

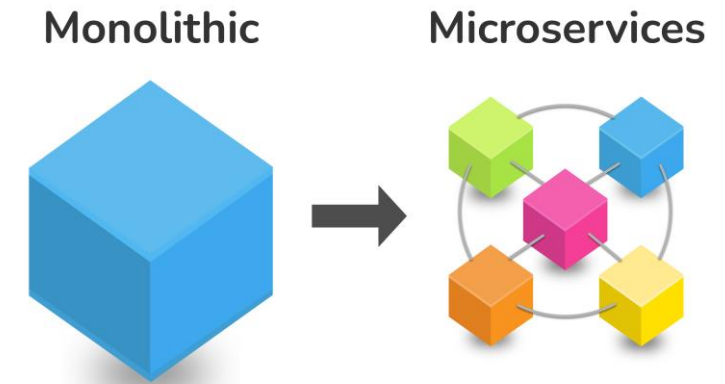


APLICACIÓN MONOLÍTICA

Natalia Reyes Altamirano

¿QUÉ ES UNA ARQUITECTURA MONOLÍTICA?

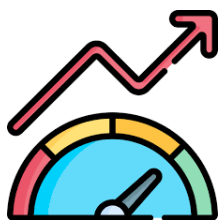
- Es un modelo tradicional de un programa de software que se compila como una unidad unificada y que es autónoma e independiente de otras aplicaciones.
- Una arquitectura monolítica es una red informática grande y única, con una base de código que aúna todos los intereses empresariales. Para hacer cambios en este tipo de aplicación, hay actualizar toda la pila, lo que requiere acceder a la base de código y compilar e implementar una versión actualizada de la interfaz del lado del servicio. Esto hace que las actualizaciones sean restrictivas y lentas.



VENTAJAS DE UNA ARQUITECTURA MONOLÍTICA



- **Implementación sencilla:** un único archivo o directorio ejecutable facilita la implementación.



- **Desarrollo:** desarrollar una aplicación compilada con una única base de código es más sencillo.
- **Rendimiento:** en una base de código y un repositorio centralizados, una API suele realizar la misma función que muchas API en el caso de los microservicios.



- **Pruebas simplificadas:** una aplicación monolítica es una unidad única y centralizada, por lo que las pruebas integrales se pueden hacer más rápido que con una aplicación distribuida.
- **Depuración sencilla:** con todo el código ubicado en un solo lugar, es más fácil rastrear las solicitudes y localizar incidencias.

DESVENTAJAS DE UNA ARQUITECTURA MONOLÍTICA



- **Velocidad de desarrollo más lenta:** con una aplicación grande y monolítica, el desarrollo es más complejo y lento.



- **Escalabilidad:** no se pueden escalar componentes individuales.
- **Fiabilidad:** si hay un error en algún módulo, puede afectar a la disponibilidad de toda la aplicación.



- **Barrera para la adopción de tecnología:** cualquier cambio en el marco o el lenguaje afecta a toda la aplicación, lo que hace que los cambios suelen ser costosos y lentos.
- **Falta de flexibilidad:** un monolito está limitado por las tecnologías que se utilizan en él.
- **Implementación:** un pequeño cambio en una aplicación monolítica requiere una nueva implementación de todo el monolito.

