

ACADEMIA JAVA

P1- MONOLITICOS VS MICROSERVICIOS - XIDERAL: SEMANA 2

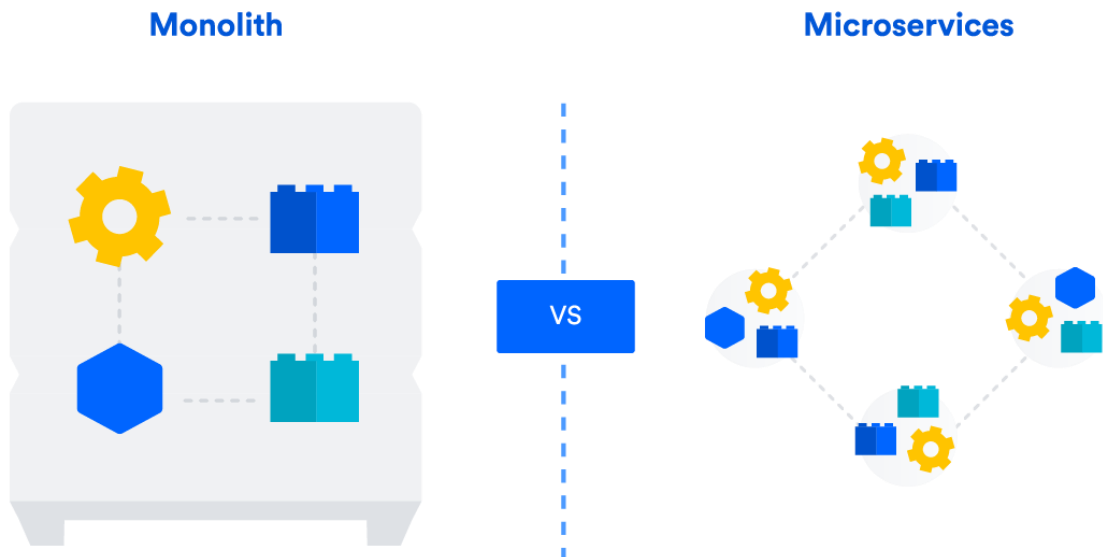
NOMBRE: JOSE MANUEL JIMENEZ HDZ

¿Qué es una aplicación monolítica?

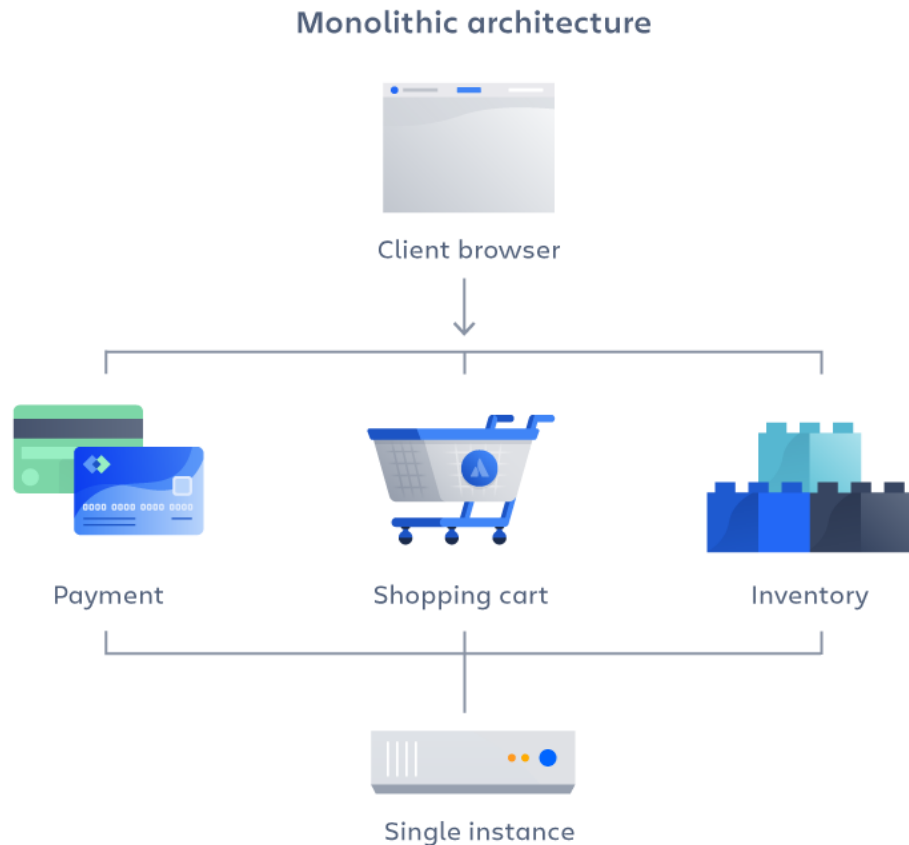
Una aplicación monolítica se compila como una sola unidad unificada.

¿Qué es un microservicio?

Es un conjunto de pequeños servicios que están listos siempre para ejecutarse, pero cada uno de forma independiente y según se requieren



MONOLITICO:



Ventajas de una arquitectura monolítica:

Estas son algunas 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:

Estas son algunas 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.

