COMO EVITAR MONOLITOS DISTRIBUÍDOS?

No hype dos micro serviços, quais os pontos de atenção que devemos ter para não construirmos monolitos distribuídos?

ThoughtWorks®

ABOUT ME

Juliana Fernandes

Software Developer

ThoughtWorks®

PRIMEIRAMENTE...

“Como empresa, eu quero me adaptar para mudar rapidamente...”

Ciclo rápido de feedback

- Entreguar software
- Aprender
- Adaptar

Entrega contínua

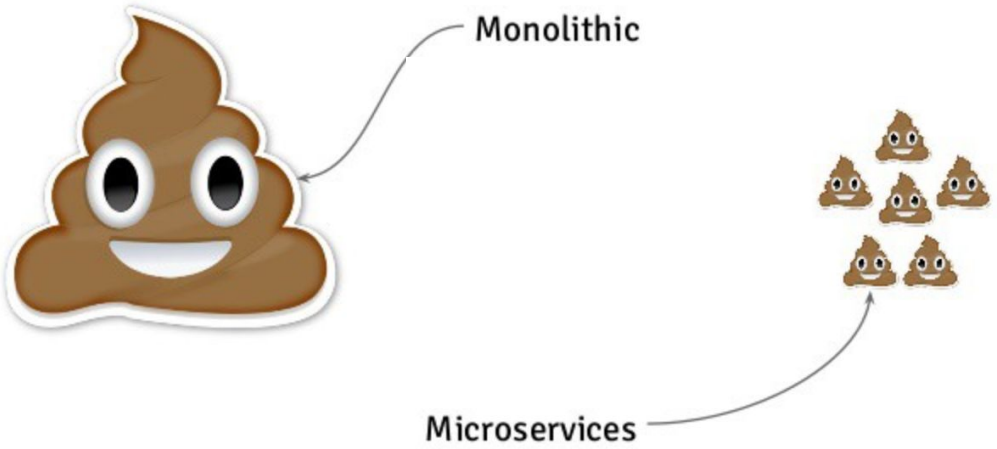
- Deploys rápido e frequentes
- Testes automatizados
- Infraestrutura automatizada

Mas como fazer isso nesse nosso monolito?

Quebrar o monolito pareceu uma boa ideia agora, não?

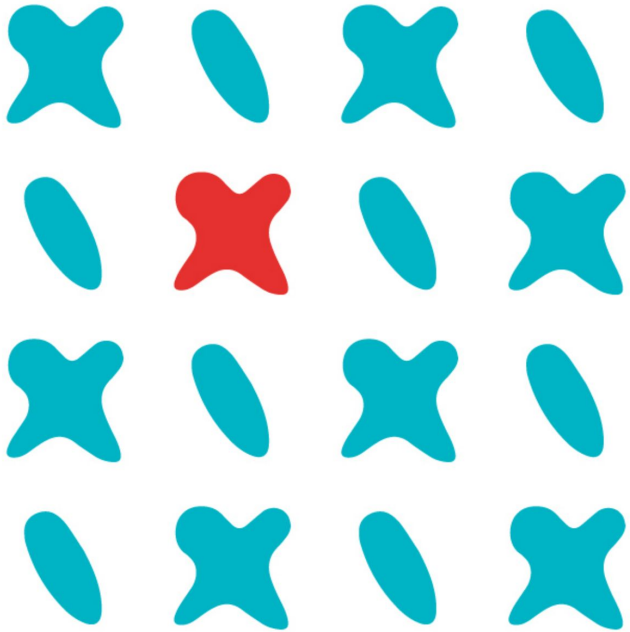
Let's build a Services Oriented Architecture(SOA)...

