códigofacilito

Base de datos para Microservicios

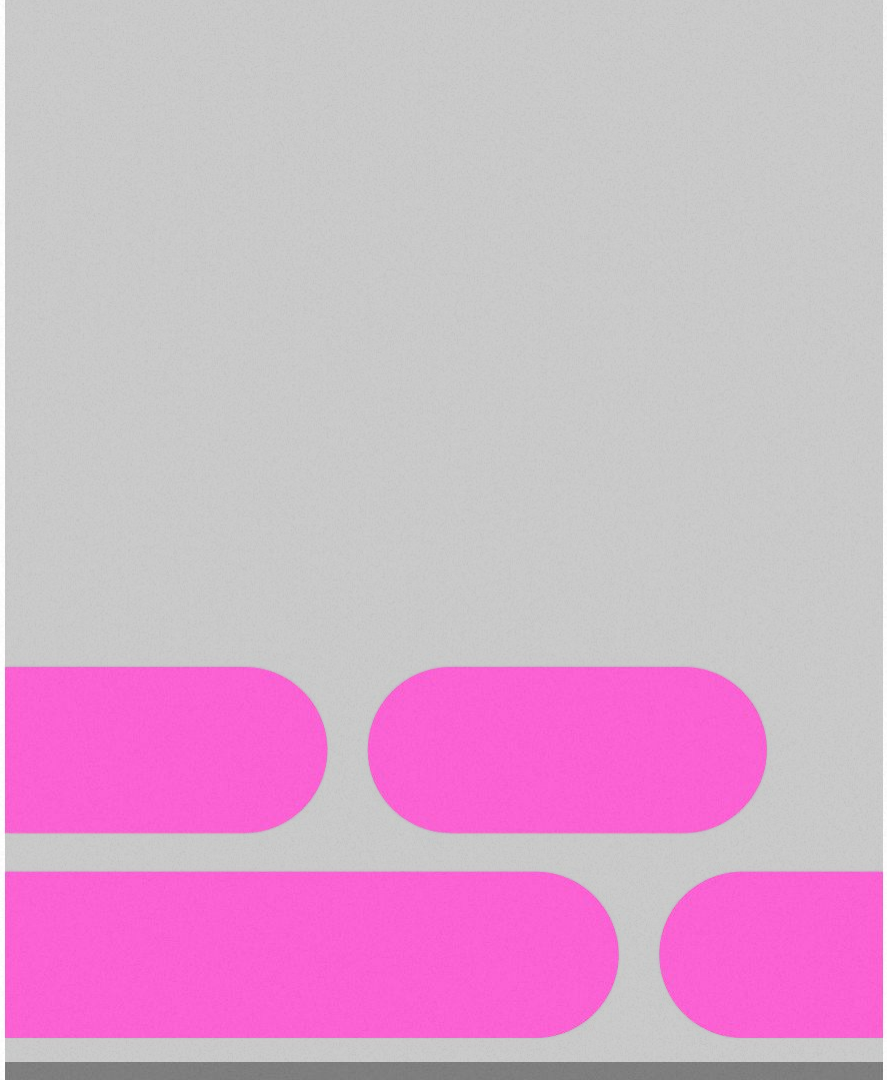
Bootcamp - Backend Avanzado

Eduardo I. García Pérez.

Objetivos.

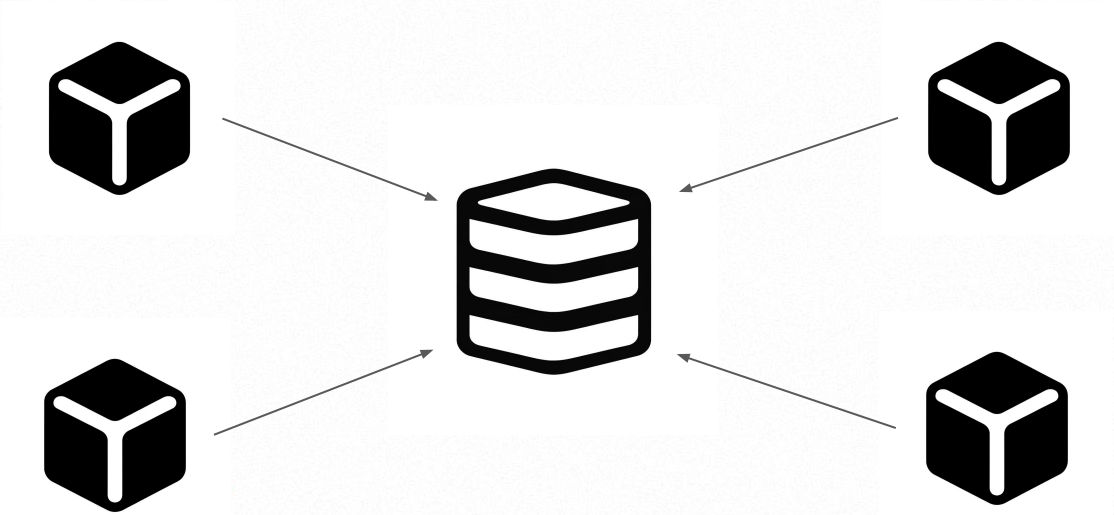
- Database Shared.
- Database per service.

