

# Desafios da Arquitetura de Microsserviços

Design e Comunicação

Victor Osório





**Victor Osório**

**Twitter: @vepo**

**LinkedIn: [linkedin.com/in/victorosorio/](https://www.linkedin.com/in/victorosorio/)**

- **Engenheiro de Computação pela UNICAMP**
- **MBA em Arquitetura de Software na FIAP**
- **Arquiteto e desenvolvedor Java/Apache Kafka**
- **+ 18 anos com Java**
- **+ 5 anos com Apache Kafka**
- **+ 3 anos na Amdocs**
- **Autor: Roadmap back-end (Casa do Código)**
  - **<https://www.casadocodigo.com.br/products/livro-roadmap-backend>**





# Agenda

- 1 O que são *Microservices*?
- 2 Existem padrões de *Microserviços*?
- 3 Como eles se comunicam?
- 4 Como começar?
- 5 Referências e próximos passos



# O que são Microseriços

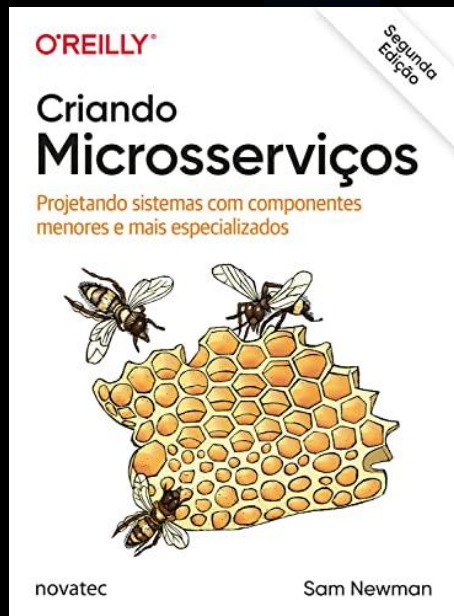




# O que são Microservices?

Microserviços (microservices) são serviços que podem ser lançados de forma independente e são modelados com base em um domínio de negócios.

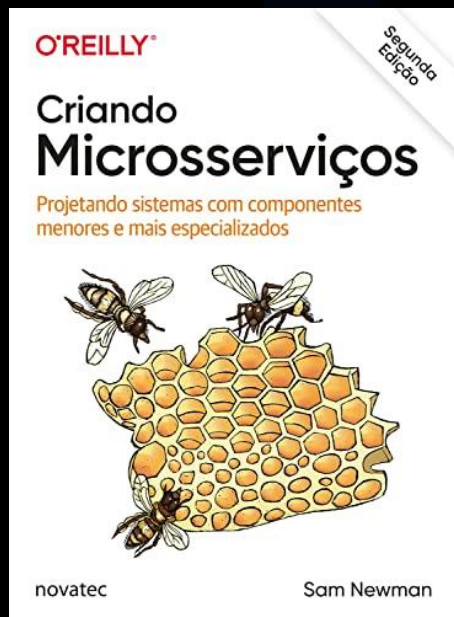
Sam Newman em Criando Microseriços



# O que são Microservices?

Microserviços (microservices) são serviços que **podem ser lançados de forma independente** e são modelados com base em um domínio de negócios.

Sam Newman em Criando Microserviços



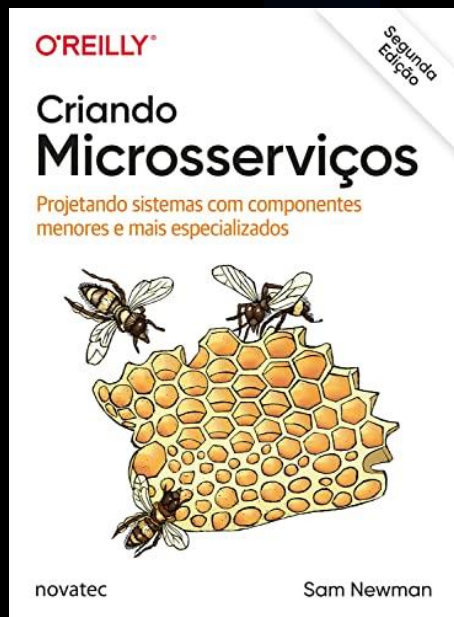
## Independência



# O que são Microservices?

Microserviços (microservices) são serviços que podem ser lançados de forma independente e **são modelados com base em um domínio de negócios**.

Sam Newman em Criando Microseriços



Independência  
Domínio

# O que significa Independência?

Times pequenos entregando software em produção constantemente!

Impulsionado pelas metodologias ágeis

- Rough Design Up Front
- Two Pizza Team





# O que significa Independência?

Times pequenos entregando software em produção constantemente!

