

CI/CD Prática

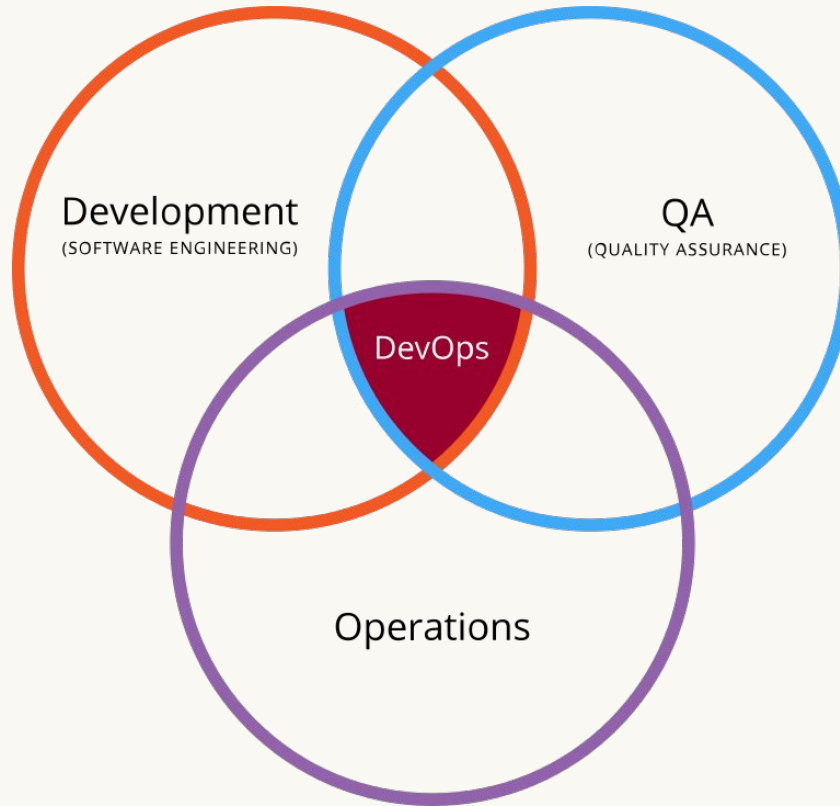
DEVinHouse

Parcerias para desenvolver a sua carreira

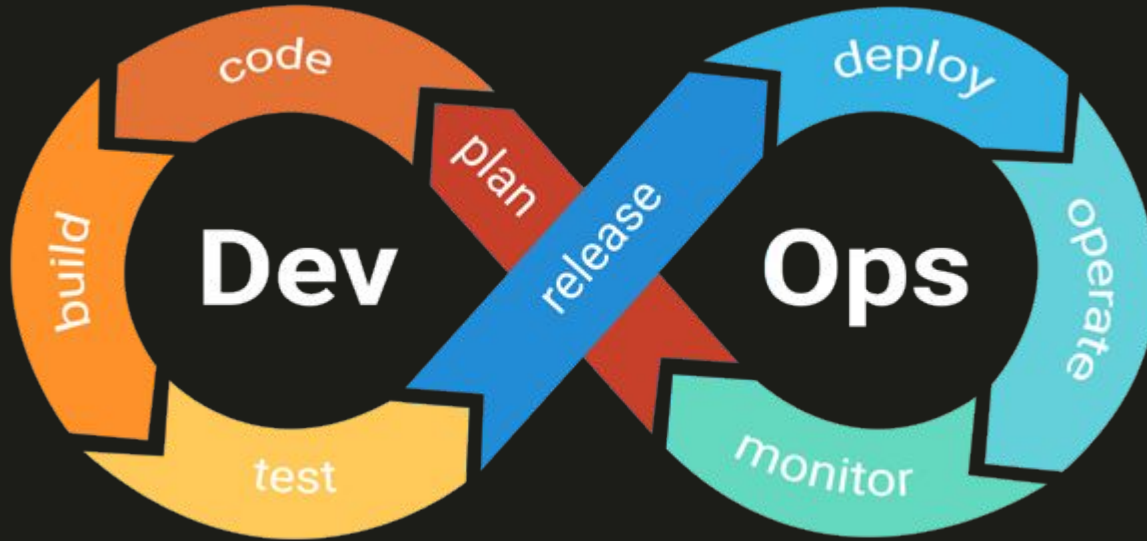
SENAI

<LAB365>

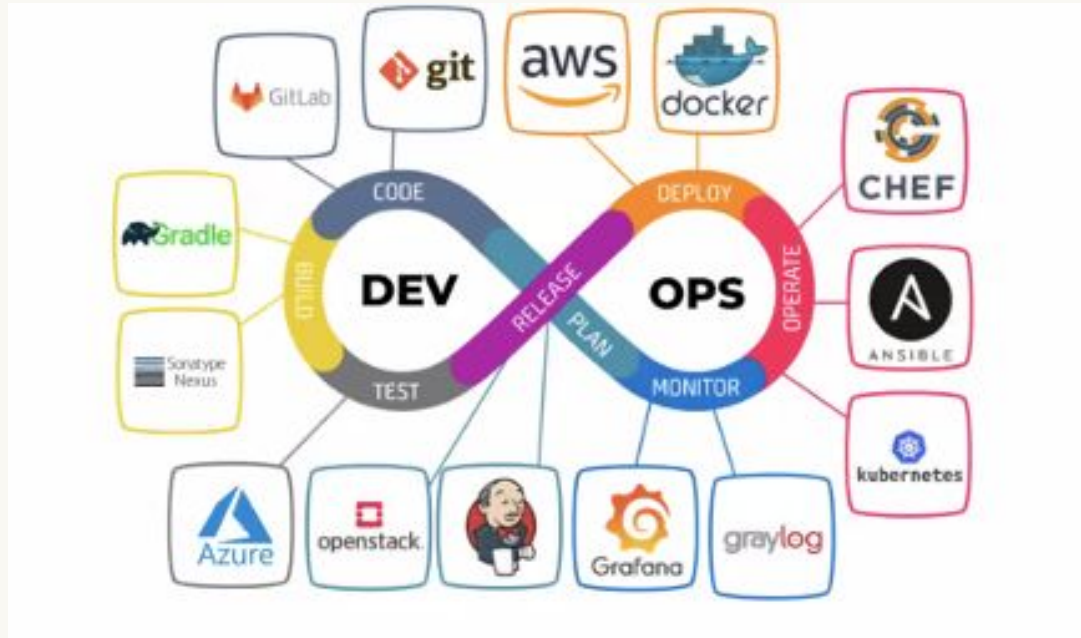
Revisão DevOps



Revisão DevOps



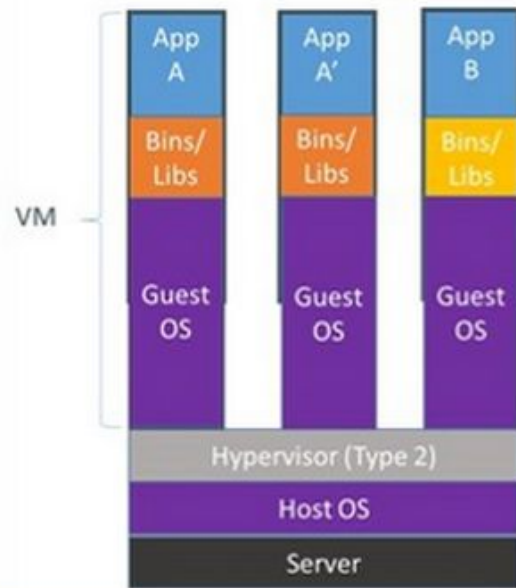
Revisão DevOps



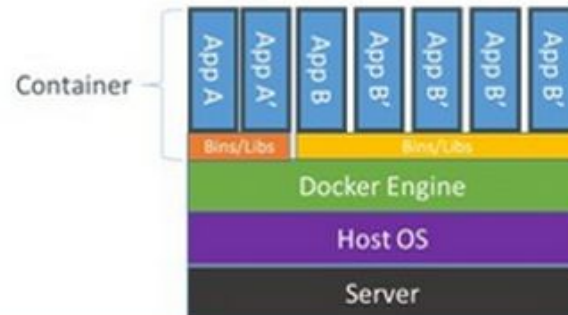
Containers

- Os **containers** compõe o provisionamento de uma aplicação de forma escalonável
- Entre os container temos o **Docker** como a ferramenta mais popular para criar serviços.
- Os containers precisam ser **orquestrados** por outros serviços, no caso os serviços da **GCP**, mas em um ambiente local podemos usar um orquestrador como o **Kubernetes**.

