

Monolitos vs. Microservicios: ¿qué arquitectura elegir?

10/03/2026 · **Fernando de la Torre** · Arquitecto de Software

Contenido

¿Qué es una arquitectura monolítica o de monolitos?

Ventajas de los monolitos

Desventajas de los monolitos

¿Qué es una arquitectura de microservicios?

Ventajas de los microservicios

Desventajas de los microservicios

¿Qué arquitectura elegir?

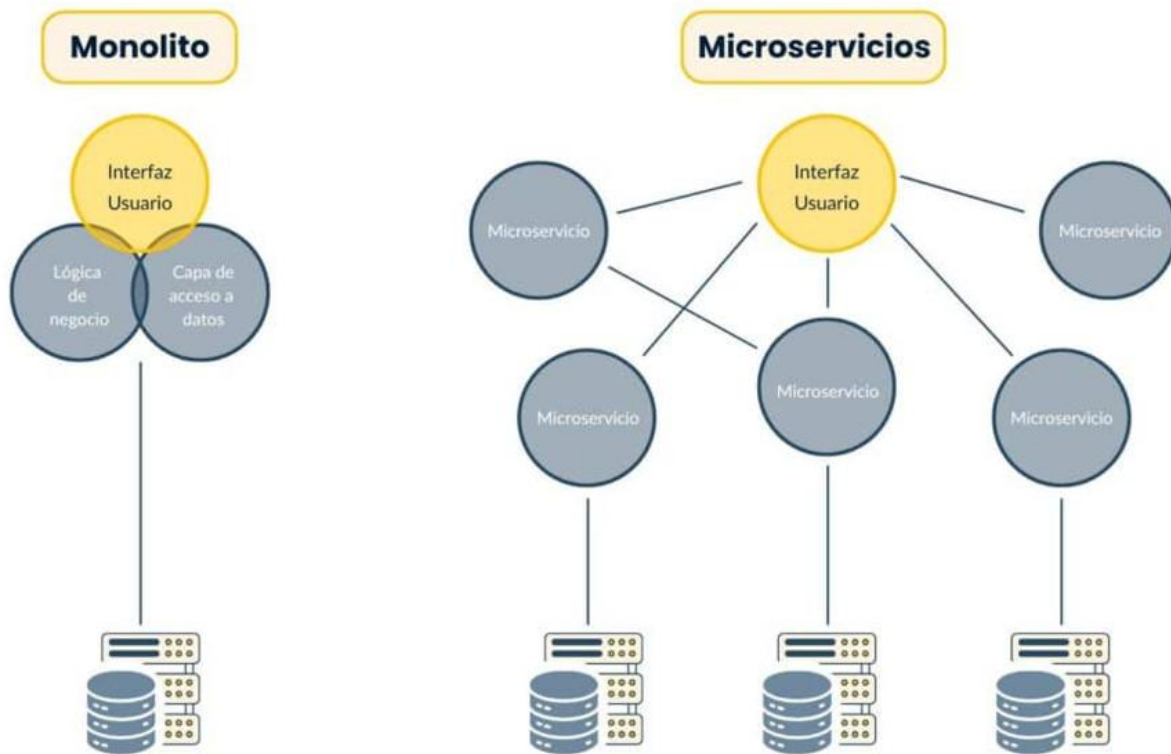
El monolito modular: una opción actual

Conclusión

Elegir cómo construir tu software siempre será clave para tu equipo. En 2026, la cuestión no es si un **monolito** o **microservicios** es mejor, sino **qué opción se ajusta mejor a tu situación**.

Después de que muchos adoptaran los microservicios (a veces con problemas), aprendimos algo importante: **no todas las apps necesitan dividirse en muchos servicios**. Ahora somos más prácticos y menos influenciados por tendencias pasajeras.

Aquí exploraremos las **características, pros y contras de ambas arquitecturas**, así como lo que está de moda, para que puedas decidir mejor en 2026.



La tecnología ha progresado mucho. Usar la nube, contenedores con **Docker**, [Kubernetes](#), **plataformas PaaS**, **CI/CD** y herramientas para ver cómo funciona todo es común.

Pero esta experiencia dejó lecciones. Muchas empresas usaron microservicios muy pronto, sin tener el tamaño, el equipo o la forma de trabajar correctos. Esto resultó en **sistemas complicados, caros de mantener y difíciles de entender**.

Actualmente, el mensaje es claro: **la arquitectura debe resolver tu problema, no al revés**.

¿Qué es una arquitectura monolítica o de monolitos?

Es un modelo de desarrollo de software en el que **todas las funciones de una aplicación se integran en un solo código base**, es decir, cuando toda tu aplicación está en un solo bloque de código. La **interfaz, la lógica y el acceso a los datos están conectados y funcionan juntos**.

Por años, fue lo normal y aún se usa mucho, sobre todo en apps pequeñas y medianas.

