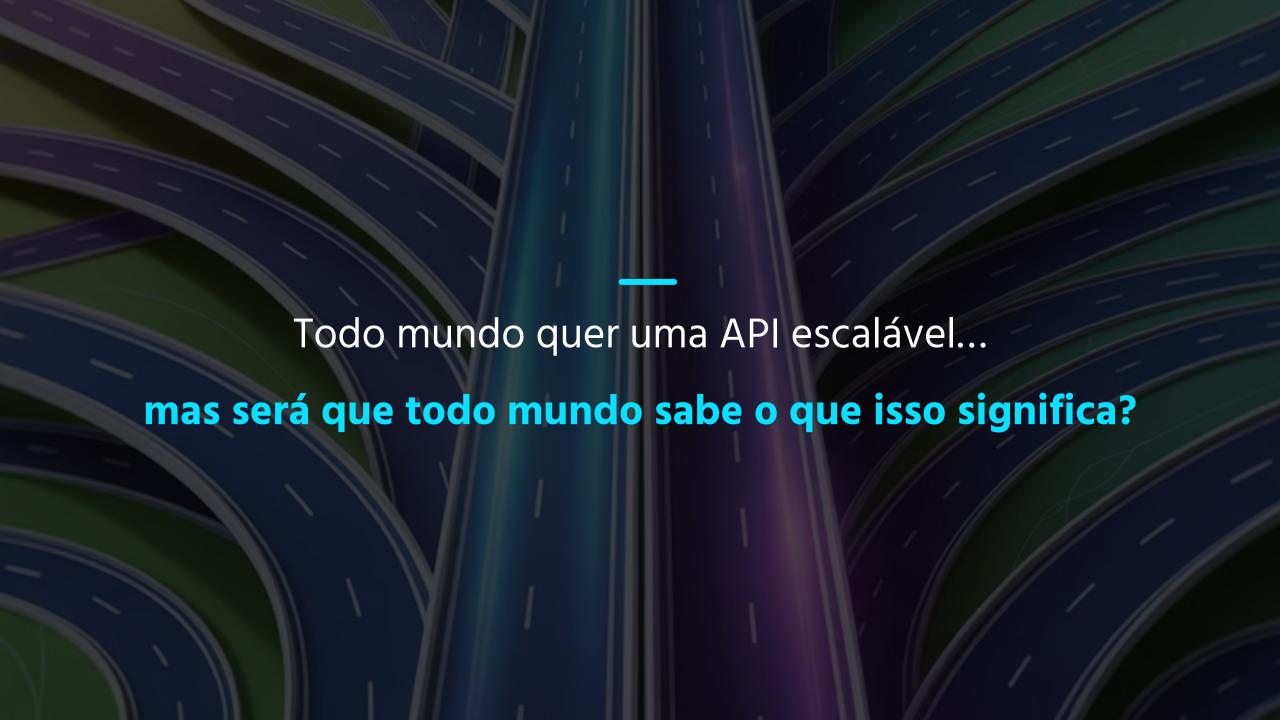
DESENVOLVIMENTO DE APIS ESCALÁVEIS: BOAS PRÁTICAS

E PADRÕES

Arquitetura, Código e Sustentabilidade



Arquitetura: o alicerce da escalabilidade

- 1. Importância das decisões arquiteturais;
- 2. Impacto na performance e manutenção.



