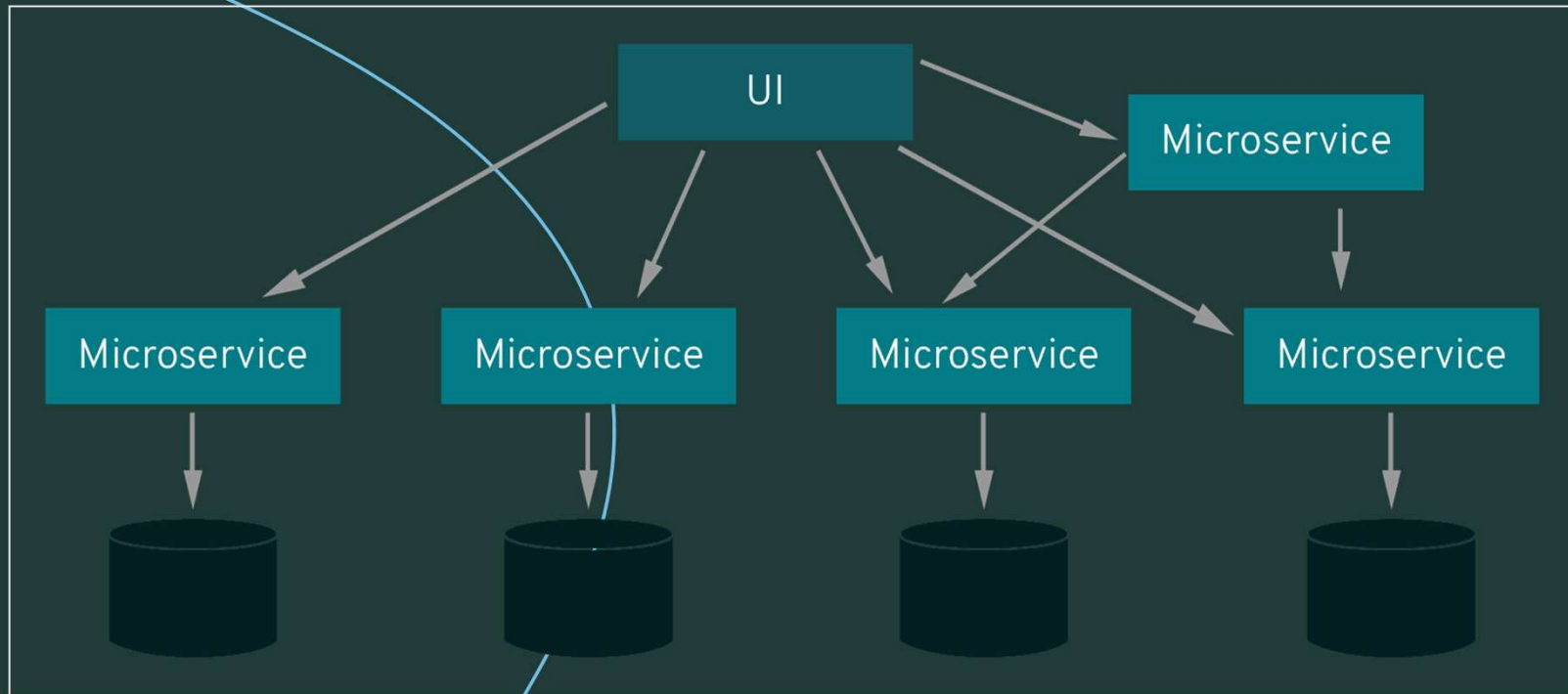


# MICROSERVICIOS



Alberto Fernández Gutiérrez

Christian Díaz González

# CONTENIDOS

- Introducción a los microservicios
- Ventajas e inconvenientes de su uso
- Desmontando mitos
- Cuando usar microservicios
- Alternativas a los microservicios
- Conclusiones
- Bibliografía

# INTRODUCCIÓN A LOS MICROSERVICIOS

## ¿Qué son los microservicios?

- ❑ Son tanto un estilo de arquitectura como una forma de programar software
- ❑ Dividen las aplicaciones en elementos pequeños e independientes entre sí
- ❑ Cada microproceso es detallado y sencillo

## ¿Por qué surgen?

- ❑ Necesidad de escalar servicios IT a un gran tamaño
- ❑ Para poder competir en mercados muy competitivos
- ❑ Se quiere un sistema de alta calidad y tiempo de respuesta corto

# VENTAJAS E INCONVENIENTES



## □ Ventajas

- Límites de módulos fuertes
- Implementación independiente
- Diversidad tecnológica