*Eis o **Monolito Distribuído™!***
(É um monólito, mas distribuído)



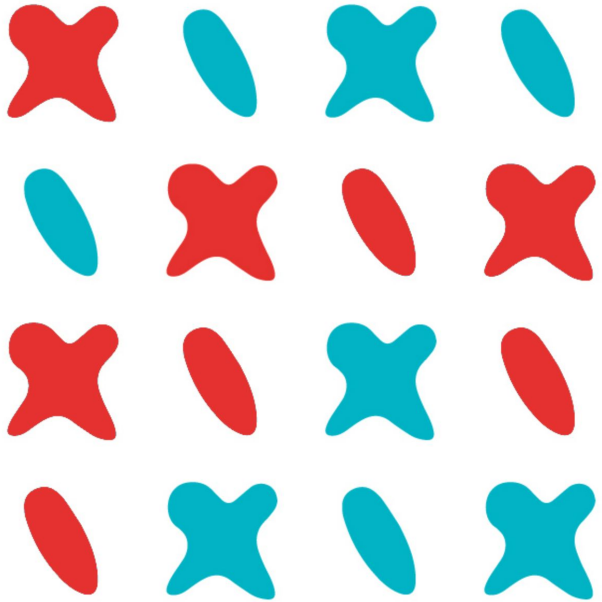
O MONOLITO DISTRIBUÍDO

- Mudança em uma parte



O MONOLITO DISTRIBUÍDO

- Quebra o sistema inteiro



> *Making changes in lots of different places is slower, and deploying lots of services at once is risky — both of which we want to avoid.*
>
> **— Sam Newman**

Regra das duas pizzas

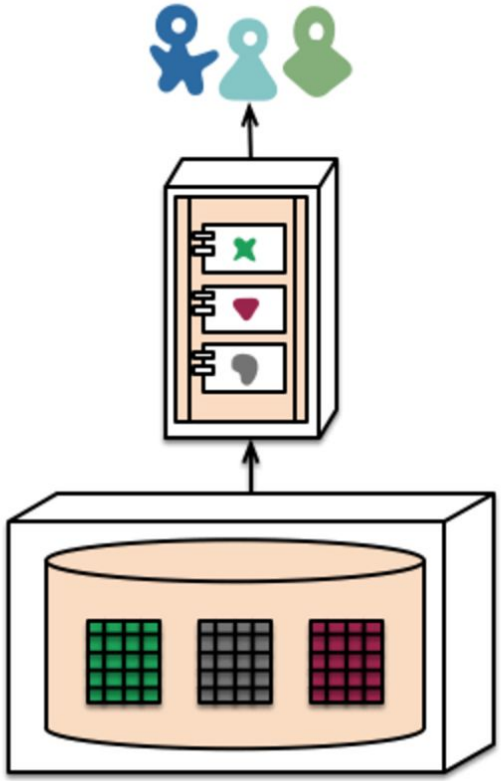
Jeff Besos - Amazon

Lei de Conway

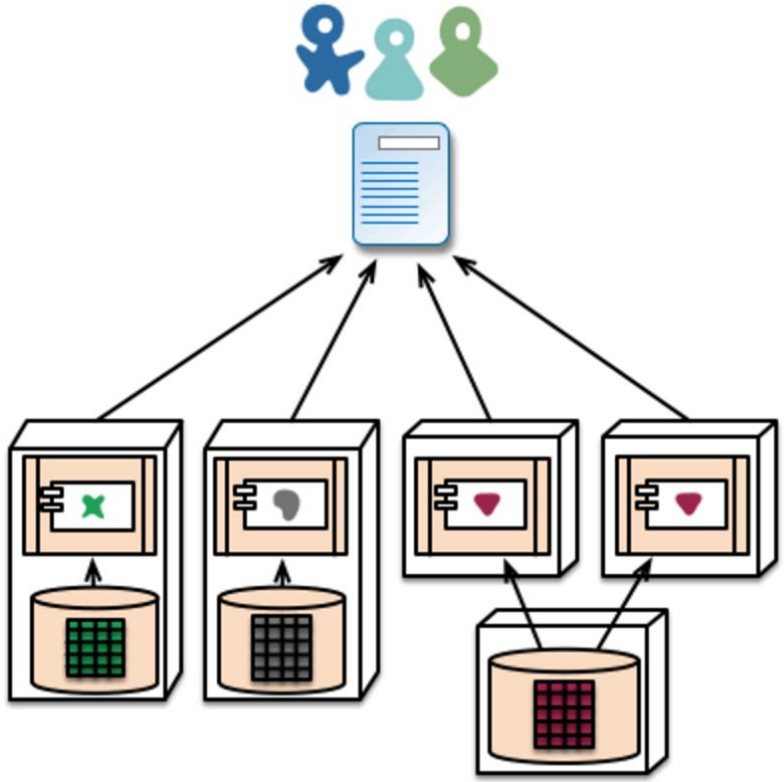
> *Organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations*
>
> — **M. Conway**