Containers vs. VMs



Containers are isolated, but share OS and, where appropriate, bins/libraries

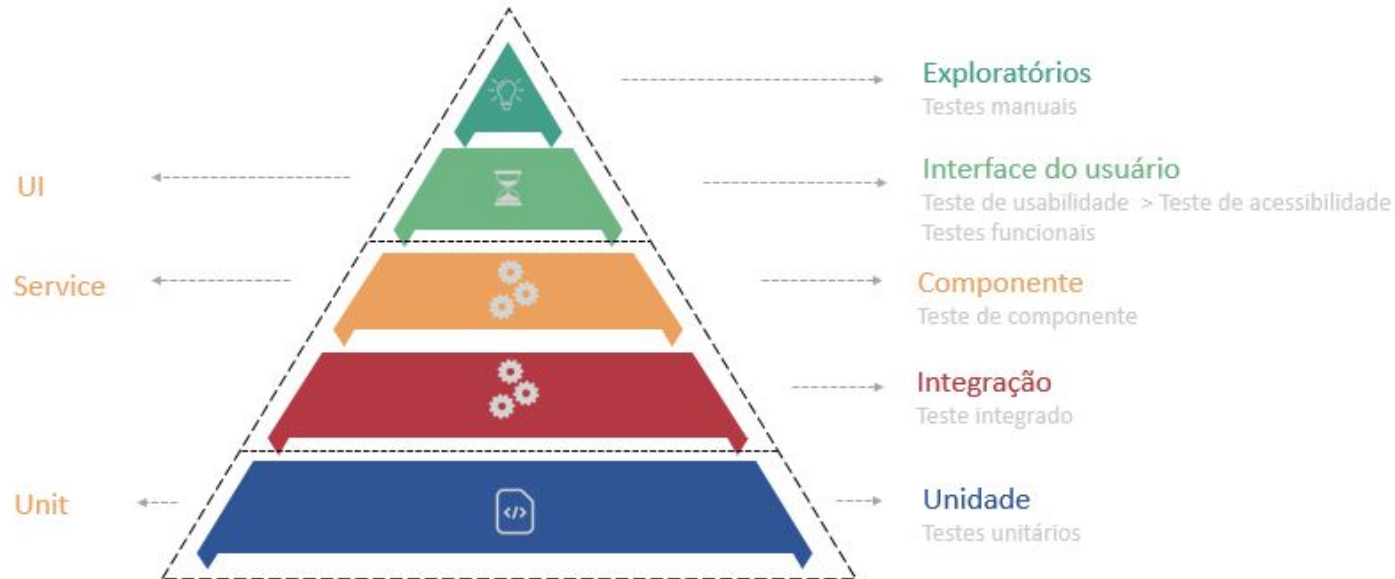


Revisão Introdução CI

- Continuous integration ou Integração contínua.
- É uma prática de **desenvolvimento de software** na qual você compila e testa software toda vez que um desenvolvedor envia código para a aplicação, e isso pode acontecer várias vezes ao dia
- Após enviarmos o código para um **repositório de versionamento** ele passa por um processo onde é feito o **Build**, após isso os testes são executados, entre esses podemos ter Unitários, de Integração, ou de qualquer outro tipo.

- Um dos pontos chave do **CI** é o **teste**, esse pode ter vários níveis de granularidade, mas deve ser feito tanto na esteira quanto antes de ser enviado para ela. Sendo assim o desenvolvimento é feito no modelo **TDD - Test Driven Design**
- A quantidade de testes recomendados segue um modelo de **pirâmide**, onde a base são os testes unitários e no topo temos os testes manuais ou de exploração
- Uma unidade no código depende do tipo de código desenvolvido, no caso de um programa Orientado a Objeto, essa unidade pode ser um método.

Revisão CI - Testes



- **Continuous Delivery** ou Entrega Contínua
- É uma abordagem de **engenharia de software** na qual a integração contínua, testes automatizados e recursos de implantação automatizada permitem que o software seja desenvolvido e implementado de forma rápida, confiável e repetitiva com intervenção humana mínima.
- A implantação na produção é definida **estrategicamente** e é acionada manualmente.

- Após realizarmos a parte de **Build** e **Testes** do **CI** podemos fazer a entrega de software em um ambiente, os mais comuns encontrados são:
 - **Desenvolvimento**
 - **Homologação**
 - **Produção**

