

# Aplicações escalonáveis e Prática CI/CD



DEVinHouse

Parcerias para desenvolver a sua carreira

**SENAI**

<LAB365>

# AGENDA

- Escalabilidade
- Microsserviços
- Revisão DevOps e CI/CD
- LAB

- A **escalabilidade** se refere a habilidade de um software ser **escalável**, ou seja, que possa atender um grande número de clientes conforme a necessidade e também deve ser capaz de diminuir a sua capacidade quando a demanda diminui.
- Através da **escalabilidade** de software temos menos **sobrecargas** de programas e mais tolerâncias a falhas, além disso temos uma **redução de custos**, pois a escalabilidade permite que o programa utilize apenas os recursos **necessários**.

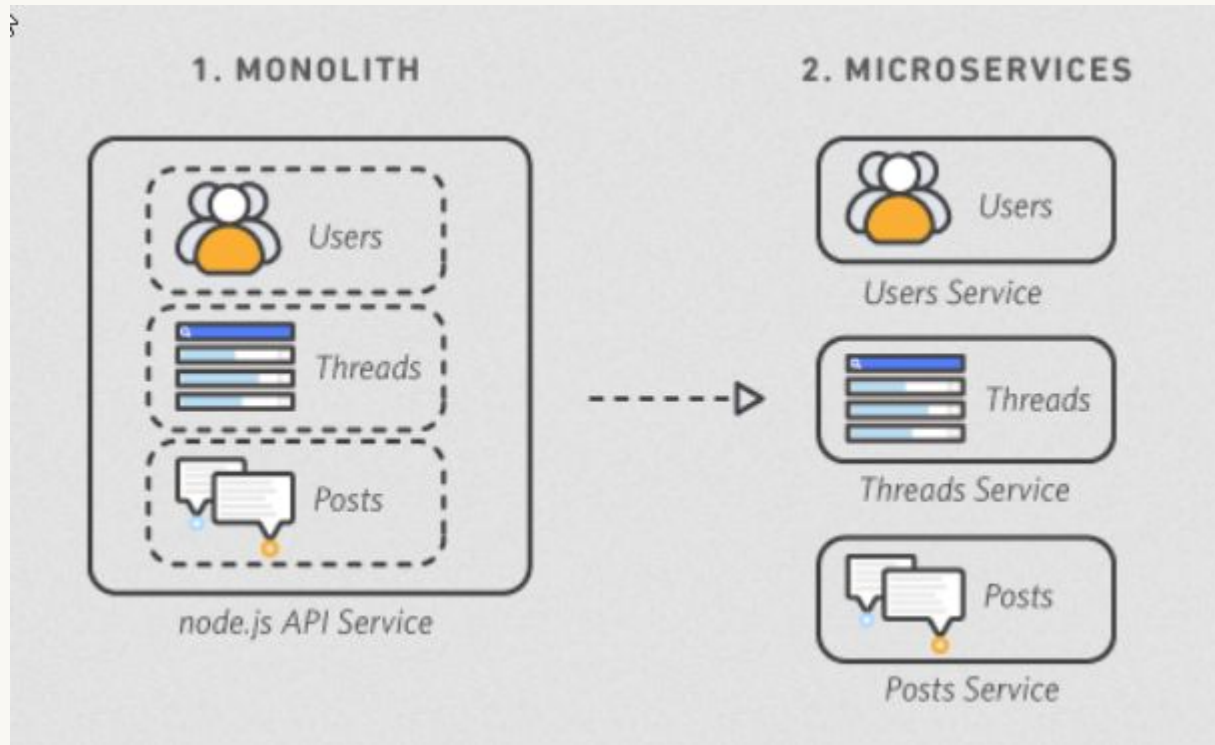
- Para criarmos um programa escalonável precisamos juntar diversos conceitos e práticas de tecnologia, entre elas:
  - **Cloud** - fornece um ambiente capaz de entregar capacidade computacional sob demanda
  - **Microserviços** - é uma arquitetura que nos permite criar aplicações mais ágeis e escalonáveis
  - **DevOps** - é o processo para criar software em escala
  - **CI/CD** - é o processo que permite a entrega de software em escala
  - **Containers** - é o software entregue e pronto para ser escalonado, pode ser uma Kubernetes ou em outro serviço de containers.