Database shared.



Una sola instancia de base de datos. ⛰️

Microservicios utilizan una misma base de datos.



Configuración de nuestros servicios.

- Host (123.456.7.89)
- Username cody
- Password password123
- Port 3306
- Timeout 22000*
- Database Base de datos

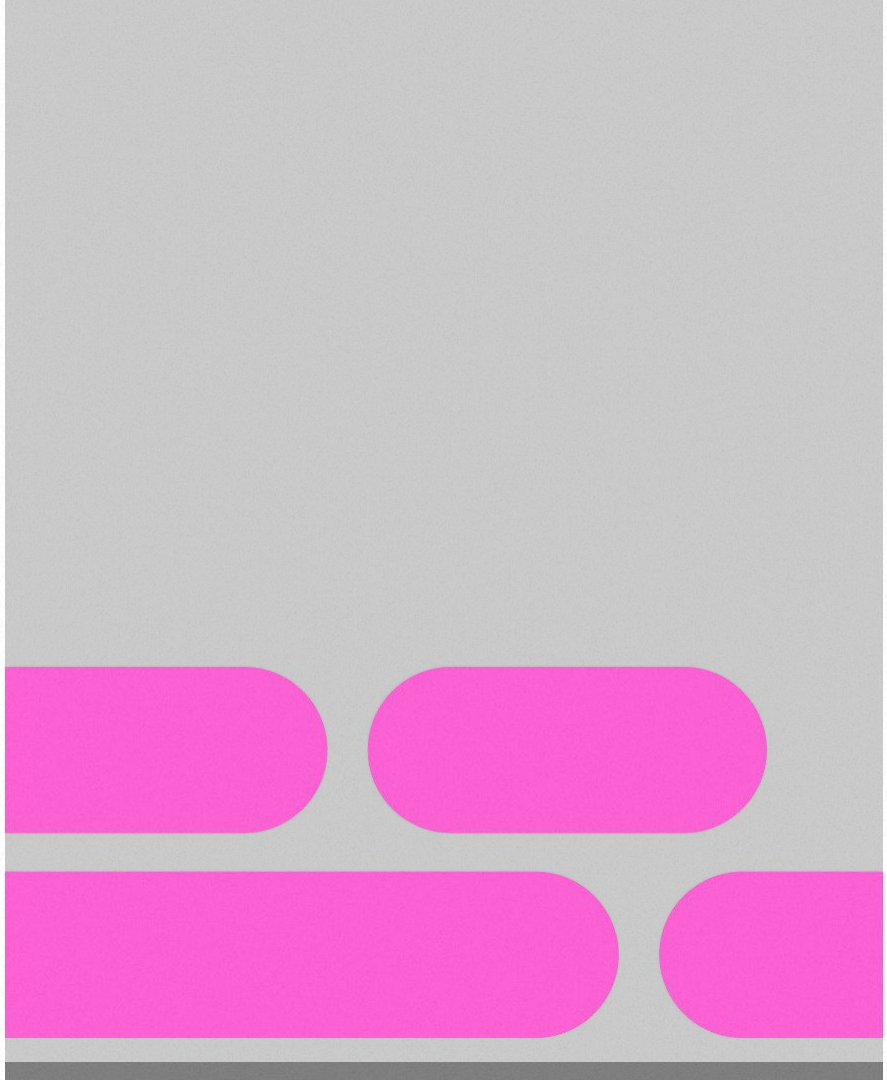
'postgresql://username:password@localhost/dbname'

Pros. ✅

- Perfecto para un presupuesto bajo.
- Fácil de administrar.
- Fácil de configurar.
- Sin inconsistencia de datos.
- Menor riesgo en la seguridad.

