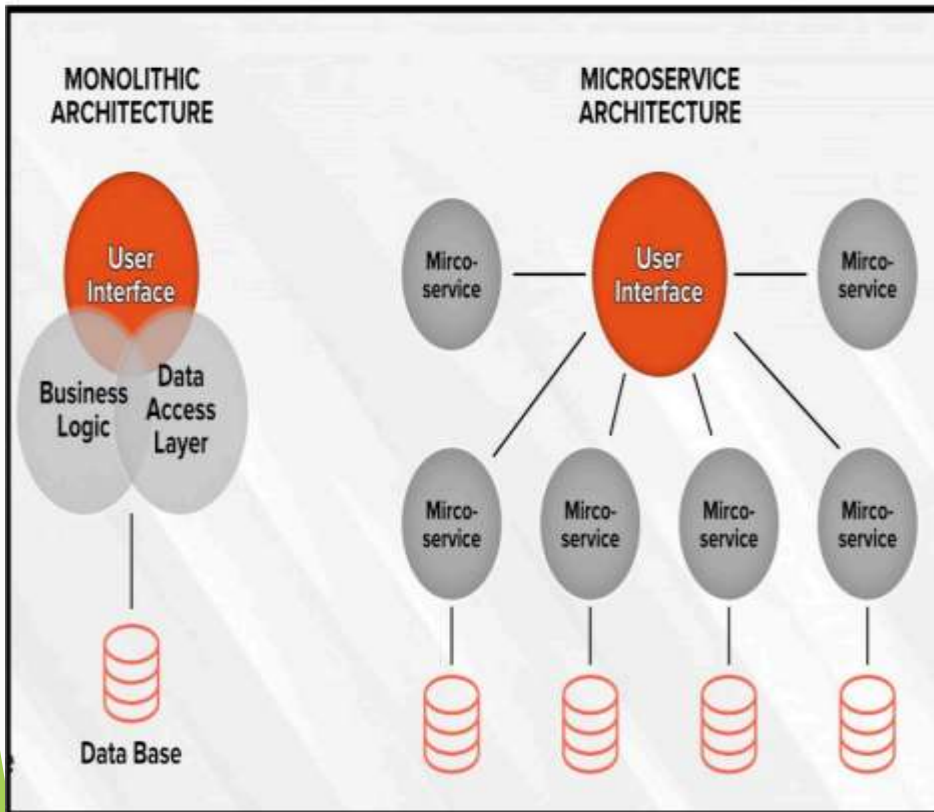
Limitaciones del enfoque Monolítico

- ▶ Acoplamiento fuerte: En una arquitectura monolítica, los componentes del sistema están fuertemente acoplados, lo que significa que cualquier cambio en una parte del sistema puede tener efectos secundarios en otras partes del sistema.
- ▶ Escalabilidad limitada: La arquitectura monolítica tiene una capacidad limitada para escalar
- ▶ Dificultad para adoptar nuevas tecnologías: En una arquitectura monolítica, todos los componentes del sistema utilizan la misma tecnología y lenguaje de programación

- ▶ gran cantidad de interdependencias entre los diferentes componentes.
- ▶ Mayor tiempo de desarrollo: El enfoque monolítico puede ser más lento para desarrollar y desplegar en comparación con el enfoque de microservicios
- ▶ No existe un cambio aislado, o individual como tal, ya que debe ser probado en todo el sistema