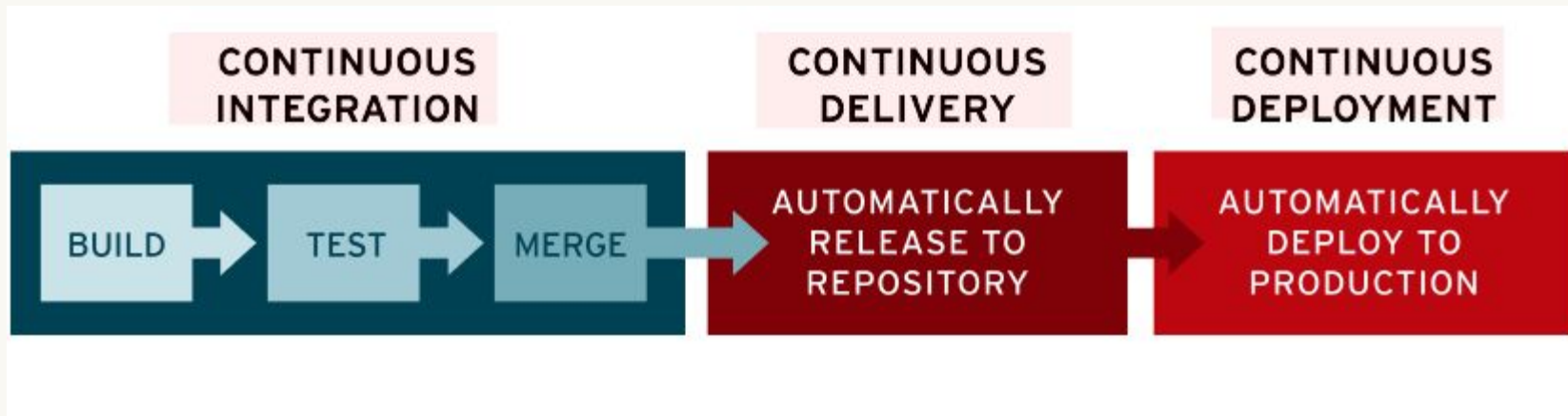
- **Desenvolvimento** - É onde colocamos o código para ser executado em um ambiente parecido com o de Produção pela primeira vez, porém o objetivo é apenas averiguar o funcionamento da aplicação.

- **Homologação** - É o ambiente mais próximo de Produção, sendo que os dados são feitos para serem o mais parecido possível com produção e é onde testamos o comportamento junto da funcionalidade da aplicação.

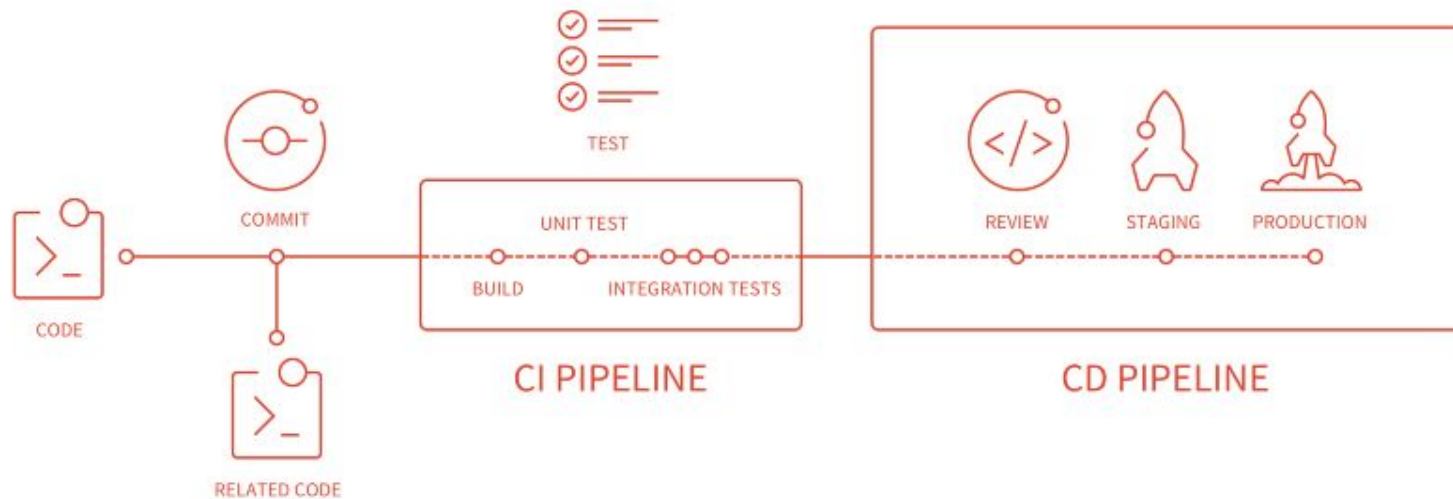
- **Produção** - É o ambiente que é acessado pelo usuário final, este pode ser interno ou externo a empresa, nessa fase nós apenas monitoramos o comportamento do serviço e verificamos se não há falhas no comportamento dele.

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>