- A discussão sobre microserviços data de **2004**, na época sendo chamados de **Micro-Web-Services**, e hoje em dia eles são adicionados ao **DevOps** na construção de soluções de software
- A **arquitetura de microserviços** é uma forma de estruturarmos um sistema para que cada serviço interno dele seja setorizado e criado de forma independente, assim ao invés de termos um sistema com diversos módulos ou responsabilidades temos vários sistemas com escopes reduzidos e que devem atuar de forma autônoma.

- Um **microserviço** pode ser estruturado da seguinte forma:
  - ◆ A aplicação central que tem apenas uma função;
  - ◆ Um armazenamento, nesse caso é opcional;
  - ◆ Um meio de comunicação com aplicações externas, pode ser através de HTTP REST, Filas ou Tópicos.

- Temos então que os microserviços são:
  - ◆ **Autônomos** - pode ser desenvolvido, alterado, testado sem interferir em outro microserviço
  - ◆ **Especializados** - os recursos utilizados por um microserviço e o seu próprio código existem para resolver um problema específico.

# Microsserviços





# Adoção de Microsserviços

- A **arquitetura orientada a microsserviços** vem sendo grandemente adotada porque as aplicações crescem demais, muito rápido, e com muitas pessoas envolvidas.
- Sendo assim, há uma necessidade de uma empresa **quebrar** os seus serviços em microsserviços capazes de realizar as **diversas tarefas** das aplicações que elas têm sem gerar **gargalos** de quantidade de pessoas por projeto.

# Adoção de Microsserviços

- Atualmente também temos o DevOps e o Ágil que nos levam a criarmos equipes pequenas, o que colabora para a adoção de microsserviços. Que são unidades independentes de um serviço maior
- No DevOps ainda temos a presença do conceito de provisionamento de infraestrutura por código que nos permite preparar ambientes de forma muito mais confiável, segura e replicável, pois elimina parte do fator humano.

# Vantagens dos Microsserviços

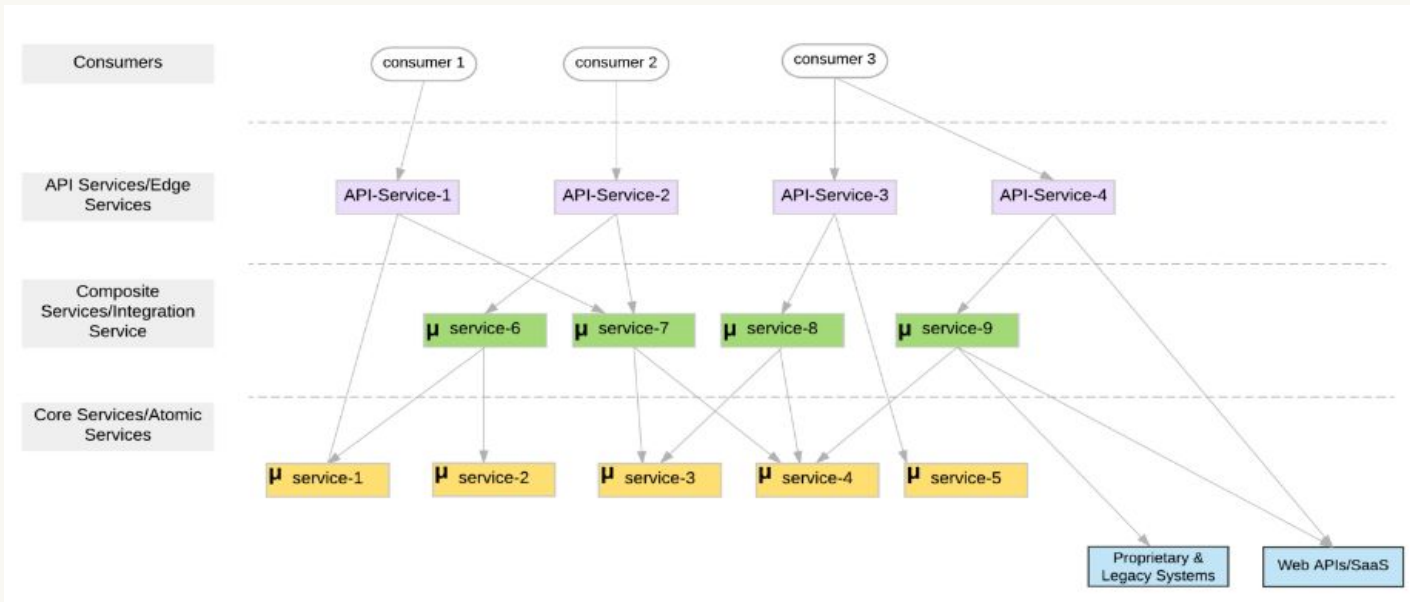
- **Agilidade** - ciclos de desenvolvimento mais curto;
- **Escalabilidade flexível** - o serviço tem a capacidade correspondente a demanda;
- **Fácil implantação** - são pequenos e por isso fáceis de terem entregas contínuas e em ciclos curtos;
- **Liberdade tecnológica** - uma equipe ou empresa está livre para escolher a tecnologia de um determinado serviço;
- **Resiliência** - eles têm alta tolerância a falhas, pois se um microsserviço cai, os outros ainda ficam no ar;