Não construa micro serviços, ao menos que...

- Codebase muito grande
- Time grande
- Difícil deployar mudanças isoladas
- Muita sobrecarga de comunicação
- Mudança na estrutura da organização
- Testes demoram muito tempo para executar



Monolito - Banco de dados único



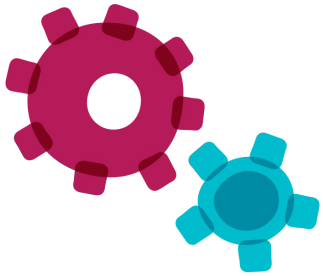
Micro serviços - multiplos banco de dados

BENEFÍCIOS

Autonomia com responsabilidade

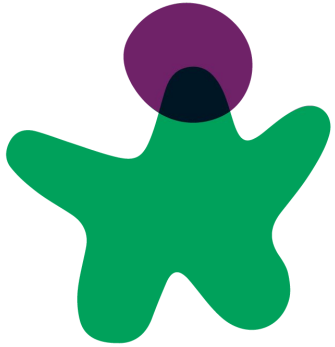
Agilidade para mudar

Composição de inovação



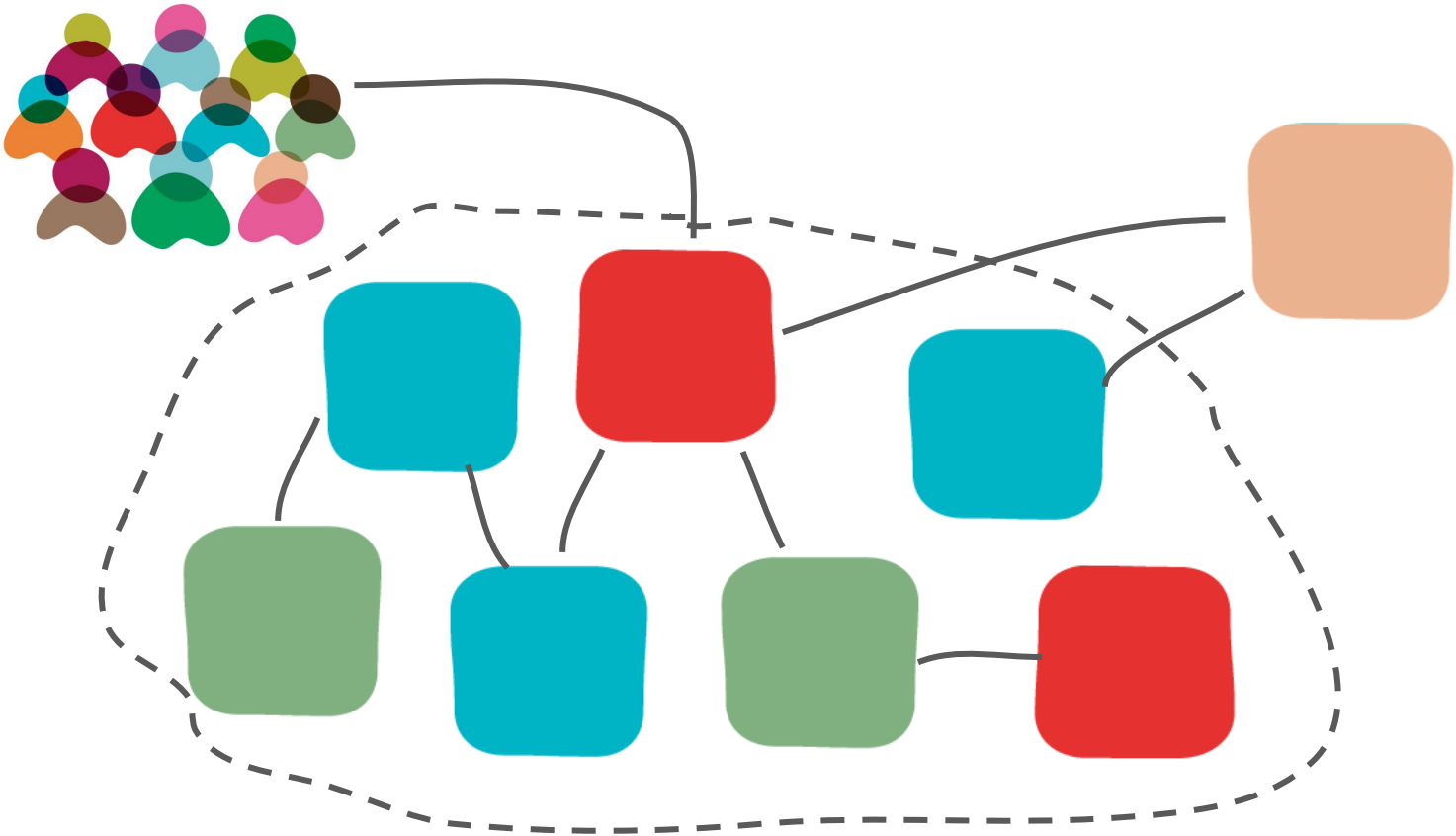
Diversidade técnica

Escalabilidade

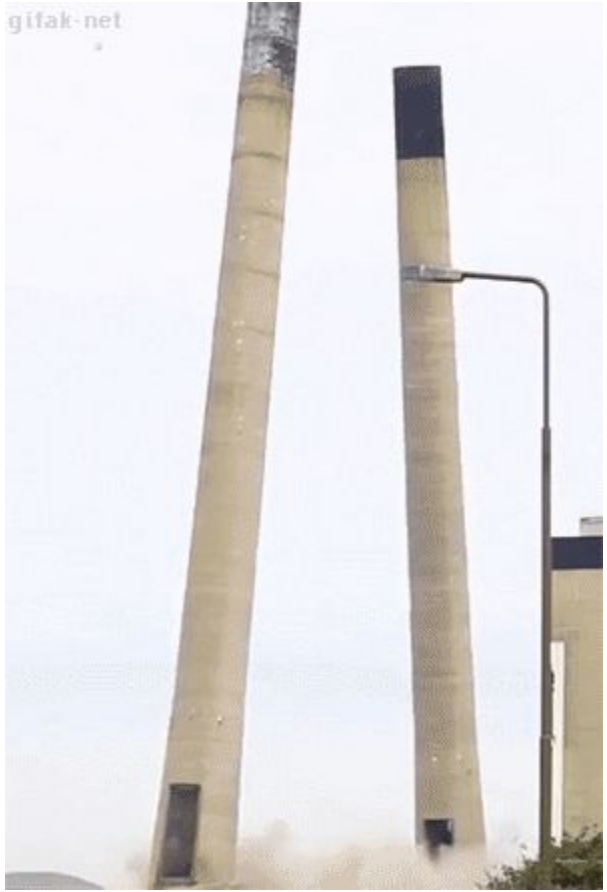


Carga cognitiva

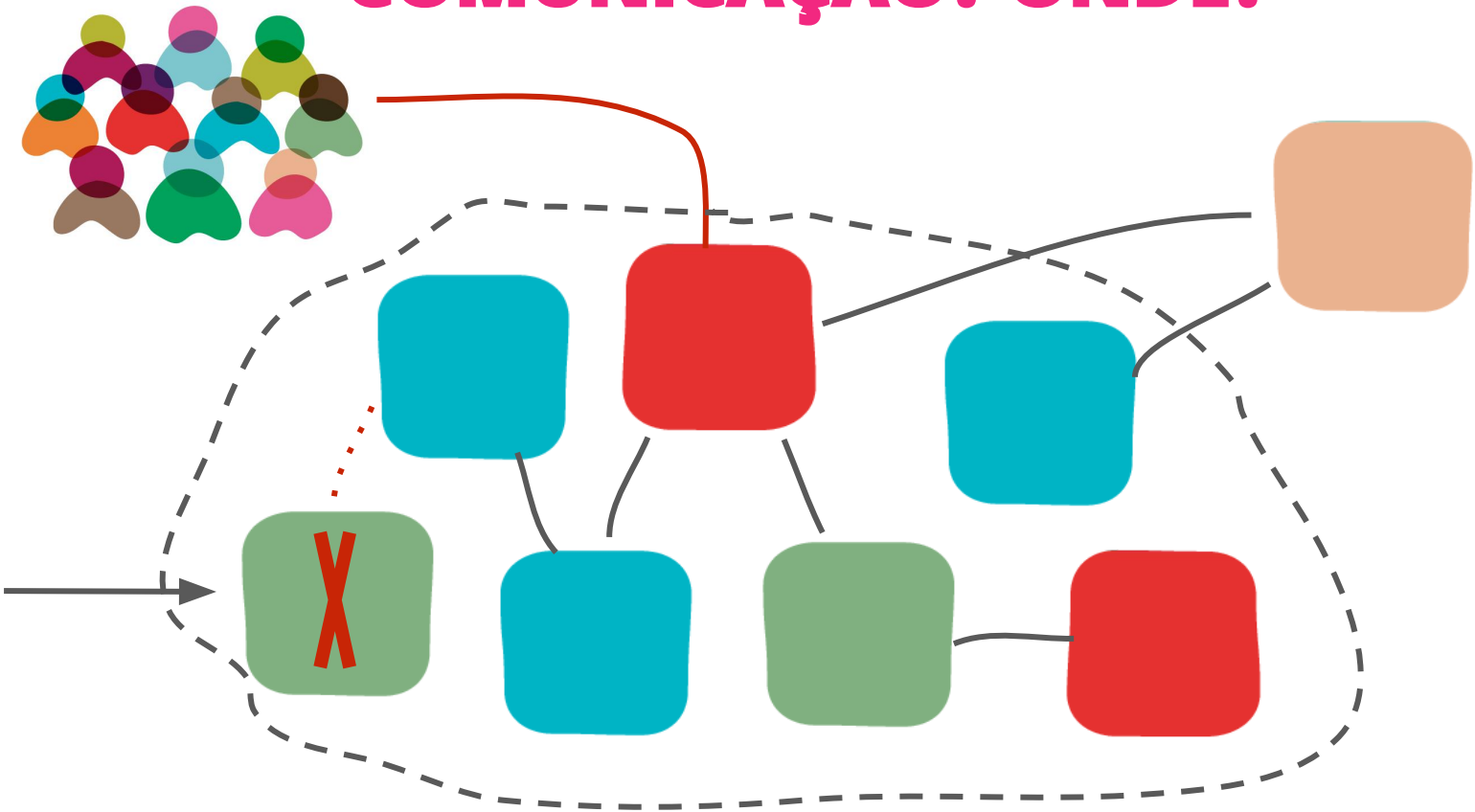
MAAASS...