Comparación entre arquitecturas



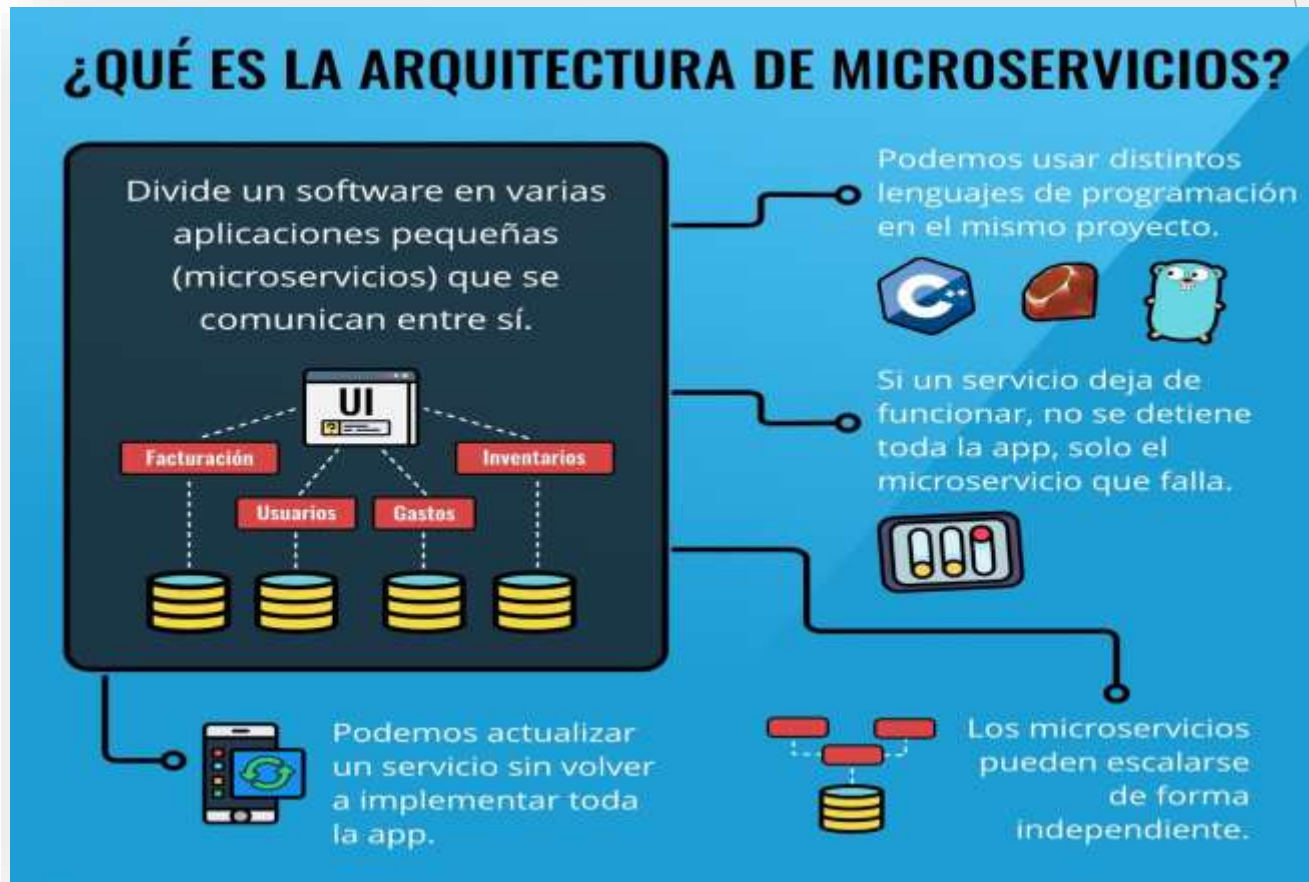
Monolito

- Arquitectura de capas.
- Una sola BBDD.

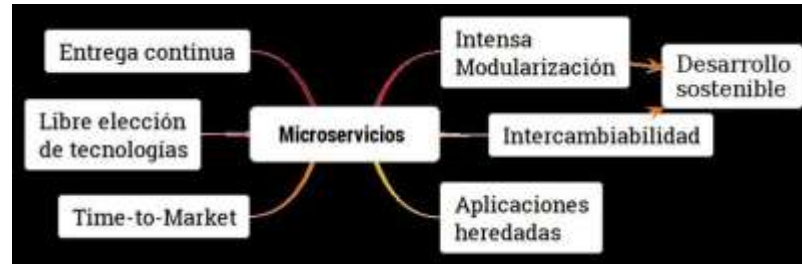
Microservicio

- Modularización.
- BBDD por bloque.
- Desarrollo independiente.

Beneficios y costos de la arquitectura de microservicios y en qué casos usarla



Los beneficios



- ▶ **Permite una programación políglota:** los microservicios pueden estar escritos en diferentes lenguajes de programación según la necesidad. Si un micro servicio necesita mucha velocidad, lo puedes escribir en Rust o en Go.
- ▶ **Más fácil de escalar la aplicación:** porque se pueden escalar los microservicios más solicitados (que forman el cuello de botella) sin tener que escalar toda la aplicación.
- ▶ **Más fácil de entender:** los microservicios son más fáciles de entender individualmente, porque cada micro servicio tiene una función precisa y es bastante independiente.

Los costos

Resulta que la arquitectura de micro servicios también tiene muchos costos. Estos son unos de los más importantes:

- ▶ **Mayor complejidad de operación:** con un monolito, cada vez que despliegues, despliegues solamente un servicio. Ahora, imagina que tienes que desplegar 5 microservicios. Seguramente el esfuerzo necesario va a aumentar drásticamente.
- ▶ **Difícil de cambiar las juntas entre los servicios:** si usaste micro servicios demasiado temprano en tu aplicación y te das cuenta que no definiste bien sus fronteras, puede resultar muy costoso arreglar este error.
- ▶ **Más difícil de depurar:** si hay un defecto en tu micro servicio de pagos, puede ser que esto resulte en un error en otro micro servicio. Es mucho más difícil rastrear de dónde viene el problema cuando usas micro servicios.



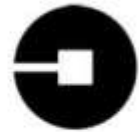
Por qué usar micro servicios?

- Los servicios en sí son muy simples de construir, pues se centran en hacer solamente una cosa bien, de forma que son fáciles de probar y se puede asegurar mayor calidad.
- Cada servicio podría construirse con las tecnologías y herramientas más adecuadas, permitiendo "Polyglot Programming" (las aplicaciones se deben escribir en una mezcla de lenguajes para explotar sus mejores características).
- Múltiples equipos pueden trabajar independientemente. Esto fomenta "continuous delivery" debido a que permite actualizaciones frecuentes mientras el resto del sistema se mantiene estable.
- Si un servicio deja de funcionar, solo afectará las partes que dependen directamente de él (si las hay). El resto operará normalmente.

Empresas que han implementado
multiservicios y algunas ventajas que
han obtenido



Spotify®



UBER

NETFLIX

amazon

- Escalabilidad
- Mejorar su Desarrollo
- Eficiencia
- Facilitar nuevas funcionalidades