Impulsionado pelas metodologias ágeis

- Rough Design Up Front
- Two Pizza Team

Flickr criou o DevOps

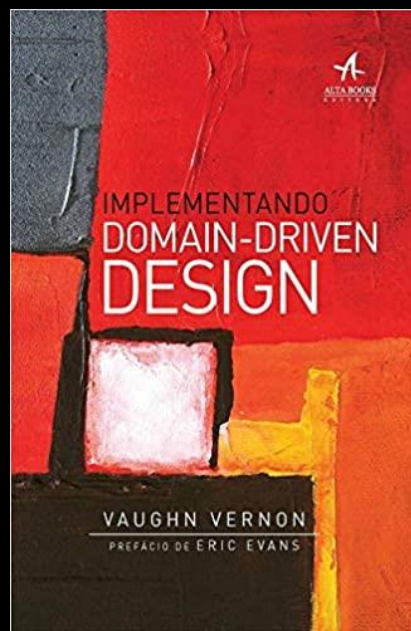
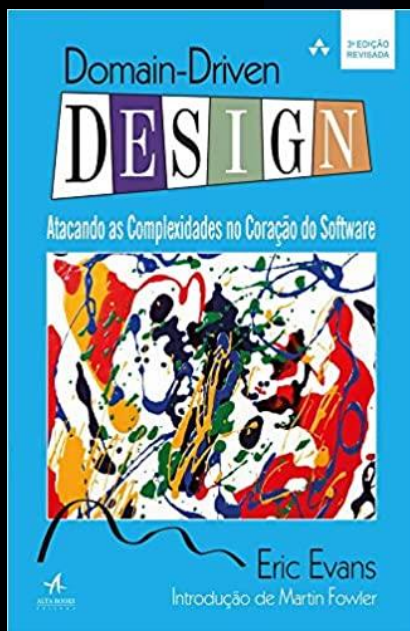
10 deploys per day  
Dev & ops cooperation at Flickr

John Allspaw & Paul Hammond  
Velocity 2009

# O que significa Domínio?

Domain-Driven Design e Microserviços tem uma estreita relação.

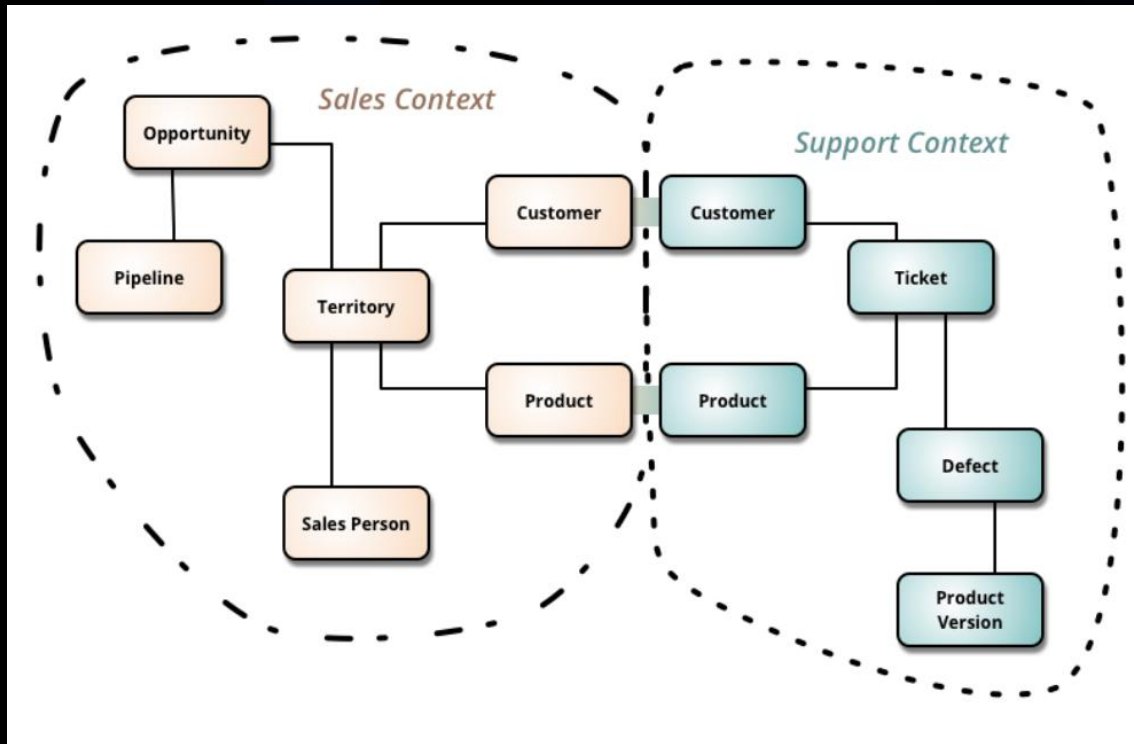
Contexto Limitado foi um padrão do DDD que facilitou a comunicação entre times distribuídos.





