



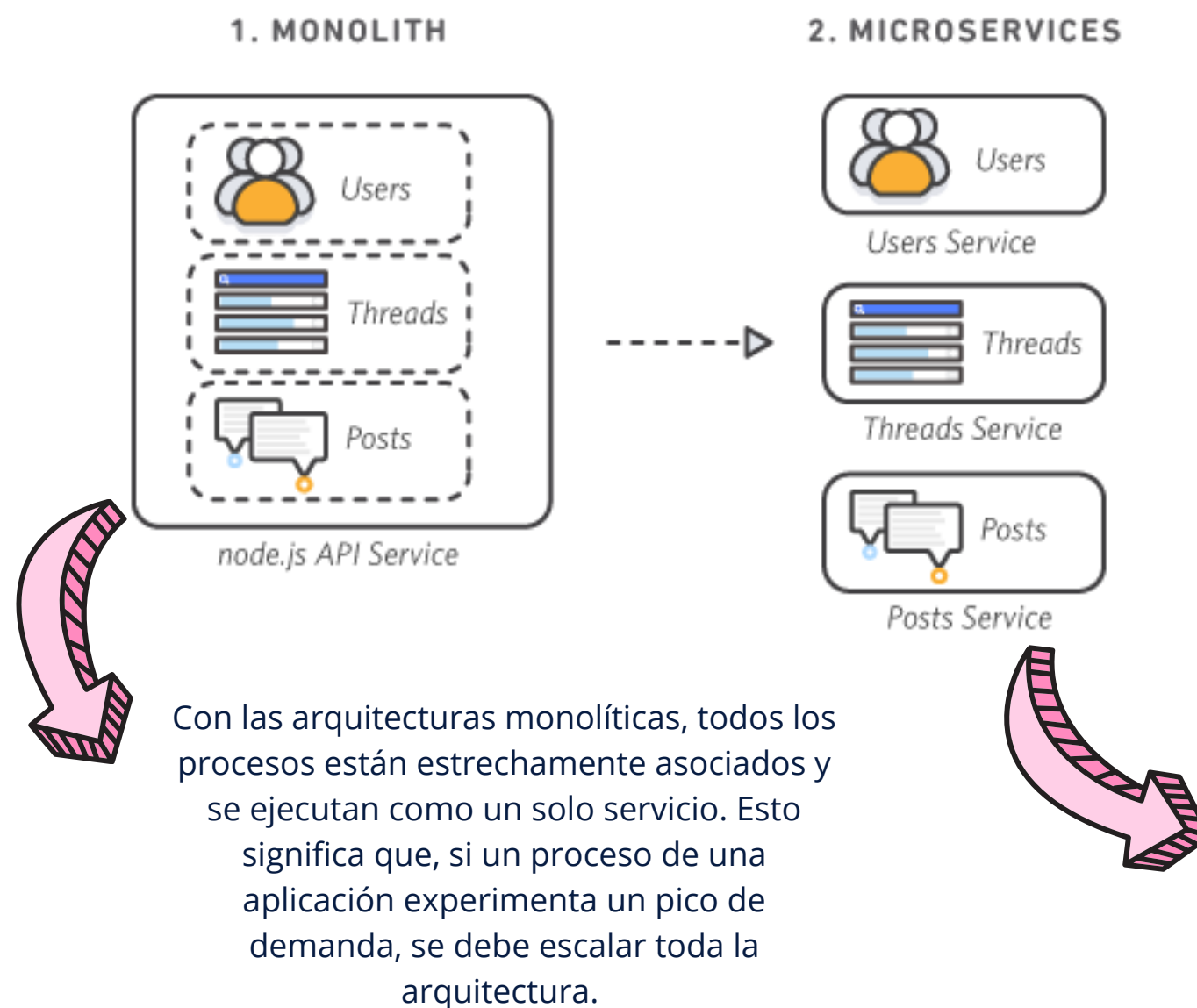
Microservicios

Presentado por: Guadalupe López Velázquez



Microservicios

Arquitectura monolítica en comparación con la arquitectura de microservicios



Los microservicios son un enfoque arquitectónico y organizativo para el desarrollo de software donde el software está compuesto por pequeños servicios independientes que se comunican a través de API bien definidas. Los propietarios de estos servicios son equipos pequeños independientes.

Las arquitecturas de microservicios hacen que las aplicaciones sean más fáciles de escalar y más rápidas de desarrollar. Esto permite la innovación y acelera el tiempo de comercialización de las nuevas características.

Características de los microservicios

Autónomos

Cada servicio componente en una arquitectura de microservicios se puede desarrollar, implementar, operar y escalar sin afectar el funcionamiento de otros servicios.

Los servicios no necesitan compartir ninguno de sus códigos o implementaciones con otros servicios.

Especializados

Cada servicio está diseñado para un conjunto de capacidades y se enfoca en resolver un problema específico.

Si los desarrolladores aportan más código a un servicio a lo largo del tiempo y el servicio se vuelve complejo, se puede dividir en servicios más pequeños.

Beneficios de los microservicios



Agilidad

Fomentan una organización de equipos pequeños e independientes que se apropian de los servicios.

Los equipos actúan en un contexto pequeño y bien comprendido, y están facultados para trabajar de forma más independiente y más rápida. Esto acorta los tiempos del ciclo de desarrollo.

Implementación sencilla

permiten la integración y la entrega continuas, lo que facilita probar nuevas ideas y revertirlas si algo no funciona.

El bajo costo de los errores permite experimentar, facilita la actualización del código y acelera el tiempo de comercialización de las nuevas características.

Código reutilizable

La división del software en módulos pequeños y bien definidos les permite a los equipos usar funciones para diferentes propósitos. Un servicio escrito para una determinada función se puede usar como un componente básico para otra característica. Esto permite que una aplicación arranque por sí sola, ya que los desarrolladores pueden crear nuevas capacidades sin tener que escribir código desde cero.

Resistencia

La independencia del servicio aumenta la resistencia de una aplicación a los errores. En una arquitectura monolítica, un error en un solo componente, puede provocar un error en toda la aplicación. Con los microservicios, si hay un error en todo el servicio, las aplicaciones lo manejan degradando la funcionalidad sin bloquear toda la aplicación.