# Desvantagens dos Microserviços

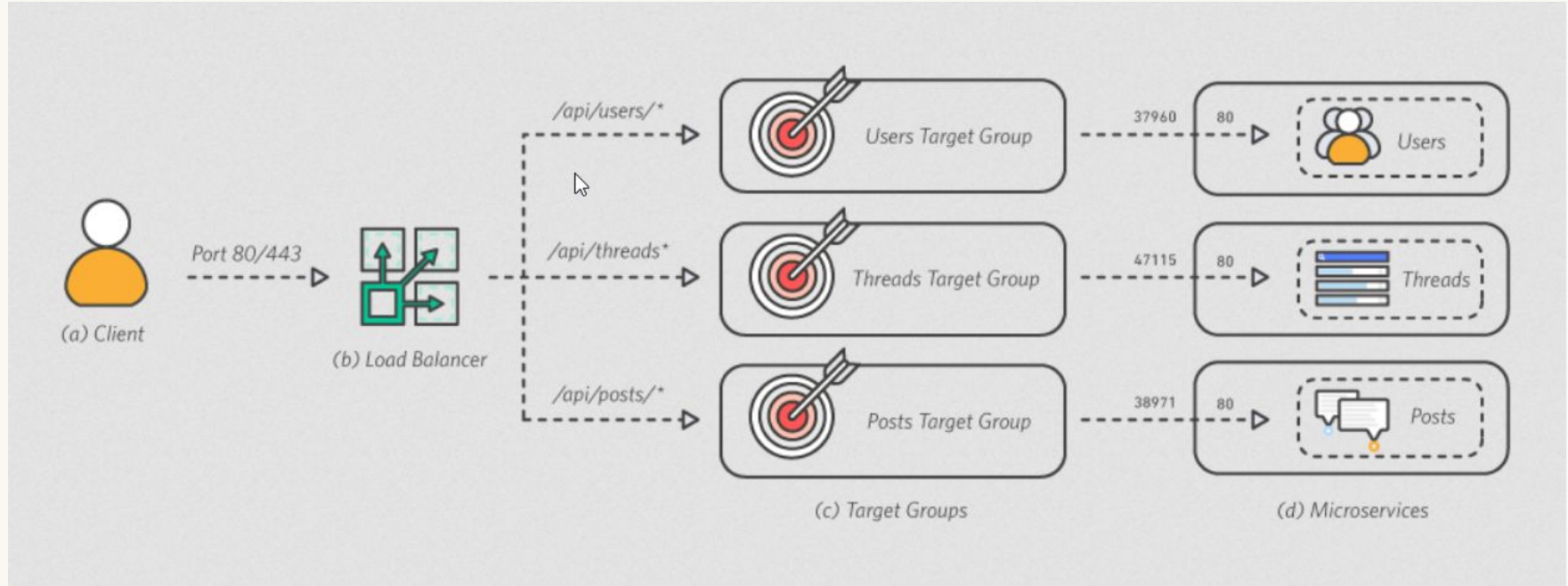
- Maior complexidade de desenvolvimento e de criação de infraestrutura;
- Debug mais complexo, esse caso pode acontecer com programas muito dependentes uns dos outros;
- Diversas tecnologias podem causar um problema de busca de profissionais;
- O monitoramento é muito mais crucial e complexo.