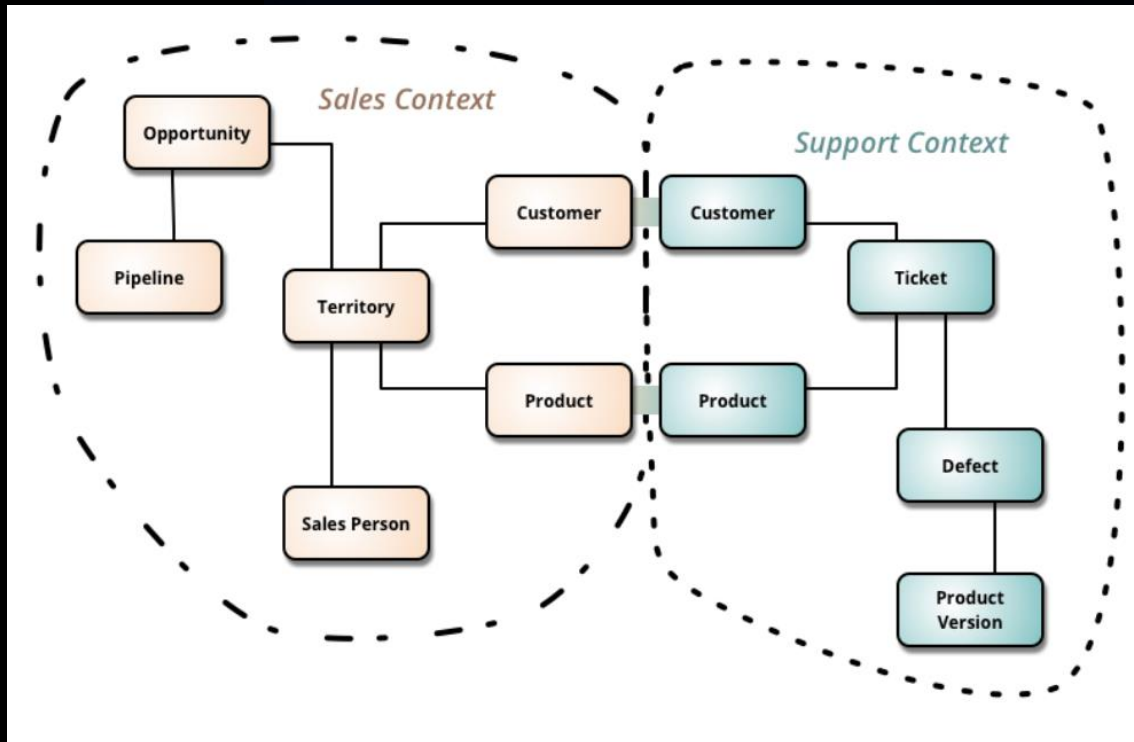
# O que é DDD?

**Domain-Driven Design** é uma técnica de modelar software baseado no modelo de negócios e não pelas características técnicas do software.



# O que é DDD?

**Domain-Driven Design** incentiva a limitação do contexto para facilitar a compreensão e manutenabilidade do software.



Limitando o contexto é possível:

- Diminuir o acoplamento,
- Criar interface entre módulos
- Criar uma camada Anti-Corrupção para evitar problemas de integração.





# Existem padrões de Microserviços?



# Existem padrões de Microsserviços?

**TL;DR;**

SIM!

- Descoberta de Serviços
- Decomposição
- Testes
- Observabilidade
- Estilo de Comunicação
- Deployment



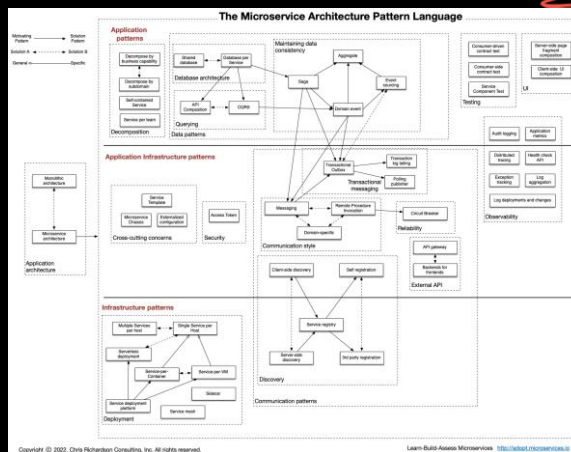
# Existem padrões de Microserviços?

TL;DR;

SIM!