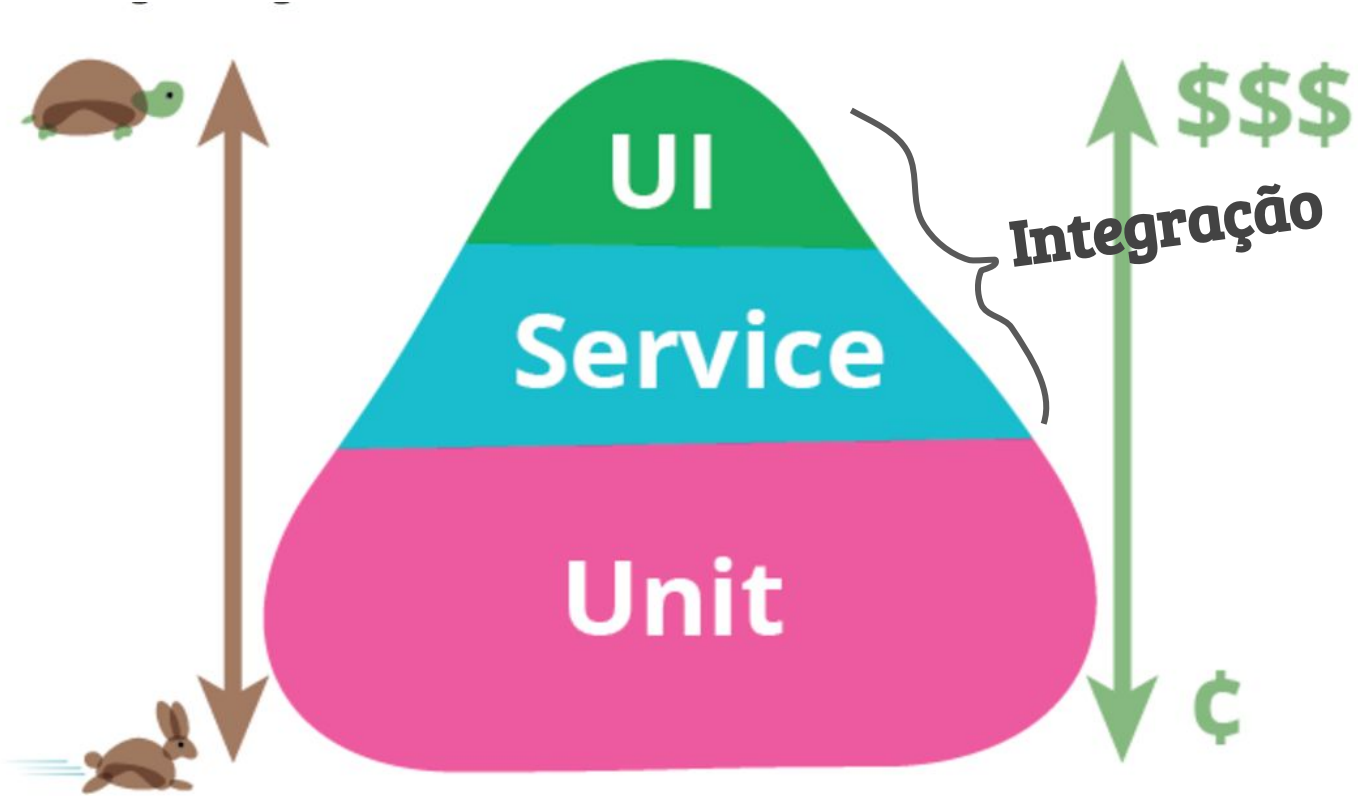
DIA DE RELEASE



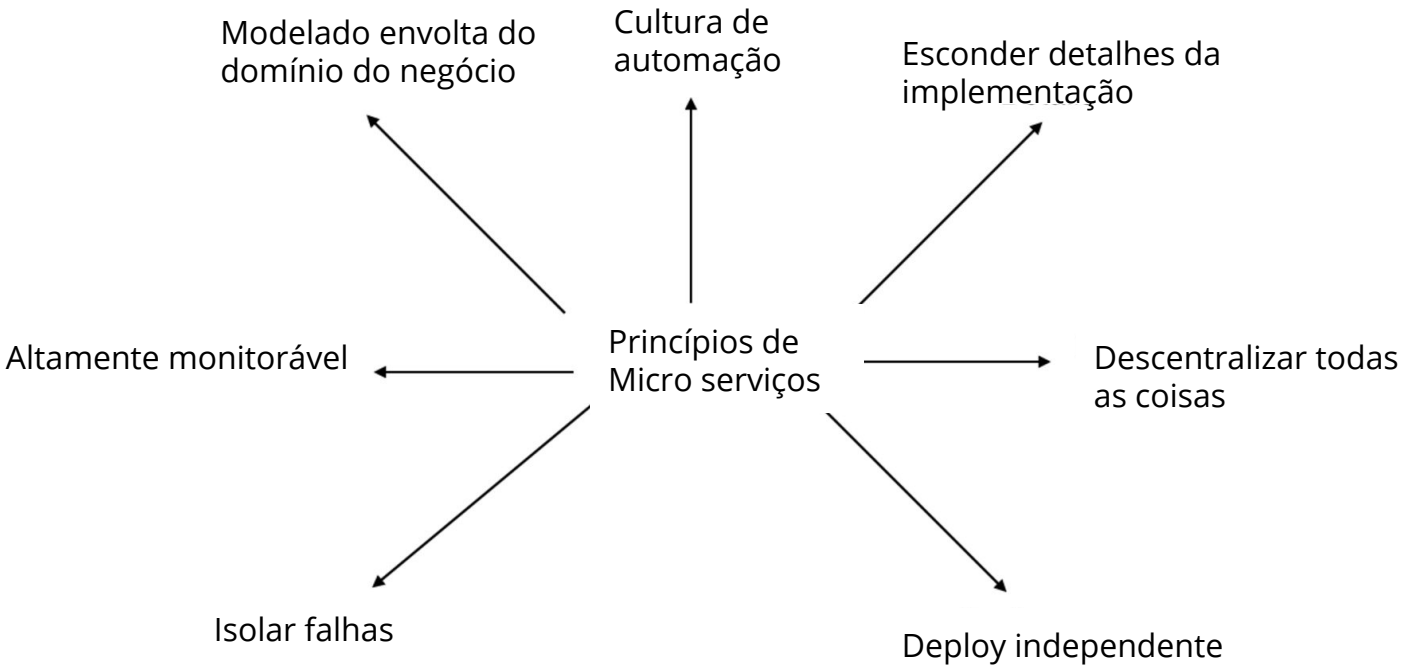
QUEBRA DE CONTRATO? FALHA NA COMUNICAÇÃO? ONDE?



E A PIRÂMIDE?



MICRO SERVIÇOS: PRINCÍPIOS



ENTÃO..MANTENHA EM MENTE:

- Como os serviços irão se comunicar?
- Como saber se quebramos alguma funcionalidade?
- Onde está o código que implementa a lógica de pagamento?
- E o banco de dados? Duplicação/sincronização de dados?
- Como será o deploy dos serviços?
- Logs, Monitoramento?! Como isolar um problema?

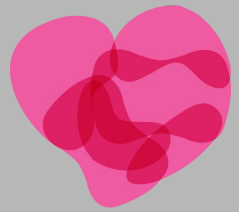
O QUE VOCÊ APRENDEU HOJE

- Nem sempre quebrar o monolito é a solução dos seus problemas.
- Micro serviços não resolvem todos os seus problemas, aliás, criam outros.
- Quebrar seu monolito te ajuda a ter agilidade e entrega contínua.
- Micro serviços, pra que? pra quem?
- A arquitetura da sua aplicação é diretamente relacionada com a arquitetura da sua organização.
- Teste, monitore e isole o problema.

Bônus / Referências

- Microservices - Martin Fowler
- Integration Contract Test - Martin Fowler
- Testing Strategies in a Microservice Architecture - Toby Clemson
- Backends For Frontends Architecture - Sam Newman
- Our journey to microservices: One repo vs Multiple repos - Avi Cavale
- Microservice trade offs - Martin Fowler
- O que aprendi com as mudanças de uma arquitetura monolitica para micro serviços - André Ronquetti

DÚVIDAS?



avalia.se/phprs

Juliana Fernandes

littlejuh@live.it
Twitter: @littlejuh_
Github: littlejuh