



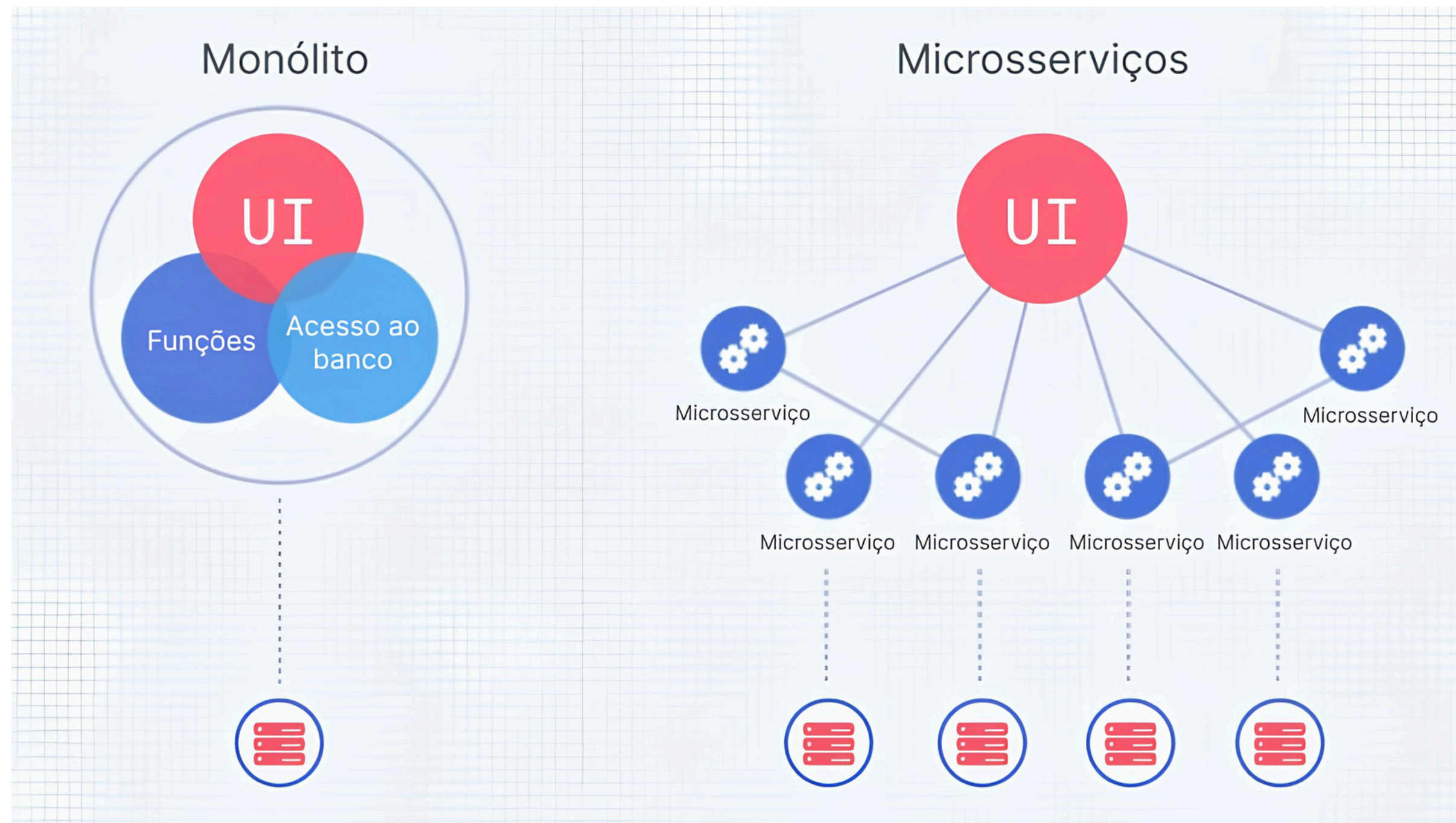
MICROSERVIÇOS



Aluno: Kelvin Santos Almeida Lima

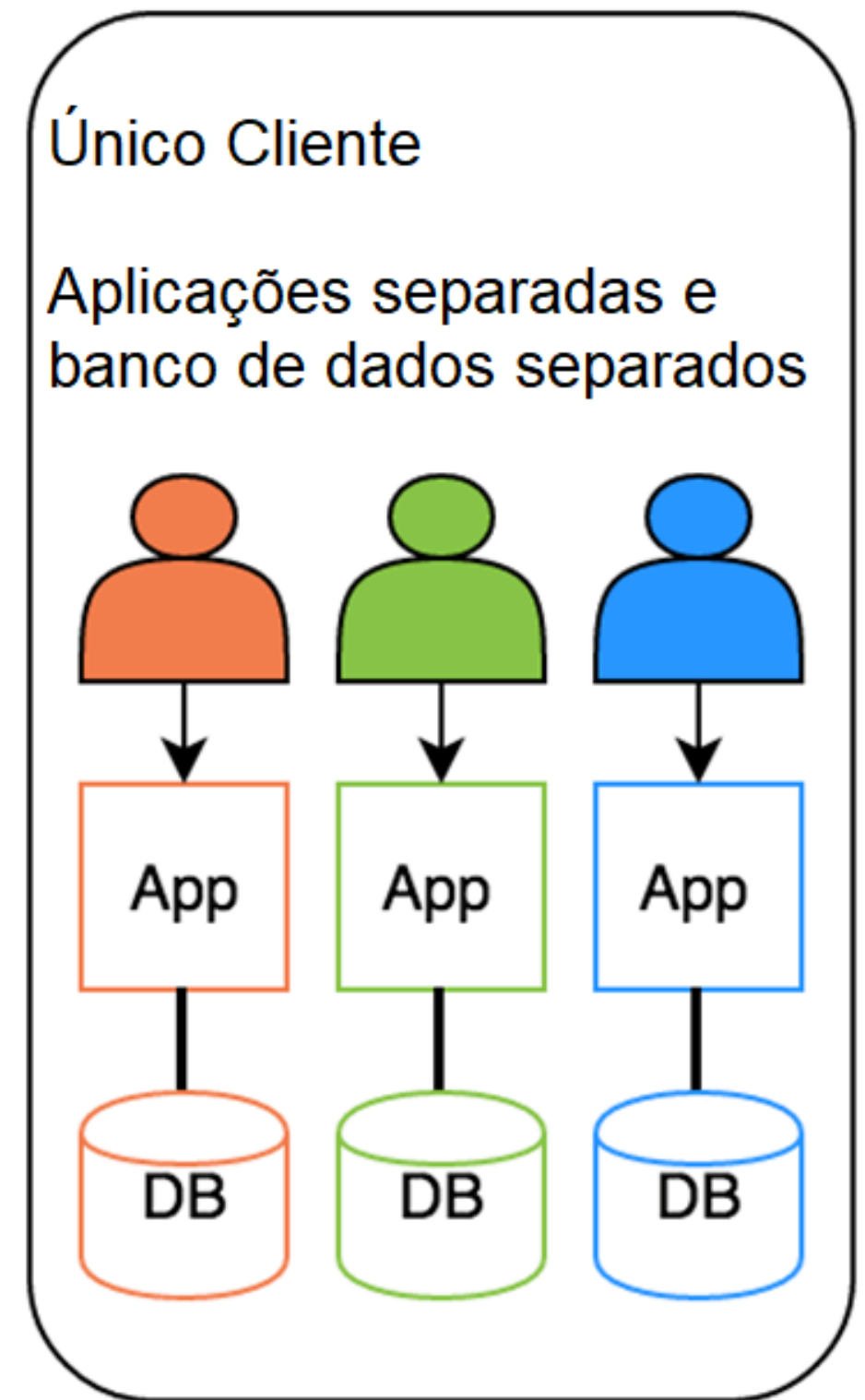
"Microserviços: Desconstruindo o Monólito"

"Serviços pequenos e autônomos trabalhando juntos"



O Que São Microserviços?