Cons. ❌

- Una solo gestor de base de datos (MySQL, MongoDB, etc..)
- Posibles cuellos de botellas.
- Violación del principio de microservicios.



Database per service.

Múltiples base de datos.

Cada microservicio posee su propia base de datos.

- Diferentes gestores de base de datos.
- Diferentes servidores.

Cada microservicio posee y maneja su propia información.

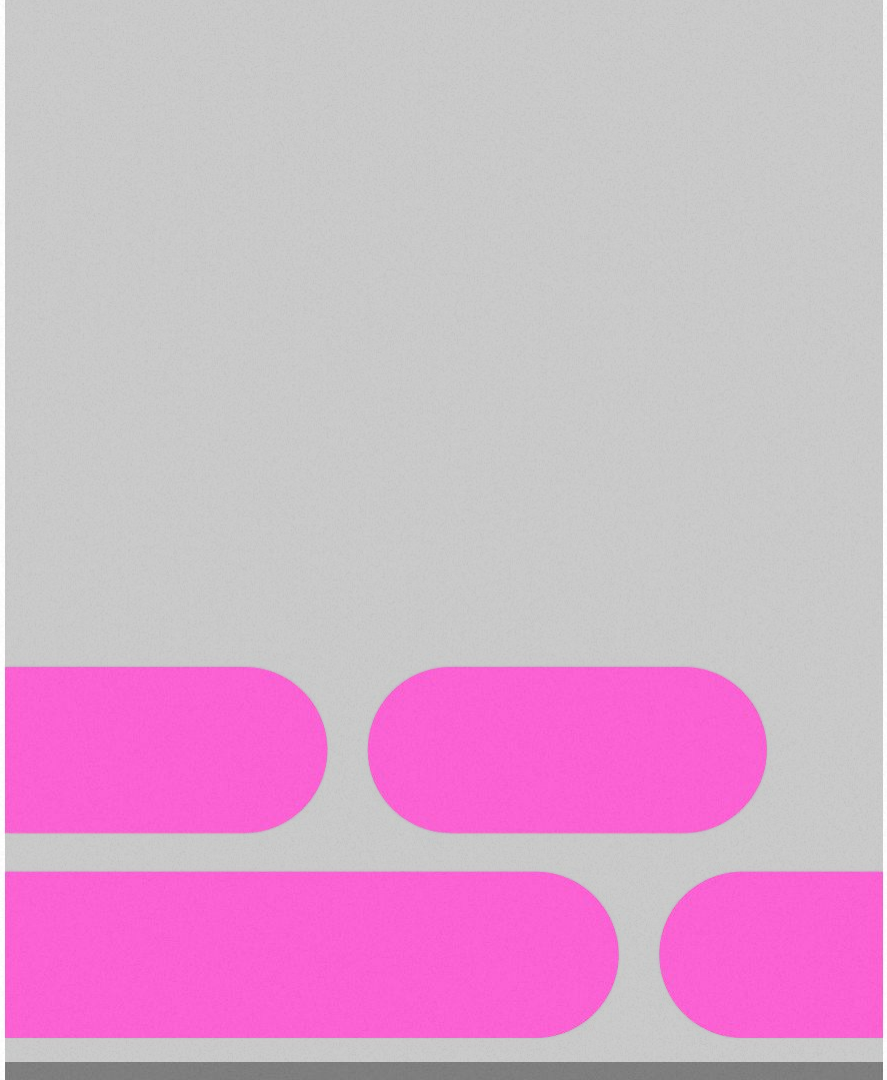
Pros. ✅

- Desacoplamiento de datos.
- Fácil escalabilidad.
- Optimización de datos.

Cons. ❌

- Duplicación de datos.
- Complejidad en transacciones.
- Consistencia de datos.
- Complejidad operacional.

Saga pattern.



Saga Pattern.

- El patrón Saga permite realizar una secuencia de transacciones locales para cada microservicio de forma grupal.

