

The background features a dark, stylized illustration of server racks in the foreground and several large, dark blue and purple clouds in the background. The overall aesthetic is tech-oriented and modern.

DESENVOLVIMENTO DE APIs ESCALÁVEIS: BOAS PRÁTICAS E PADRÕES

Arquitetura, Código e Sustentabilidade



—

Todo mundo quer uma API escalável...

mas será que todo mundo sabe o que isso significa?

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.



Microserviços

Escalabilidade, deploys independentes, complexidade aumentada.





É possível ter o melhor dos dois mundos?

Combinação estratégica de monolito e microsserviços
para atender necessidades específicas.



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?