- Definição:
 - Serviços independentes e fracamente acoplados
 - Cada serviço gerencia seus dados e regras
- Características:
 - Alinhados a capacidades de negócio
 - Banco de dados dedicado por serviço
 - Comunicação leve (APIs/eventos)

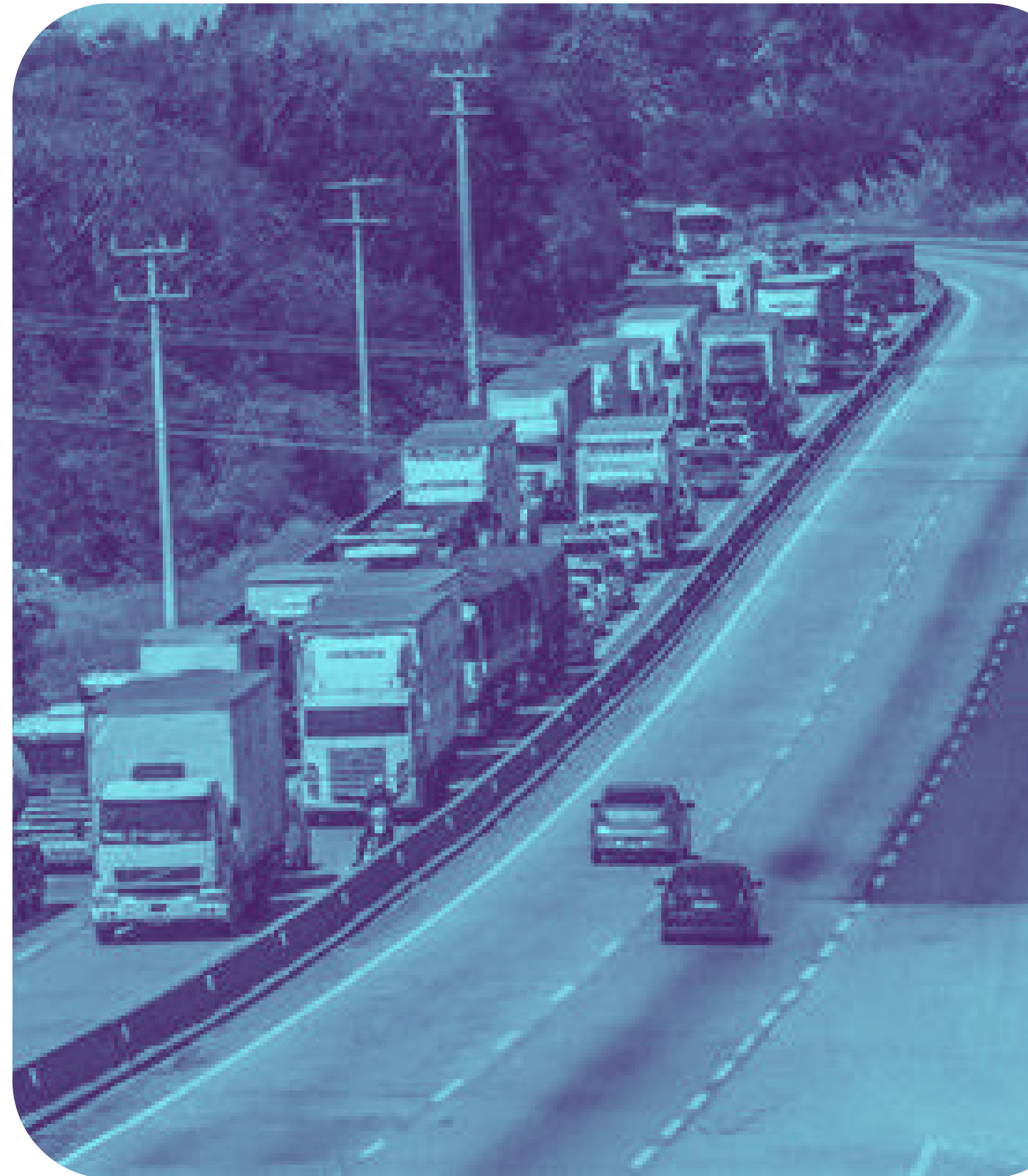


Por Que Microserviços?

Problemas dos Monolitos:

- Dificuldade para escalar
- Implantações lentas
- Ponto único de falha

- Vantagens dos Microserviços:
 - Escala por componente
 - Iterações rápidas
 - Isolamento de falhas