- Descoberta de Serviços
- Decomposição
- Testes
- Observabilidade
- Estilo de Comunicação
- Deployment

Ver site [microservices.io](https://microservices.io)





# Existem padrões de Microserviços?

**Não existe transação! Existe Sagas**

**Base de dados compartilhada?**

**Registro de Serviços?**

**Descoberta Client-Side?**

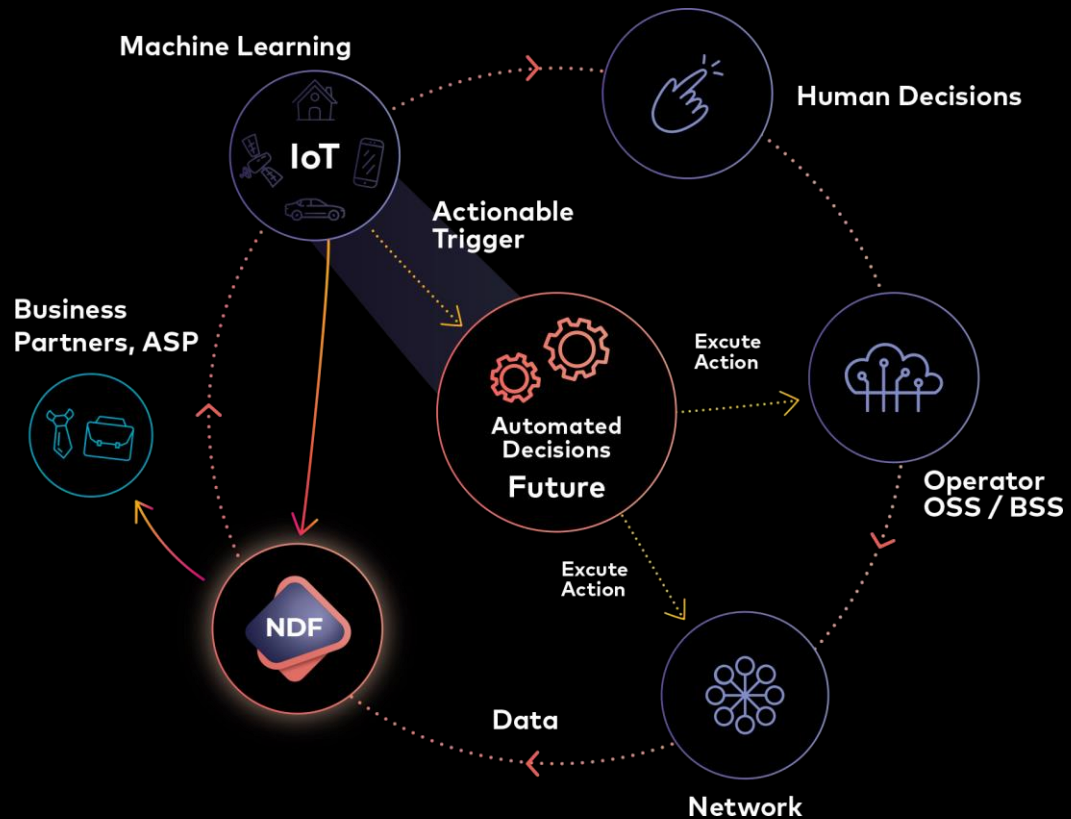
**API Gateway?**

**Existem padrões complexos que podem facilitar a escolha arquitetural**



# O que usamos na Amdocs?

## Event Sourcing com Apache Kafka







# Como eles se comunicam?





# Protocolos?!?

Não existem padrões, MAS..

Alguns métodos tem um lugar de destaque no mercado

- HTTP/REST
- gRPC
- Pub/Sub



# REST/HTTP

O que vale a pena saber?

- API Composition
- Menos acoplamento
- API First com OpenAPI





# gRPC

O que vale a pena saber?

- Baixa latência
- Interfaces bem definidas (Protobuf)
- Não recomendável para clientes externos



# Pub/Sub

O que vale a pena saber?

1. Modelo Reativo ([www.reactivemanifesto.org](http://www.reactivemanifesto.org))
2. Orientado a Mensagens
3. Escolha do middleware (RabbitMQ? Apache Kafka?)



# Como começar?



# Escolha um Framework!









# Frameworks Cloud Native

O que eles oferecem?

# Como avaliar um Framework?

- Performance 
- Observabilidade  
Resiliência  
Monitoramento 
- Boa documentação 
- Aceito pela comunidade 





# Como usar um Framework?

- Cloud Native?!?
  - Não dependa de máquina
  - Automatize o deploy
  - Externalize configurações
  - Crie containers
  - Entenda os frameworks



# Referências e próximos passos





# Comunidade

Fique atento!

Como aprender mais?

- Fique atento a Meetups
- Procure palestras no Youtube
- Livros
- [github.com/vepo/docker-tutorial](https://github.com/vepo/docker-tutorial)
- [github.com/vepo/k8s-tutorial](https://github.com/vepo/k8s-tutorial)

Sempre existe alguém falando sobre o assunto nas redes sociais!

Thank you

Let's  
**make it**  
**amazing**