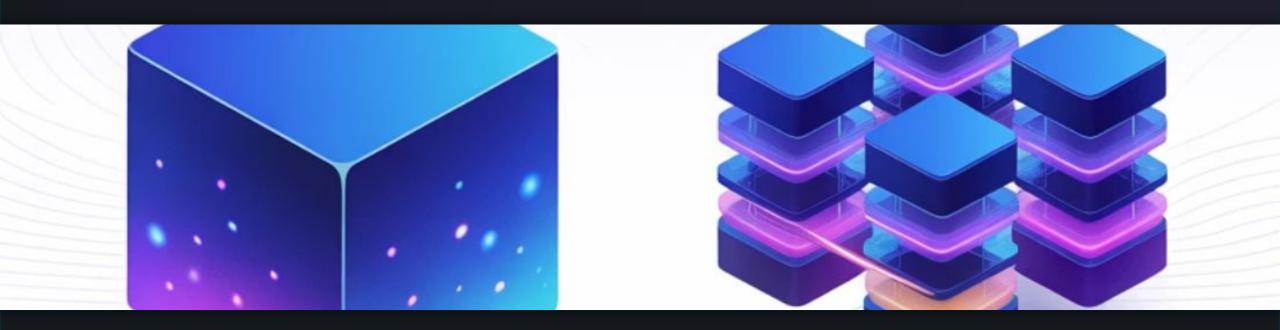
Escolhas arquiteturais

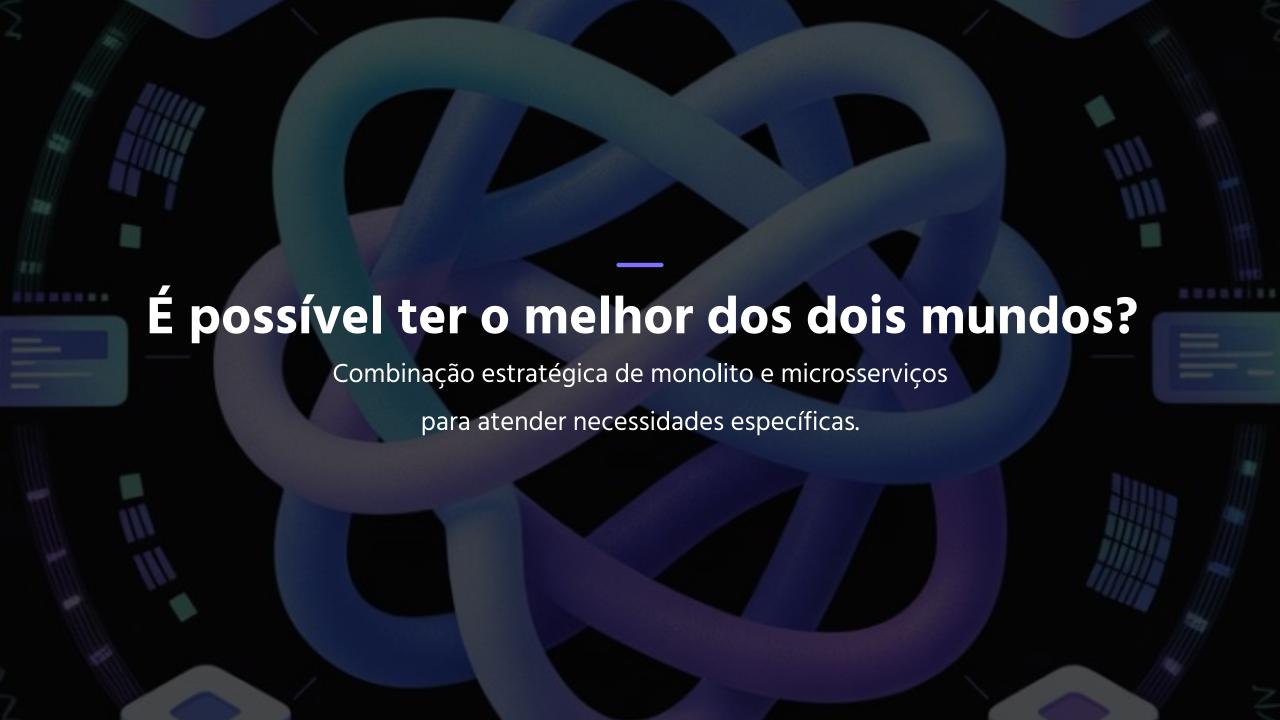
Monolito

Simplicidade, deploy único, difícil de escalar partes específicas.

Microsserviços

Escalabilidade, deploys independentes, complexidade aumentada.







Boas práticas de desenvolvimento

- RESTful ou GraphQL bem projetado;
 - Versionamento de API;
- Stateless (essencial para escalar horizontalmente);
 - Desacoplamento por filas;
 - Clusterização e balanceamento.



Segregação de bases de dados

- Base Transacional (OLTP) vs Base Analítica (OLAP);
- Benefícios da separação: **performance, segurança, especialização.**



Banco de dados

- Base Transacional (OLTP) vs Base Analítica (OLAP);
 - Índices bem definidos para consultas frequentes;
 - Big Data e Data Lake.

Pensando no futuro

- Importância da documentação;
- Testes automatizados;
- Observabilidade;
- Segurança.



Evolução sem overengineering

Adapte a arquitetura conforme o crescimento e as necessidades reais do sistema.



Resumo e reflexão

- Arquitetura bem pensada;
- Padrões de código consistentes;
- Suporte e manutenção facilitados.

Qual a decisão arquitetural que você está postergando hoje e que pode te custar caro amanhã?