



Microservicios

Conceptos básicos

Programación II
Grado en Business Analytics



¿Qué es un Microservicio?

Es un patrón arquitectónico, en el que cada componente del sistema se ejecuta como una aplicación independiente que interactúa con otras.



¿Qué ventajas presentan?

- 1. Facilita la escalabilidad de la aplicación**
- 2. Encapsula funcionalidades**
- 3. Encapsula entorno**
- 4. Por tanto, el mantenimiento funcional es más fácil**



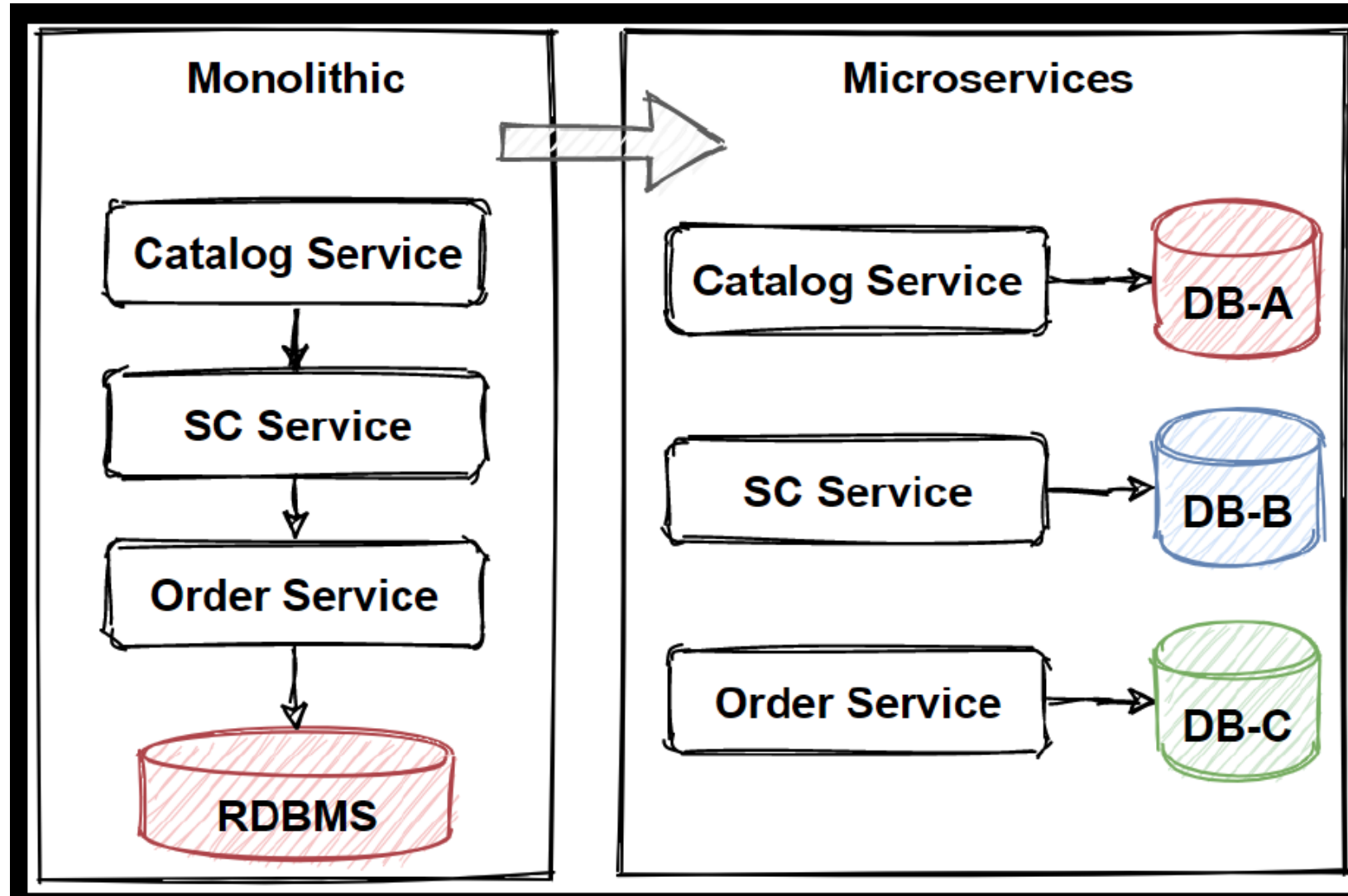
¿Qué desventajas tiene?

- 1. Complica el desarrollo**
- 2. Complica las pruebas: no sólo funcionalidad, sino integración**



Monolito vs microservicio

Monolito vs microservicio



En una app monolítica, se encapsula toda la lógica de negocio en un mismo ejecutable (desplegable), con independencia de que esta lógica siga patrones como el MVC o arquitectura en capas.

API REST

Un sistema necesita comunicarse con otra a través de algún mecanismo común.



Monolito vs microservicio

Ese mecanismo puede ser un API (Application Programming Interface). Un API permite la comunicación vía programática entre varias aplicaciones: es un contrato que deben seguir las aplicaciones que utilicen ese API para comunicarse entre ellas.

Un API REST (RESTful) debe cumplir varios criterios para considerarse como tal. Los más relevantes, son los siguientes:

1. Arquitectura cliente (consumidores de la información) servidor (quien da o genera la información) de la aplicación
2. Cliente y servidor totalmente desacoplada
3. Pueden existir varias capas: la comunicación entre cliente y servidor no tiene por qué ser directa
4. Comunicación asíncrona
5. No se guarda el estado de las comunicaciones, pero se puede guardar el mensaje en memoria caché
6. Estandarización de las comunicaciones
7. Recursos identificables e independientes de la representación del cliente
8. El cliente debe poder modificar los recursos a través de la representación que recibe
9. Los mensajes deben identificar cómo manipular las operaciones



Monolito vs microservicio

Un API REST (RESTful) permite la comunicación entre aplicaciones (para nosotros, a partir de ahora, servicios) a través del protocolo HTTP.

Operaciones CRUD: Create Read Update Delete:

1. Solicitud HTTP GET: solicitar recursos
2. Solicitud HTTP POST: enviar recursos
3. Solicitud HTTP PUT: actualizar recursos
4. Solicitud HTTP DELETE: borrar recursos.



Operaciones CRUD: Create Read Update Delete a través de HTTP

1. Solicitud `HTTP GET`: solicitar recursos
2. Solicitud `HTTP POST`: enviar recursos
3. Solicitud `HTTP PUT`: actualizar recursos
4. Solicitud `HTTP DELETE`: borrar recursos.

El mensaje enviado por un API REST puede ir, entre otros, en los siguientes formatos:

1. JSON
2. HTML
3. XML
4. Texto Plano
5.



Microservicios

Conceptos básicos

Programación II
Grado en Business Analytics