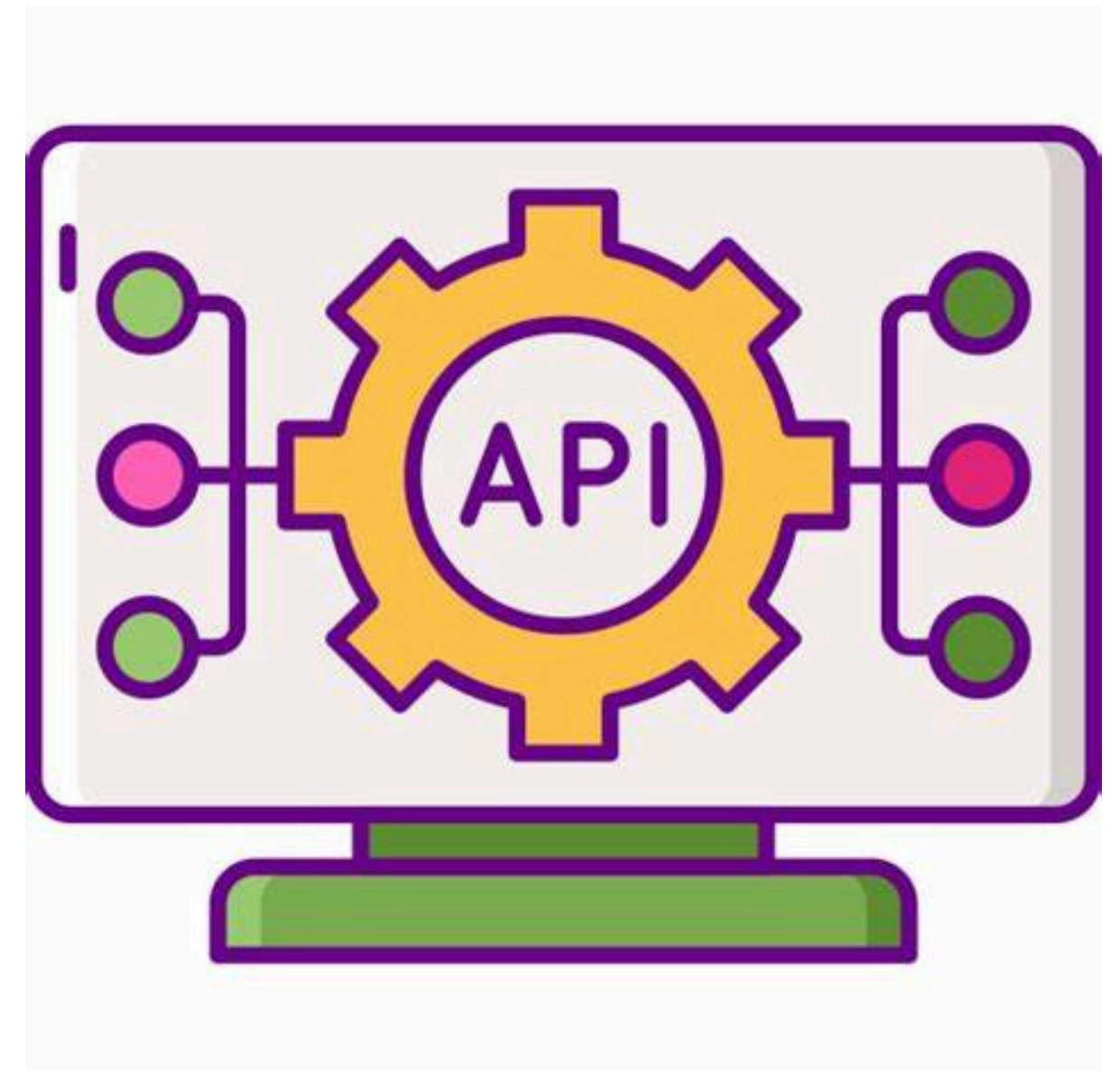
Princípios-Chave

1. **Responsabilidade Única:** Uma função por serviço
2. **Autonomia**
3. **Descentralização:**
 - Times são donos do ciclo completo
 - Bancos de dados heterogêneos (ex.: SQL + NoSQL)



Comunicação Entre Serviços

- **Síncrona:** REST/...
- **Assíncrona:** Eventos (Kafka, RabbitMQ)
- **Regra de Ouro:**
 - Nunca acessar o banco de dados de outro serviço diretamente



Implantação e Escalabilidade

- **Contêineres:** Isolamento (Docker)
- **Orquestração:** Escala automática (Kubernetes)
- **Cultura DevOps:** "Quem desenvolve, opera"



