

# Aplicaciones monolíticas y microservicios

## ***Aplicación monolítica***

Una aplicación monolítica es una aplicación de software que se desarrolla como una sola unidad de código fuente. En una aplicación monolítica, todas las funciones y componentes de la aplicación se implementan como un solo programa y se ejecutan en una sola instancia de proceso.

En este tipo de aplicación, todos los módulos y componentes comparten el mismo espacio de memoria y la misma base de código, lo que facilita la comunicación entre ellos. Sin embargo, a medida que las aplicaciones se vuelven más grandes y complejas, las aplicaciones monolíticas pueden ser difíciles de mantener y actualizar. Por ejemplo, si se necesita actualizar una función en una aplicación monolítica, se debe volver a compilar y desplegar toda la aplicación y esto puede ser un proceso costoso y lento.

## ***Microservicio***

Un microservicio es una arquitectura de software que se enfoca en construir aplicaciones como un conjunto de servicios pequeños e independientes que se comunican entre sí a través de interfaces bien definidas y estándares. Cada servicio se desarrolla y se implementa de manera autónoma y se puede escalar y actualizar de forma independiente.

Cada microservicio se centra en una función de negocio específica y se desarrolla como un servicio independiente, lo que permite una mayor modularidad y flexibilidad en el diseño de aplicaciones. Los microservicios se pueden implementar y desplegar utilizando diferentes tecnologías, estos se comunican entre sí a través de interfaces bien definidas, como APIs RESTful, que permiten a los servicios interactuar de forma coherente y consistente. Cada servicio puede tener su propia base de datos, lo que significa que los datos se pueden escalar y actualizar de forma independiente.

## ***Aplicación monolítica vs microservicio***

Mientras que una app de microservicios está constituida por diversos módulos y componentes que se comunican entre sí, las aplicaciones monolíticas destacan por agrupar todas las funciones en un solo gran código.

Algunos conceptos clave que diferencian una de otra:

- **Eficiencia:** En un comienzo, se creía que aplicaciones monolíticas tendían a ser más eficientes, al estar todo aglomerado en un mismo código. Con todo, el avance en las tecnologías de la información, y a la rapidez de las arquitecturas cloud y de internet han permitido que los códigos de microservicios se desplieguen de forma igualmente efectiva.
- **Probabilidad de errores:** Al tratarse de un proceso de desarrollo lineal, la arquitectura monolítica tendía a mostrar de forma inmediata los posibles bugs, razón por la que muchas veces se prefería frente a microservicios, que supuestamente eran más propensos a las fallas de código al ser estructuras más complejas. Pero hoy, gracias a metodologías más ágiles y a procesos de QA Testing más elaborados, se ha logrado reducir de forma considerable los errores entre microservicios conectados.
- **Rigidez y flexibilidad:** Una aplicación de microservicios tiende a ser más flexible, adaptable y multifuncional frente a todo tipo de entornos, ofreciendo además cierta autonomía a cada uno de los equipos desarrolladores.
- **Tamaño y complejidad:** En una aplicación monolítica, todos los componentes de la aplicación se desarrollan, empaquetan y despliegan juntos en una sola unidad. En cambio, en una arquitectura de microservicios, la aplicación se divide en servicios más pequeños e independientes. Esto significa que una aplicación monolítica puede ser más grande y compleja que una aplicación de microservicios.
- **Escalabilidad:** En una aplicación monolítica, se debe escalar toda la aplicación en su conjunto, lo que puede ser ineficiente y costoso en términos de recursos. En una arquitectura de microservicios, los servicios se pueden escalar individualmente, lo que permite una mejor utilización de los recursos y una mayor eficiencia.
- **Mantenimiento y actualización:** En una aplicación monolítica, cualquier actualización o cambio en una función o componente puede requerir la

actualización y la implementación de toda la aplicación. En cambio, en una arquitectura de microservicios, los servicios se pueden actualizar y desplegar de manera independiente, lo que permite una mayor flexibilidad y eficiencia en la implementación de cambios y actualizaciones.

- **Comunicación y acoplamiento:** En una aplicación monolítica, todos los componentes están altamente acoplados y comparten la misma base de código, lo que puede hacer que la comunicación entre ellos sea más fácil pero también puede generar dependencias no deseadas. En cambio, en una arquitectura de microservicios, los servicios se comunican a través de interfaces bien definidas, lo que puede aumentar la cohesión y reducir el acoplamiento.