- Saga se asegura que, cuando múltiples microservicios participen en una transacción el resultado final sea consistente.

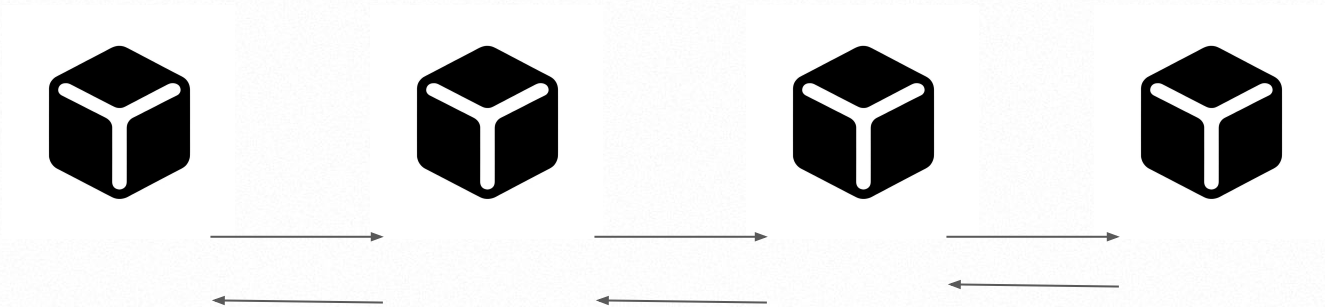
Implementación del patrón saga.

- **Choreography:** Cada microservicio produce y escucha eventos de otros microservicios. Cuando un microservicio finaliza una transacción este emite un evento.
    - No existe entidad central que controle los eventos.

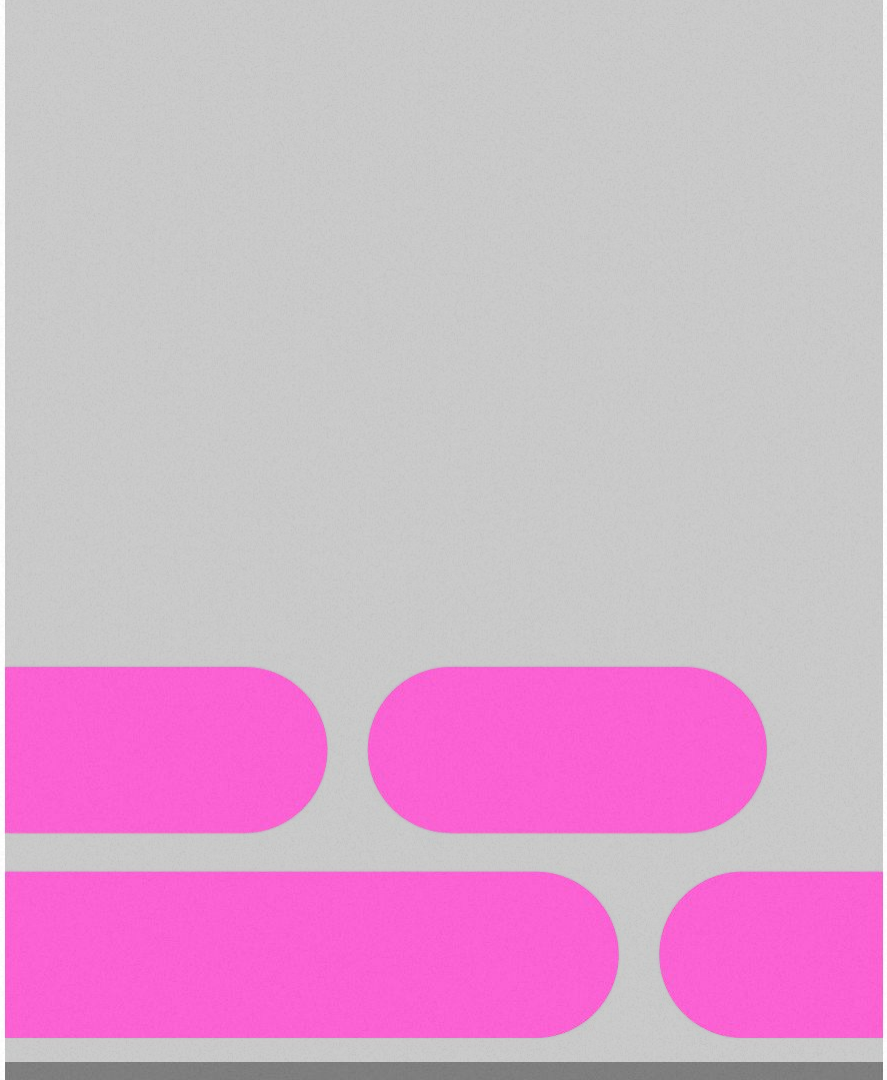
Implementación del patrón saga.