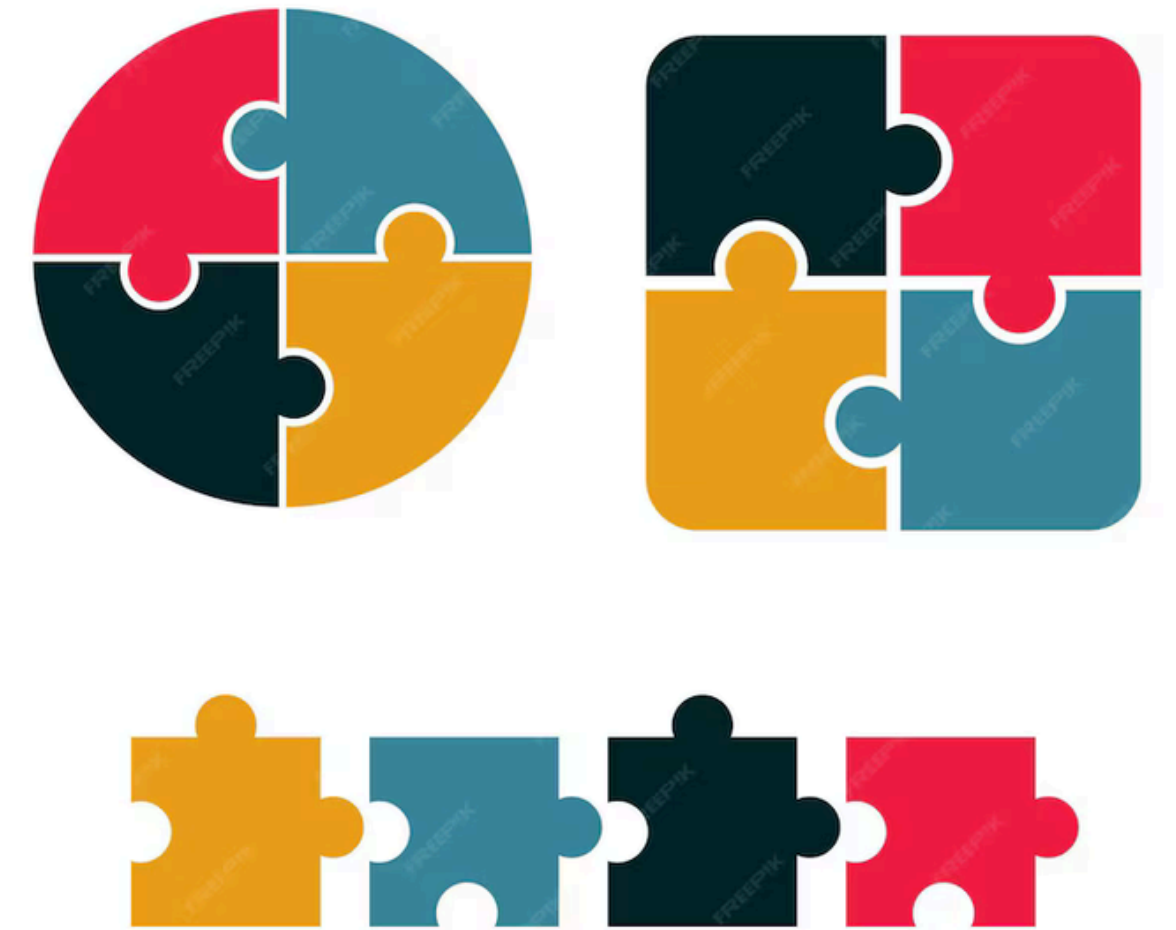
Desafios

1. Complexidade:

- a. + Microserviços + Complexo
- b. Latência de rede

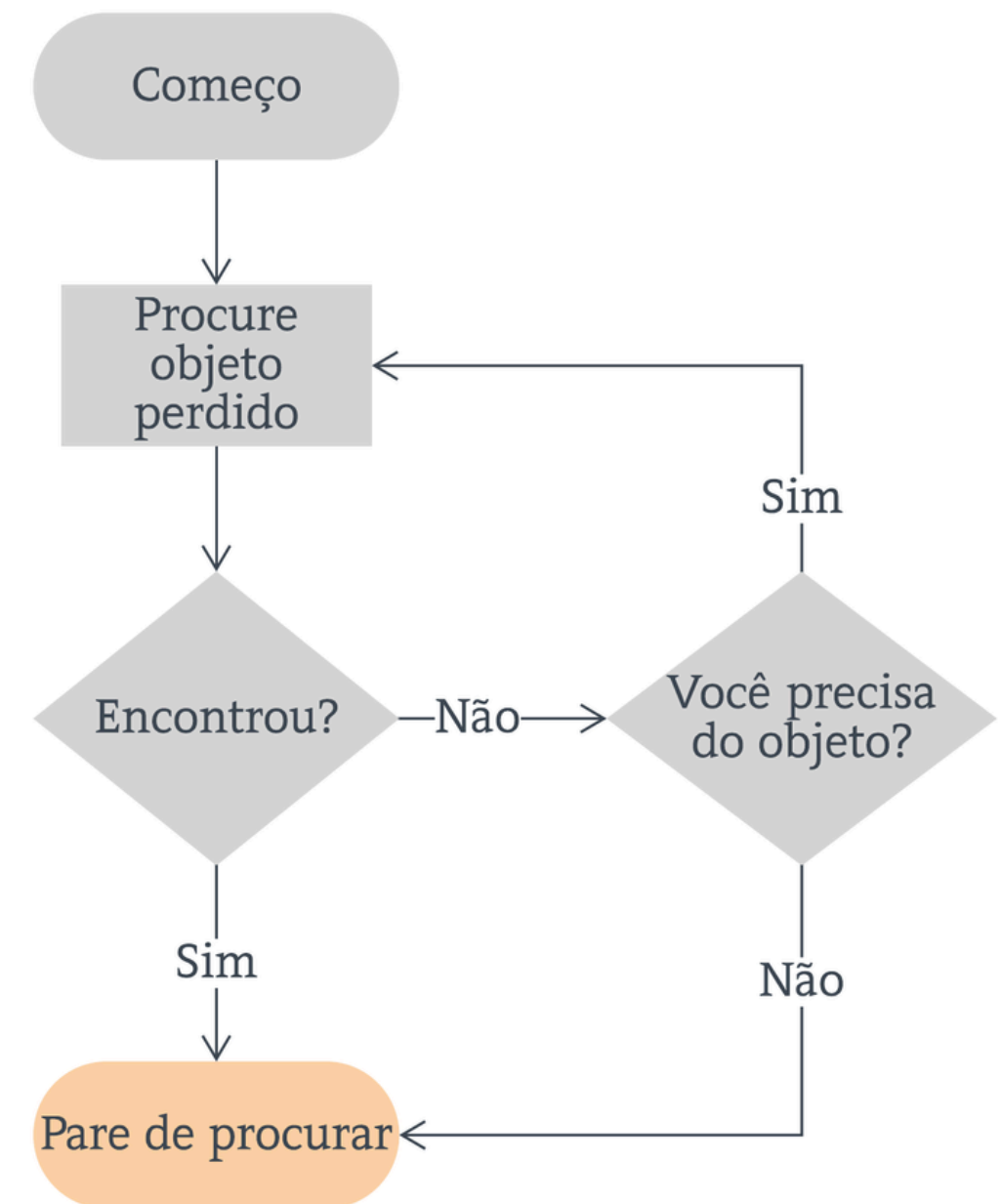
2. Consistência de Dados:

- a. Padrão Saga para transações (contratos) para não quebrar.
- b. Não quebrar a comunicação



Quando Adotar?

- **Evitar Se:**
 - Startup começando
 - Não tiver muitos recursos
- **Indicado Para:**
 - Times grandes
 - Necessidade de inovação rápida



Exemplo Prático:

1 - Quando o usuário acessa o /login

2 - APIgateway requisita dados do microserviço User.

APIGATEWAY

```
@login_bp.route('/login', methods=['POST', 'GET'])
def login():
    if request.method == 'POST':
        username = request.form.get("username")
        password = request.form.get("password")

    try:
        response = requests.post(f"{USER_URL}/login", json={"username": username, "password": password})
        if response.status_code == 200:
```

Microserviço User

```
@auth_bp.route('/login', methods=['POST'])
def login():
    data = request.get_json()
    username = data.get('username')
    password = data.get('password')

    user_student = Student.query.filter_by(user
    user_teacher = Teacher.query.filter_by(user
```

3 - O microserviço User retorna a resposta para o gateway para ser visualizado pelo usuário

