



Aplicaciones modernas: [arquitecturas de microservicios](#)

Agilidad empresarial con arquitectura moderna



La proliferación de la informática veloz y asequible ha permitido a empresas de todos los tamaños lograr eficiencias internas y llegar a más clientes a través de productos digitales. Sin embargo, la omnipresencia de las herramientas, los diferentes caminos para llegar al mercado y las preferencias cambiantes de los consumidores obligan a las empresas a innovar más rápido que nunca para seguir siendo competitivas. En todas las industrias, desde la agrícola hasta la bancaria y de las telecomunicaciones, la innovación rápida, en esencia, es digital. La forma en que los productos digitales se han desarrollado tradicionalmente no es lo suficientemente rápida como para permitir la innovación necesaria para ganar en el mercado.

Nuevos patrones de arquitectura

Los nuevos patrones de arquitectura, como los microservicios, permiten que las organizaciones aceleren el ritmo de la innovación. Las aplicaciones modernas creadas con arquitecturas de microservicios permiten la innovación al distribuir el esfuerzo y la inversión a lo largo del tiempo y entre equipos más pequeños. De esta forma, se aumenta la velocidad de las pruebas y de la entrega de cambios al mercado. Permiten una optimización de recursos minuciosa y los equipos pueden escalar rápidamente tanto en la forma en que crean los productos como en la manera en que los ejecutan.

¿Cuáles son las características de una arquitectura de microservicios?

Especializada

Cada servicio está diseñado para un conjunto de capacidades y se centra en resolver un problema específico. Si los desarrolladores contribuyen con más código a un servicio a lo largo del tiempo y este se vuelve complejo, se puede dividir en servicios más pequeños.

Distribuida

Una arquitectura de microservicios divide su aplicación de un solo proceso en múltiples componentes que funcionan juntos para entregar valor. Cualquier comunicación entre componentes individuales se produce a través de API bien definidas y poco acopladas, o mediante eventos y mensajes.

Autónoma

Cada servicio que compone una arquitectura de microservicios se puede desarrollar, implementar, operar y escalar sin afectar el funcionamiento de otros servicios. Los servicios no necesitan compartir su código o implementación con otros servicios. Actúan como cajas negras autónomas.

Un monolito puede funcionar bien hoy, pero a menudo surgen desafíos a medida que su empresa crece. Los microservicios lo ayudan a abordar desafíos comunes, como escalar e implementar nuevas características rápidamente.



Monolito



Microservicios

Separación de un monolito

La separación de un monolito puede ser intimidante. Invite a su equipo a completar un proyecto de AWS para que adquiera práctica y tenga una guía durante el proceso de separación de un monolito en microservicios.

[Pruebe el tutorial >>](#)



El traspaso a los microservicios

Descubra cómo Mobvista adoptó una arquitectura de microservicios para mejorar la escalabilidad y fiabilidad de su plataforma. [Lea el caso práctico >>](#)

La ventaja de los microservicios

Las arquitecturas de microservicios se crean con elementos modulares discretos que funcionan juntos. Si bien esta modularidad tiene el desafío de una mayor “superficie” de código, también ofrece ventajas clave para innovar más rápido, escalar de forma independiente, reducir el impacto de los errores y permitir el desarrollo de código distribuido.

Los beneficios de los microservicios

1. Agilidad: un equipo pequeño que trabaja en un componente de servicio individual se libera de las limitaciones de los otros componentes y, por lo tanto, puede ser más ágil y responder a problemas u oportunidades más rápido.

2. Implementación sencilla: los microservicios reducen el tamaño de los cambios, lo que facilita probar nuevas ideas y revertirlas si algo no funciona.

3. Libertad tecnológica: las arquitecturas de microservicios brindan a los equipos la libertad de elegir las mejores herramientas para crear cada parte de una aplicación.

4. Escalabilidad: los microservicios permiten escalar cada servicio de forma independiente para satisfacer la demanda de la característica que admite de la aplicación.

5. Resiliencia: los microservicios reducen el impacto de los errores a una sola parte de la aplicación. Esto significa que cualquier error de un componente individual solo degrada la funcionalidad en lugar de afectar a toda la aplicación.

6. Código reutilizable: la división del software en componentes pequeños y bien definidos de servicios permite que los equipos los utilicen para fines múltiples dentro de una aplicación. Esto permite que la aplicación se aproveche a sí misma, ya que los desarrolladores pueden crear nuevas capacidades aprovechando los servicios existentes a través de sus API, sin necesidad de escribir código desde cero o tener que lidiar con los detalles de la implementación.



Microservicios para aplicaciones modernas

Los microservicios ayudan a su organización a mejorar la resiliencia de las aplicaciones y optimizar la productividad del equipo. Como resultado, los equipos de desarrollo pueden experimentar e innovar más rápido para lanzar productos y características que ofrecen una ventaja competitiva.

Dónde comenzar

La adopción de patrones de arquitectura de microservicios no tiene por qué ser un esfuerzo de todo o nada. Hay dos caminos comunes hacia una arquitectura orientada a los servicios: (a) ajustar el monolito existente en varias API y tratarlo como una caja negra mientras se crean nuevas funcionalidades netas como microservicios; (b) refactorizar

el monolito para los microservicios con el patrón estrangulador. Ambos tienen ventajas y desventajas, pero cualquiera sea el camino que elija, será necesario configurar primero la infraestructura de desarrollo adecuada. Esto incluye la creación de canalizaciones de entrega de software automatizadas para crear, probar e implementar de forma independiente servicios ejecutables y la infraestructura para proteger, monitorear, operar y depurar un sistema distribuido.

Mantener un monolito como este puede funcionar si se trata de un sistema independiente que no requiere actualizaciones de la funcionalidad principal. En este caso, la mayor parte de las nuevas iniciativas de desarrollo pueden destinarse a crear nuevos microservicios que simplemente se conectan al

monolito a través de las API. Si el monolito no se puede mantener ni descartar, pero algunas de sus partes deben reescribirse, el patrón estrangulador es el mejor enfoque.

Con el **patrón estrangulador**, los equipos de desarrollo crean una funcionalidad que ya está bastante desacoplada del monolito. Esta funcionalidad está desacoplada del monolito detrás de una API para facilitar el reemplazo y el retiro una vez que se crea el microservicio. Esto significa que las capacidades que no requieren cambios en muchas aplicaciones orientadas al cliente y que potencialmente no necesitan su propio almacén de datos son las candidatas ideales que debe considerar primero. En una aplicación de comercio electrónico, por ejemplo, algunos servicios potenciales que debe considerar son

la autenticación, la facturación o los perfiles de clientes. Al crear sus primeros microservicios, la mayoría de los equipos tienen como objetivo probar y optimizar las canalizaciones de entrega de software, los enfoques de las API y capacitar a los miembros del equipo, en lugar de optimizar la funcionalidad.

Visítenos para obtener más información sobre el potencial de los microservicios en su empresa.