# Arquitetura de Microsserviços em Camadas

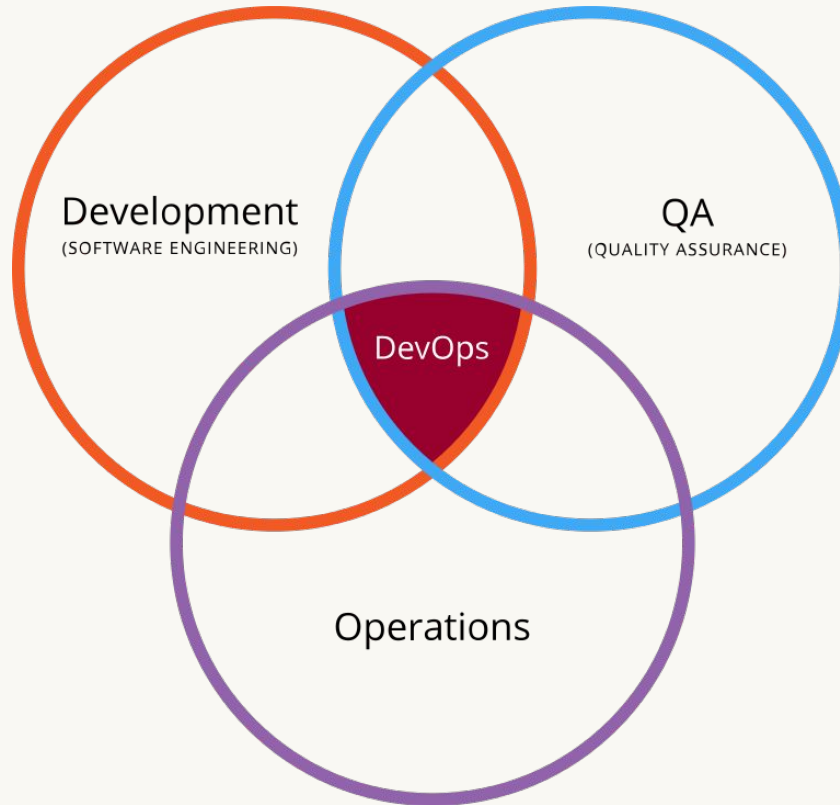
- Os microsserviços podem ser dispostos em camadas, criando uma cadeia concisa de comunicação entre vários microsserviços



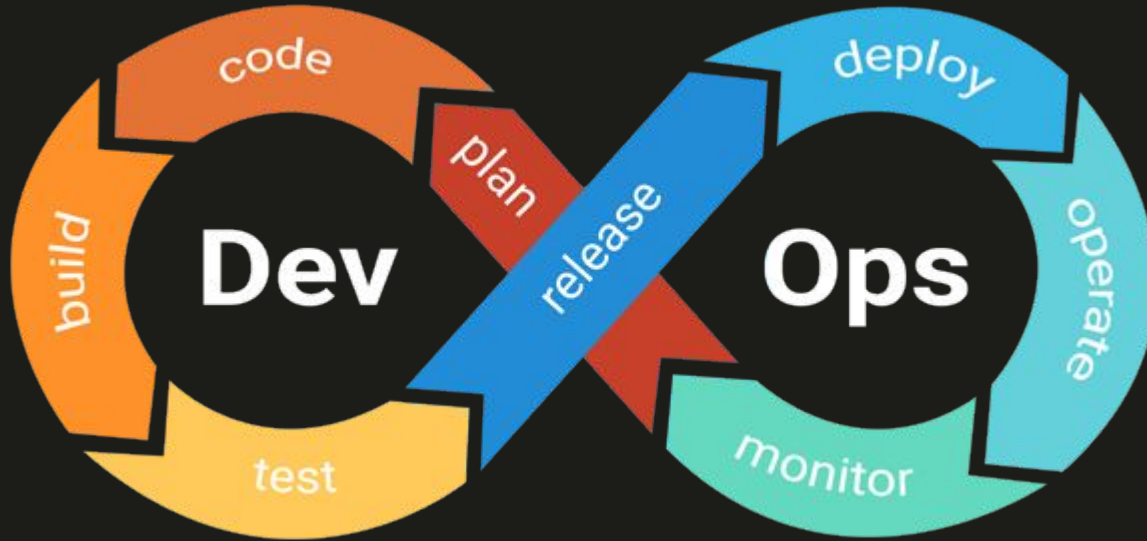
# Exemplo de Arquitetura com Microserviços