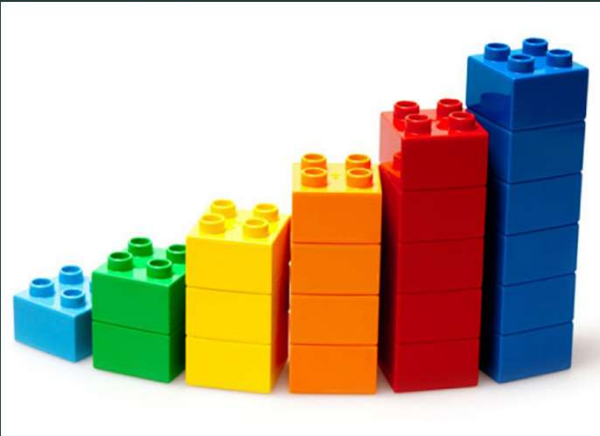
## □ Desventajas

- Distribución
- Consistencia final
- Complejidad operativa

# DESMONTANDO MITOS



# ESCALABILIDAD



- ❑ ¿Microservicios imprescindibles?
- ❑ Pocos servidores pueden atender a muchos usuarios
- ❑ Pocas aplicaciones necesitan microservicios para lograr una buena escalabilidad

# SIMPLICIDAD



- ❑ ¿Los microservicios conducen a soluciones más simples?
- ❑ Principio de conservación de la complejidad esencial
- ❑ En realidad, los microservicios nos lo ponen difícil:
  - Técnica de diseño
  - Observabilidad de errores
  - Aumenta el nivel infraestructura
- ❑ Conducen a soluciones complejas
- ❑ Lo que se necesita es modularidad

# MEJOR DISEÑO



- ❑ La estructura del código es independiente, se utilicen microservicios o no
- ❑ Microservicios no son necesarios para conseguir una buena estructuración de código
- ❑ Los sistemas monolíticos no están bien estructurados debido a la ausencia de:
  - Disciplina
  - Conocimiento
  - Experiencia
- ❑ Gran nivel de diseño para usar microservicios



# FACILITA EL CAMBIO DE TECNOLOGÍA (I)



- ❑ Facilita el cambio si sólo se reescribe una sola unidad.
- ❑ Permite migrar poco a poco, controlando así los riesgos.
- ❑ Las actualizaciones de tecnologías no son muy comunes.
- ❑ Impedimentos del cambio de tecnologías:
  - Procesos de gobernanza IT
  - Dependencias de la app en el nivel del SO

# FACILITA EL CAMBIO DE TECNOLOGÍA (II)



## ❑ PROCESOS DE GOBERNANZA IT

- Aplicaciones software Vs Bienes físicos
- Costos prelanzamiento y postlanzamiento
- No hay que migrar el problema, sino tratarlo de raíz

## ❑ APLICACIÓN DEPENDE DEL SO

- Todas las aplicaciones tienen que poder manejar el nuevo SO
- El nuevo SO debe proporcionar todo lo que la aplicación necesite
- Alto acoplamiento con el SO solucionado gracias a la tecnología de contenedores



# ¿CÚANDO Y POR QUÉ USAR MICROSERVICIOS?

- ❑ Necesitas moverte rápido
  - Adecuado para hiperescaladores con negocio IT
  - No se necesita coordinación entre los diferentes equipos de desarrollo
  - Estilo arquitectónico muy exigente y necesita de una actuación rápida
- ❑ Existen NFRs muy dispares
- ❑ Crecimiento exponencial
- ❑ Necesitas de ciclos de reloj muy cortos
- ❑ Cuando necesitas muchos desarrolladores para avanzar rápidamente



# ALTERNATIVAS A LOS MICROSERVICIOS (I)

## ➤ Moduliths

- ❑ Sistema monolítico que se basa en realizar módulos claramente definidos y aislados
- ❑ Requiere de una buena capacidad de diseño
- ❑ Hay que tener mucho cuidado de no violar la encapsulación
- ❑ Difícil de implementar en un sistema monolítico con un mal diseño funcional
- ❑ Necesitan de mucha organización entre los integrantes del equipo