CÓMO FUNCIONA LA ARQUITECTURA DE MICROSERVICIOS

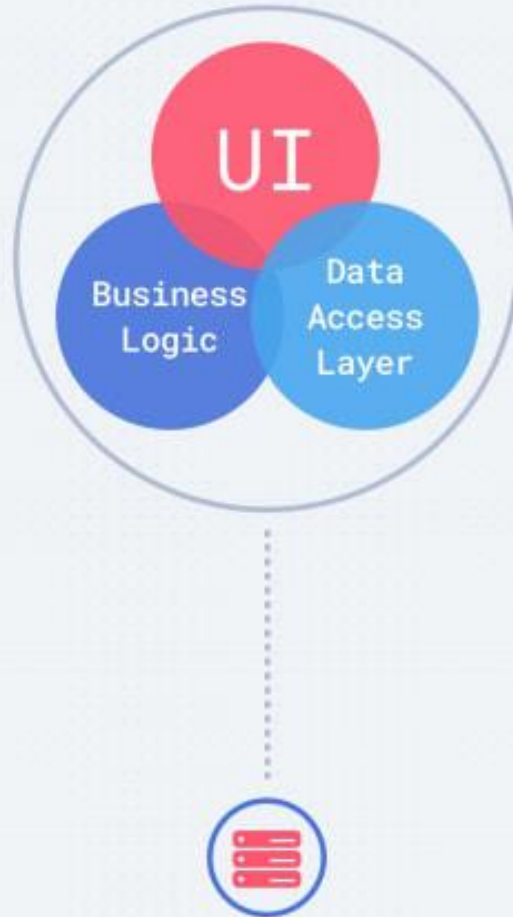


- **Componentes integrados:** En una arquitectura monolítica, todos los componentes de la aplicación, como módulos, funciones y librerías, están agrupados en una sola base de código. Esto facilita la comunicación y el acceso entre diferentes partes de la aplicación.
- **Interfaz de usuario:** La interfaz de usuario (UI) es la capa con la que los usuarios interactúan. En una arquitectura monolítica, la UI se encuentra generalmente en la misma aplicación que el resto de la lógica, lo que puede simplificar el desarrollo y la administración.
- **Lógica empresarial:** La lógica empresarial, que incluye todas las reglas y procesos específicos de la aplicación, también reside dentro del mismo monolito. Esto significa que el procesamiento de datos, la lógica de negocio y la toma de decisiones se realizan en una única unidad.
- **Acceso a la base de datos:** En una arquitectura monolítica, las interacciones con la base de datos también ocurren en la misma aplicación. Esto se hace a través de conexiones directas a la base de datos desde el código de la aplicación.
- **Escalabilidad:** A medida que la demanda de la aplicación aumenta, la escalabilidad puede ser un desafío. En general, para escalar una arquitectura monolítica, es necesario replicar la aplicación completa en múltiples servidores, lo que puede ser más complicado que escalar sistemas más modulares.
- **Mantenimiento y despliegue:** El mantenimiento y la implementación en una arquitectura monolítica pueden ser más simples, ya que solo hay un componente para gestionar. Sin embargo, las actualizaciones pueden ser más complejas, ya que cualquier cambio afectará a toda la aplicación.

DIFERENCIAS ENTRE ARQUITECTURA MONOLÍTICA Y MICROSERVICIOS

Categoría	Arquitectura monolítica	Arquitectura de microservicios
Diseño	Código base único con múltiples funciones interdependientes.	Componentes de software independientes con funcionalidad autónoma que se comunican entre sí mediante API.
Desarrollo	Al principio requiere menos planificación, pero se hace cada vez más compleja de entender y mantener.	Al principio requiere más planificación e infraestructura, pero se hace más fácil de administrar y mantener con el tiempo.
Implementación	Toda la aplicación se implementó como una sola entidad.	Cada microservicio es una entidad de software independiente que requiere una implementación individual en contenedores.
Depuración	Hace un seguimiento de la ruta de acceso al código en el mismo entorno.	Requiere herramientas de depuración avanzadas para hacer un seguimiento del intercambio de datos entre varios microservicios.
Modificación	Pequeños cambios introducen mayores riesgos, ya que afectan a todo el código base.	Puede modificar microservicios individuales sin afectar a toda la aplicación.
Escalado	Debe escalar toda la aplicación, incluso si solo ciertas áreas funcionales experimentan un aumento de la demanda.	Puede escalar microservicios independientes según sea necesario, lo que ahorra costos generales de escalado.
Inversión	Baja inversión inicial a costa de aumentar los esfuerzos continuos y de mantenimiento.	Inversión adicional de tiempo y costos para configurar la infraestructura requerida y desarrollar la competencia del equipo. Sin embargo, ahorro de costos a largo plazo, mantenimiento y adaptabilidad.

Monolithic Architecture



Microservices Architecture