Ventajas de los monolitos

- **Facilidad de despliegue:** el despliegue de una aplicación monolítica es sencillo, ya que todo el código está en un solo lugar.
- **Pruebas y depuración:** es más fácil realizar pruebas y depurar una aplicación monolítica, ya que todo el código está en un solo proyecto.
- **Menor latencia:** al no depender de múltiples servicios, la latencia de comunicación es menor.
- **Agilidad inicial:** para proyectos pequeños o en etapas iniciales, una arquitectura monolítica puede ser más rápida de desarrollar y desplegar.

Desventajas de los monolitos

- **Complejidad de mantenimiento:** a medida que la aplicación crece, el código puede volverse complejo y difícil de mantener.
- **Escalabilidad limitada:** escalar una aplicación monolítica puede ser complicado, ya que requiere desplegar toda la aplicación, incluso si solo una parte necesita más recursos.
- **Dependencia tecnológica:** una vez que se elige una tecnología para un monolito, es difícil cambiarla sin afectar a toda la aplicación.

¿Qué es una arquitectura de microservicios?

La arquitectura de [microservicios](#) divide una **app en pequeños servicios** que se comunican a través de **APIs**. Cada uno se encarga de algo específico y puede ser **armado, puesto en marcha y escalado por separado**.

Ventajas de los microservicios

- **Escalabilidad:** los microservicios permiten escalar solo las partes de la aplicación que lo necesitan, lo que puede ser más eficiente.
- **Despliegue independiente:** cada microservicio puede ser desplegado de manera independiente, lo que facilita las actualizaciones y mejoras.
- **Flexibilidad tecnológica:** los equipos pueden elegir diferentes tecnologías para cada microservicio, lo que permite utilizar la mejor herramienta para cada tarea.
- **Resiliencia:** si un microservicio falla, no necesariamente afecta a toda la aplicación.
- **Equipos autónomos:** los microservicios fomentan la formación de equipos más autónomos y responsables, cada uno encargado de un servicio específico.

Desventajas de los microservicios

- **Complejidad:** gestionar múltiples microservicios puede ser complejo y requiere una infraestructura robusta para manejar la comunicación y la orquestación.
- **Comunicación entre servicios:** la comunicación entre microservicios puede ser un punto débil y requiere una gestión cuidadosa de latencias, errores de red y problemas de seguridad.
- **Sobrecarga operativa:** la necesidad de monitorear y mantener múltiples servicios puede aumentar la carga operativa.

¿Qué arquitectura elegir?

No hay una respuesta única. **Depende del tamaño del proyecto, del equipo** y de lo que quieras a largo plazo:

Criterio	Arquitectura Monolítica	Arquitectura de Microservicios
Tamaño y complejidad del proyecto	Ideal para proyectos pequeños o simples . Toda la aplicación está en un solo bloque, lo que facilita el desarrollo inicial.	Más adecuada para proyectos grandes y complejos , donde cada servicio maneja una funcionalidad específica.
Escalabilidad	Escala como un todo; puede volverse limitada y costosa a medida que crece la aplicación.	Permite escalar servicios de forma independiente , optimizando recursos.
Flexibilidad tecnológica	Generalmente usa una sola tecnología o stack . Cambios tecnológicos son más difíciles.	Cada microservicio puede usar tecnologías distintas , mayor flexibilidad.
Recursos y habilidades del equipo	Recomendado para equipos pequeños o con poca experiencia , por su simplicidad.	Más adecuado para equipos grandes y especializados , con experiencia en DevOps, contenedores y comunicación entre servicios.
Despliegue	Despliegue simple y rápido , un solo artefacto.	Despliegue más complejo , requiere automatización y orquestación.
Mantenimiento	Fácil al inicio, pero puede volverse difícil a medida que crece el sistema.	Mejor mantenimiento a largo plazo gracias a la separación de responsabilidades.
Time-to-market	Más rápido para lanzar un producto inicial (MVP).	Inicialmente más lento, pero más eficiente en evoluciones futuras.
Objetivos a largo plazo	Adecuado si no se espera un gran crecimiento.	Ideal si se anticipa crecimiento, alta demanda y evolución constante .

El monolito modular: una opción actual

El monolito se ha convertido en modular. Sigue siendo **fácil de manejar, pero usa buenas prácticas**. La aplicación sigue siendo un solo bloque, pero el **código se organiza en partes con tareas claras**.

Esto ayuda a mantener el sistema, reduce los problemas técnicos y, si es necesario, permite separar partes como microservicios. Por eso, muchos lo ven como un punto medio entre un monolito común y los microservicios.

Conclusión

La pregunta no es “¿microservicios o monolito?”, sino “**¿qué tiene sentido para mí ahora?**”. Los **microservicios** son buenos para **sistemas grandes**, pero no son la única solución. Los **monolitos**, sobre todo los modulares, son una **buena alternativa**.

La **mejor arquitectura es la que cambia con tu producto, tu equipo y tu empresa**, evitando lo innecesario y buscando que todo sea claro y fácil de mantener a largo plazo.