# ALTERNATIVAS A LOS MICROSERVICIOS (II)

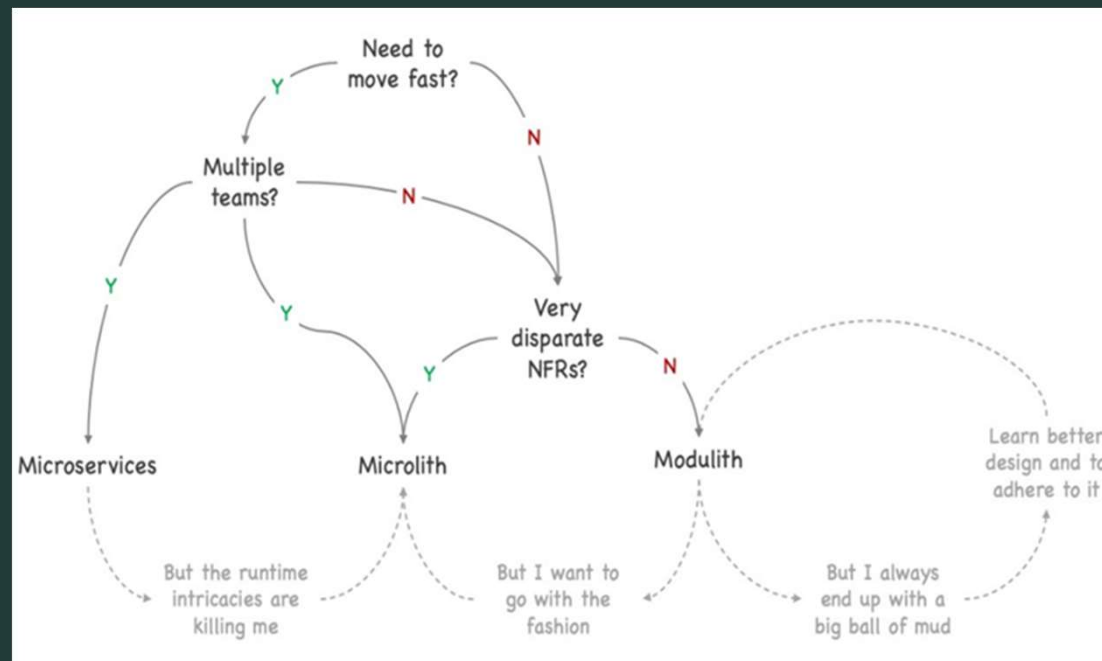
## ➤ Microliths

- ❑ Usa los principios de diseño de módulos independientes para evitar llamadas entre módulos
- ❑ Utilizado cuando se necesitan tiempos de ejecución bajos y los NFRs son muy dispares
- ❑ Empleados en el caso de que se puedan implementar microservicios, pero no se esté dispuesto a llevarlos a cabo
- ❑ Permite dividir una aplicación monolítica, librando los problemas de los sistemas distribuidos
- ❑ Requiere de un diseño de independencia mutua
- ❑ Caso de uso = Microlith

# ALTERNATIVAS A LOS MICROSERVICIOS (III)

## Microliths VS Moduliths

- ❑ Microliths hacen que los cambios de tecnología sean más sencillos.
- ❑ Debido al tamaño que tienen hace que la migración se pueda hacer microlith a microlith



# CONCLUSIONES

- ❑ Los microservicios merecen la pena cuando estamos en un marco con alta competitividad y con gran necesidad de adaptación
- ❑ A nivel usuario o cualquier aplicación que no cumpla dichas características no es recomendable el uso de microservicios
- ❑ No existe una fórmula secreta de cuando usar o no microservicios
- ❑ Equipo de desarrollo pequeño = No microservicios



# BIBLIOGRAFÍA

- ❖ <https://martinfowler.com/articles/microservice-trade-offs.html>
- ❖ [https://www.ufried.com/blog/microservices\\_fallacy\\_1/](https://www.ufried.com/blog/microservices_fallacy_1/)
- ❖ [https://ufried.com/blog/microservices\\_fallacy\\_2\\_scalability/](https://ufried.com/blog/microservices_fallacy_2_scalability/)
- ❖ [https://ufried.com/blog/microservices\\_fallacy\\_3\\_simplicity/](https://ufried.com/blog/microservices_fallacy_3_simplicity/)
- ❖ [https://ufried.com/blog/microservices\\_fallacy\\_4\\_reusability\\_autonomy/](https://ufried.com/blog/microservices_fallacy_4_reusability_autonomy/)
- ❖ [https://ufried.com/blog/microservices\\_fallacy\\_5\\_design/](https://ufried.com/blog/microservices_fallacy_5_design/)
- ❖ [https://ufried.com/blog/microservices\\_fallacy\\_6\\_technology/](https://ufried.com/blog/microservices_fallacy_6_technology/)
- ❖ [https://ufried.com/blog/microservices\\_fallacy\\_7\\_actual\\_reasons/](https://ufried.com/blog/microservices_fallacy_7_actual_reasons/)
- ❖ [https://ufried.com/blog/microservices\\_fallacy\\_8\\_choices/](https://ufried.com/blog/microservices_fallacy_8_choices/)
- ❖ [https://ufried.com/blog/microservices\\_fallacy\\_9\\_moduliths/](https://ufried.com/blog/microservices_fallacy_9_moduliths/)
- ❖ [https://www.ufried.com/blog/microservices\\_fallacy\\_10\\_microliths/](https://www.ufried.com/blog/microservices_fallacy_10_microliths/)



# PREGUNTAS