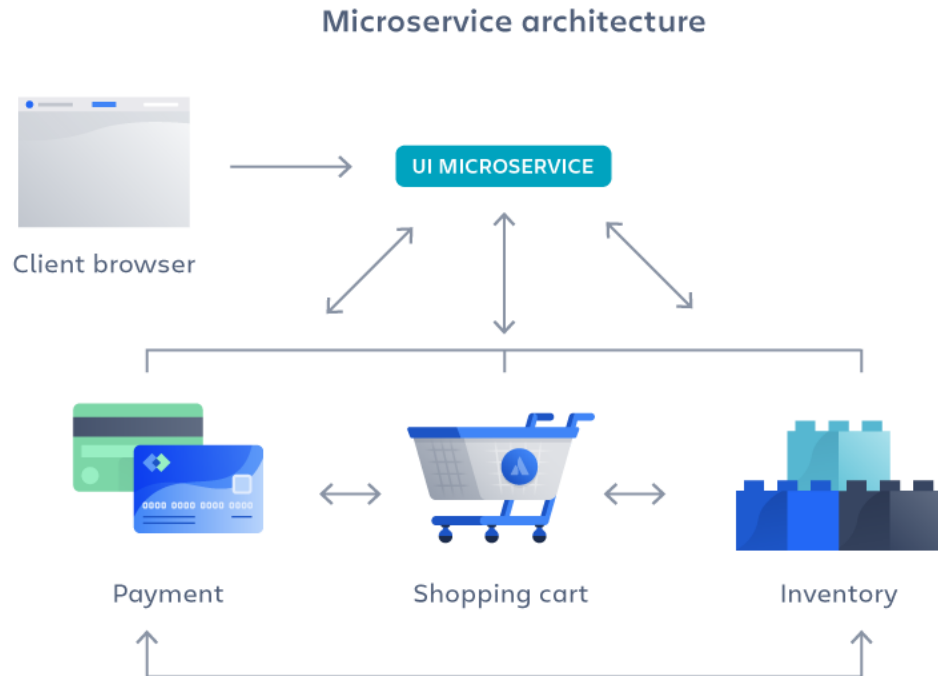
Fiabilidad: si hay un error en algún módulo, puede afectar a la disponibilidad de toda la aplicación.

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.

MICROSERVICIOS:



Ventajas de los microservicios:

En pocas palabras, estas son las ventajas de los microservicios:

Agilidad: promueve formas ágiles de trabajar con equipos pequeños que implementen con frecuencia.

Escalado flexible: si un microservicio está alcanzando su capacidad de carga, se pueden implementar rápidamente nuevas instancias de ese servicio al clúster que lo acompaña para aliviar la presión. Ahora tenemos varios inquilinos y no tenemos control de estado, con clientes repartidos en varias instancias. Así, podemos respaldar tamaños de instancias mucho mayores.

Implementación continua: tenemos ciclos de lanzamiento frecuentes y más rápidos. Antes enviábamos actualizaciones una vez a la semana y ahora podemos hacerlo dos o tres veces al día.

Muy fácil de mantener y probar: los equipos pueden hacer pruebas con el código y dar marcha atrás si algo no funciona como esperan. Esto facilita la actualización del código y acelera el tiempo de salida al mercado de nuevas funciones. Además, es fácil aislar y corregir fallos y errores en los servicios individuales.

Implementación independiente: los microservicios son unidades individuales, por lo que permiten una implementación independiente rápida y sencilla de funciones individuales.

Flexibilidad tecnológica: con las arquitecturas de microservicios, los equipos pueden elegir con libertad las herramientas que desean.

Alta fiabilidad: puedes implementar cambios para un servicio en concreto sin el riesgo de que se caiga toda la aplicación.

Equipos más satisfechos: los equipos que trabajan con microservicios están mucho más satisfechos, ya que son más autónomos y pueden compilar e implementar de forma independiente, sin esperar semanas a que se apruebe una solicitud de incorporación de cambios.

Desventajas de los microservicios:

Estas son algunas desventajas de los microservicios:

Desarrollo descontrolado: los microservicios suman complejidad en comparación con las arquitecturas monolíticas, ya que hay más servicios en más lugares creados por varios equipos. Si el desarrollo descontrolado no se gestiona adecuadamente, se reduce la velocidad de desarrollo y el rendimiento operativo.

Costes exponenciales de infraestructura: cada nuevo microservicio puede tener su propio coste para el conjunto de pruebas, los manuales de estrategias de desarrollo, la infraestructura de alojamiento, las herramientas de supervisión, etc.

Más sobrecarga organizativa: los equipos deben agregar otro nivel de comunicación y colaboración para coordinar las actualizaciones e interfaces.

Desafíos para la depuración: cada microservicio tiene su propio conjunto de registros, lo que complica la depuración. Además, un solo proceso empresarial puede ejecutarse en varias máquinas, lo que aún lo dificulta más.

Falta de estandarización: sin una plataforma común, puede haber una proliferación de idiomas, de estándares de registro y de supervisión.

La propiedad no está clara: a medida que se introducen más servicios, también lo hace el número de equipos que los ejecutan. Con el tiempo, se hace difícil conocer los servicios disponibles que un equipo puede aprovechar y con quién hay que hablar para obtener asistencia.