# Revisão DevOps

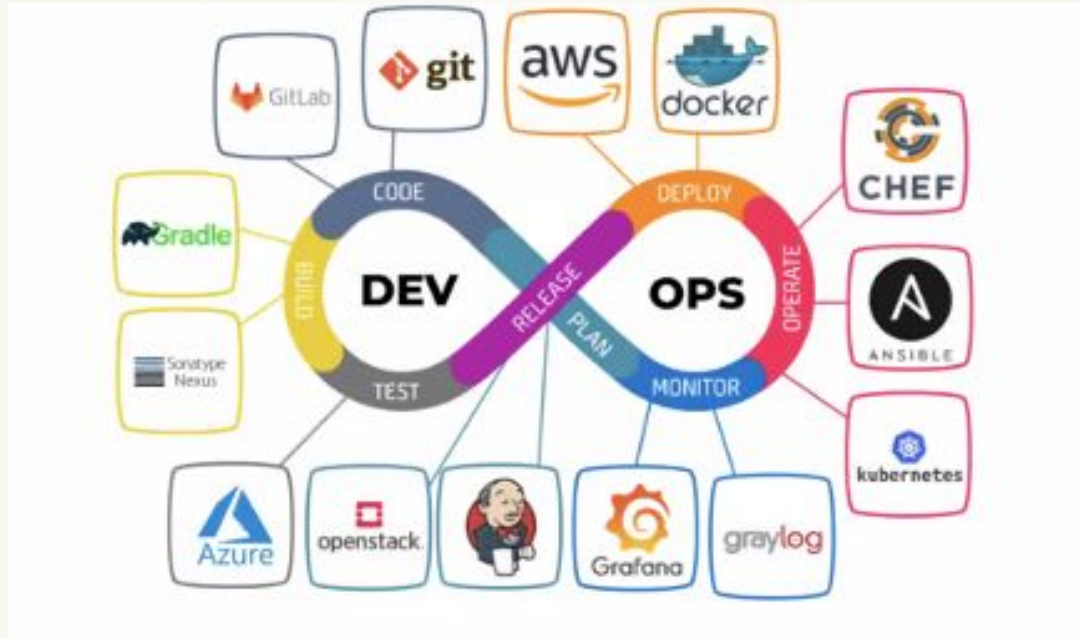


# Revisão DevOps



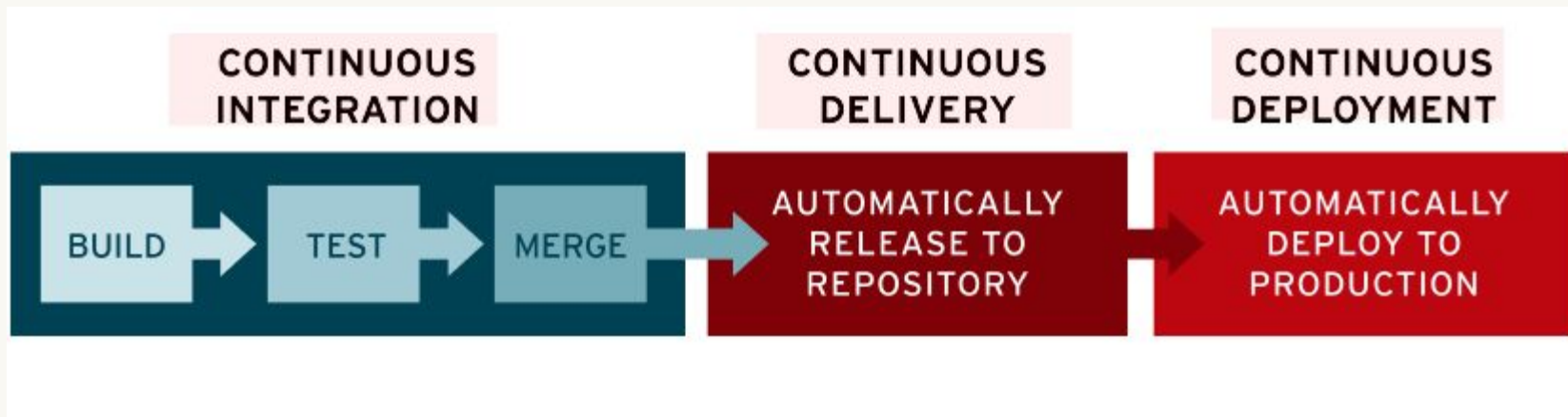


# Revisão DevOps

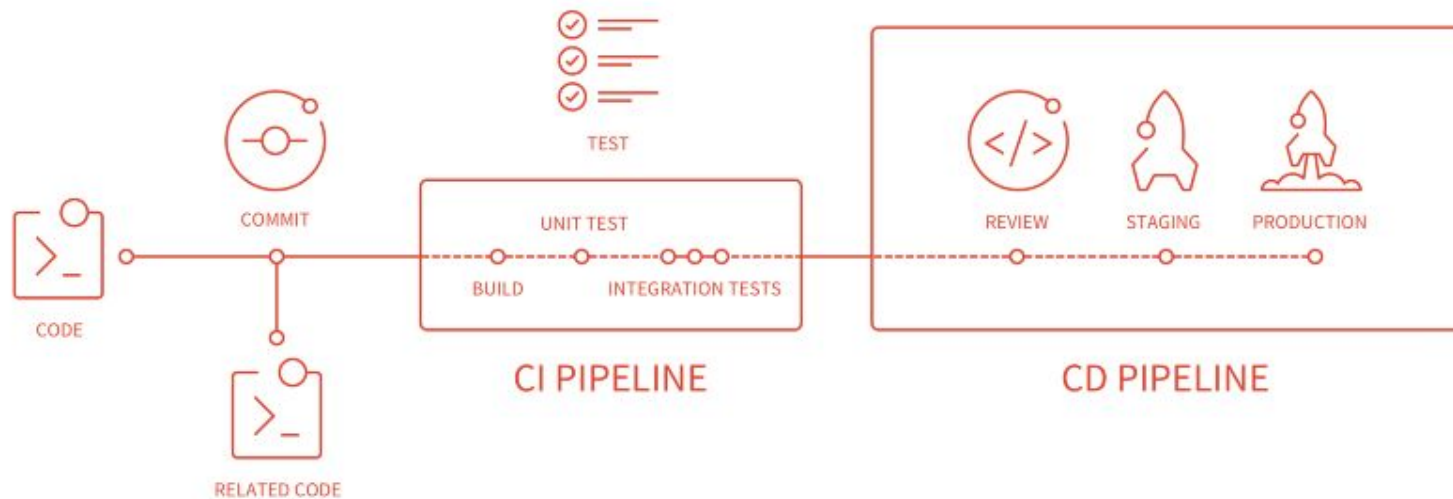


# Revisão Pipelines

- As pipelines **CI** e **CD** são automatização do processo de entrega de software;
- São uma sequência de passos que são utilizadas;
- É rápida, confiável e precisa.



# Revisão Pipelines



## GCP (Google Cloud Platform)

- Para a nossa nuvem onde será feito o deploy da nossa aplicação vamos usar o **GCP**. Portanto precisaremos criar uma conta na plataforma.

## Bitbucket

- O Bitbucket será a nossa ferramenta principal para exercitarmos os conceitos de **devOps**.



# DEVinHouse

Parcerias para desenvolver a sua carreira

**OBRIGADO!**



<LAB365>