- **Orchestration:** Este approach utiliza un orquestador (Un servicio central) que indica a los participantes que transacción debe ejecutarse.

Implementación del patrón saga.



Ejemplo.



Conexión entre microservicios.

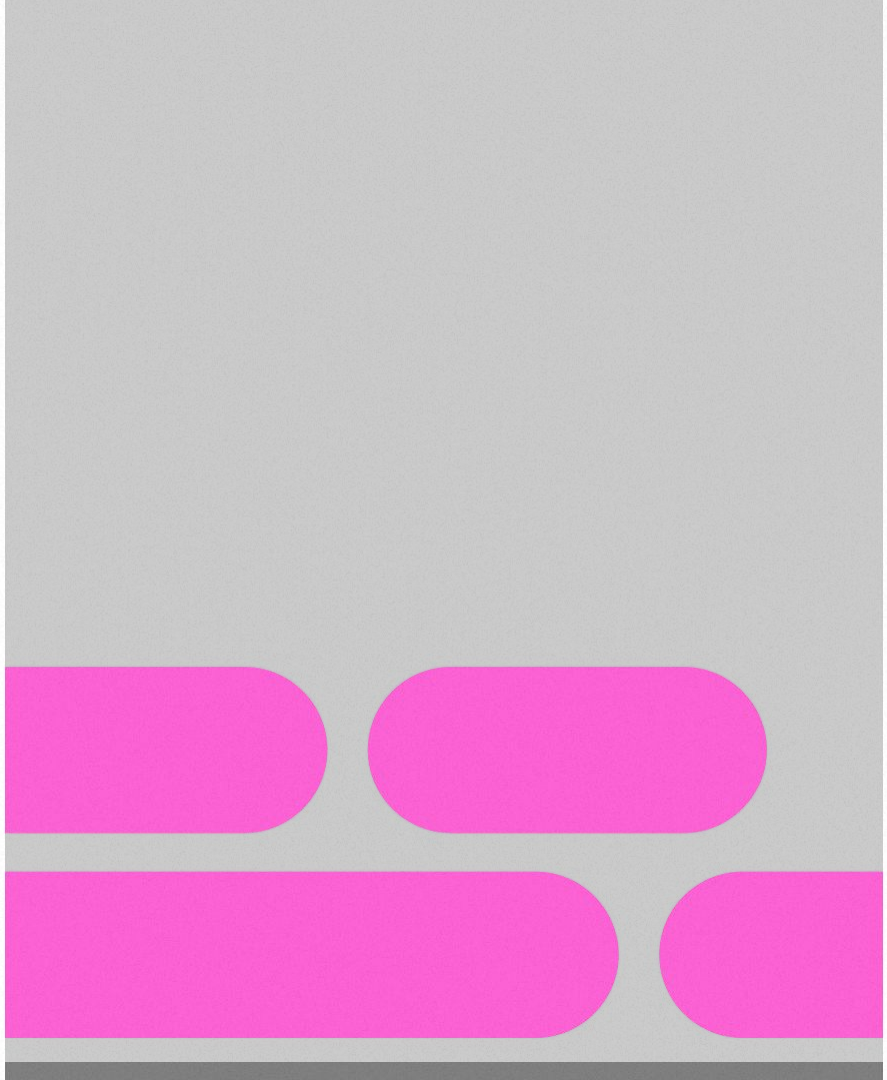
- Servicio Principal
- Servicio de Cursos.
- Servicio de mensajería.

https://github.com/codigofacilito/api_call

Herramientas a utilizar.

- Django.
- requests

Ejercicio práctico.



Añadir seguridad en la comunicación.

Solo servicios autorizados pueden hacer el llamado de los Endpoints.

códigofacilito

Base de datos para Microservicios

Bootcamp - Backend Avanzado

Eduardo I. García Pérez.