**DEVOPS**

Sumário

[Introdução 3](#_Toc56627453)

[Jenkins 3](#_Toc56627454)

[Plug-in GitHub 4](#_Toc56627455)

[Plug-in Fortify 4](#_Toc56627456)

[Plug-in SonarQube 4](#_Toc56627457)

[Plug-in Jasmine 4](#_Toc56627458)

[Plug-in Cucumber 4](#_Toc56627459)

[Plug-in Nexus 5](#_Toc56627460)

[Plug-in ServiceNow 5](#_Toc56627461)

[Plug-in Heroku 5](#_Toc56627462)

[Plug-in XL Deploy 5](#_Toc56627463)

[Plug-in XL Release 5](#_Toc56627464)

[Plug-in Docker 5](#_Toc56627465)

[Pipelines 5](#_Toc56627466)

[PipeLine Blue 6](#_Toc56627467)

[Pipeline Green 6](#_Toc56627468)

[Canary Deploy 6](#_Toc56627469)

[Conclusão 7](#_Toc56627470)

Introdução

Neste trabalho estamos descrevendo a proposta para um pipeline de Delivery Continuo e Integração Continua para o projeto de AM Doe+.

Nele descrevemos a estratégia e as ferramentas que iremos usar.

# Jenkins

Jenkins é um servidor de Integração Contínua open-source feito em Java, pode ser rodado de forma standalone ou como uma aplicação web dentro de um servidor web.

Com ele é possível realizar testes automatizados, analise de códigos, identificar e corrigir erros de programação com antecedência, fácil de operar e configurar.

Com o Jenkins possui grande integração com diversos plug-ins que o deixam mais adaptado ao projeto.

Em nosso projeto especificamos alguns plug-ins importantes para garantir a qualidade e segurança do nosso portal. São eles:

Plug-in GitHub

Este plugin integra ao Jenkins com projetos do Github. Atualmente, o plugin tem três funcionalidades principais que nos ajuda no controle do código e na automação.

* Crie hiperlinks entre seus projetos do Jenkins e o GitHub.
* Disparo de um job ao realizar um push para o repositório através do recurso hook, iniciando o pipeline automaticamente.
* Report do status do Build de volta para o GitHub.

### Plug-in Fortify

O Fortify on Demand é o serviço de testes de segurança de aplicativos baseados em nuvem. Este plug-in permite que do Fortify façam SAST (Static Application Security Testing) diretamente do Jenkins.

### Plug-in SonarQube

O SonarQube oferece a capacidade de mostrar não apenas a integridade de um aplicativo, mas também de destacar problemas recém-introduzidos. Com um Quality Gate no lugar, você pode consertar o vazamento e, portanto, melhorar a qualidade do código sistematicamente.

### Plug-in Jasmine

O Jasmine é uma estrutura de desenvolvimento orientada por comportamento para testar o código JavaScript. Não depende de nenhum outro framework JavaScript. Não requer um DOM. E tem uma sintaxe clara e óbvia para que você possa escrever facilmente testes.

### Plug-in Cucumber

Cucumber é uma ferramenta de software que os programadores de computador usam para testar outros softwares. Ele executa testes de aceitação automatizados escritos em um estilo de desenvolvimento orientado a comportamento (BDD). Pepino é escrito na linguagem de programação Ruby. Projetos de pepino estão disponíveis para outras plataformas além do Ruby.

### Plug-in Nexus

Nexus é um gerenciador de repositório.

### Plug-in ServiceNow

O ServiceNow utilizaremos para gerenciar o fluxo de aprovação necessária para aprovar a entrada em produção.

### Plug-in Heroku

Serviço de Plataforma como serviço onde vamos hospedar nossos sistemas.

### Plug-in XL Deploy

Aplicação para deploy de release automatizado.

### Plug-in XL Release

Visibilidade, automação e orquestração dos pipelines e releases.

### Plug-in Docker

Docker é uma tecnologia de software que fornece contêineres

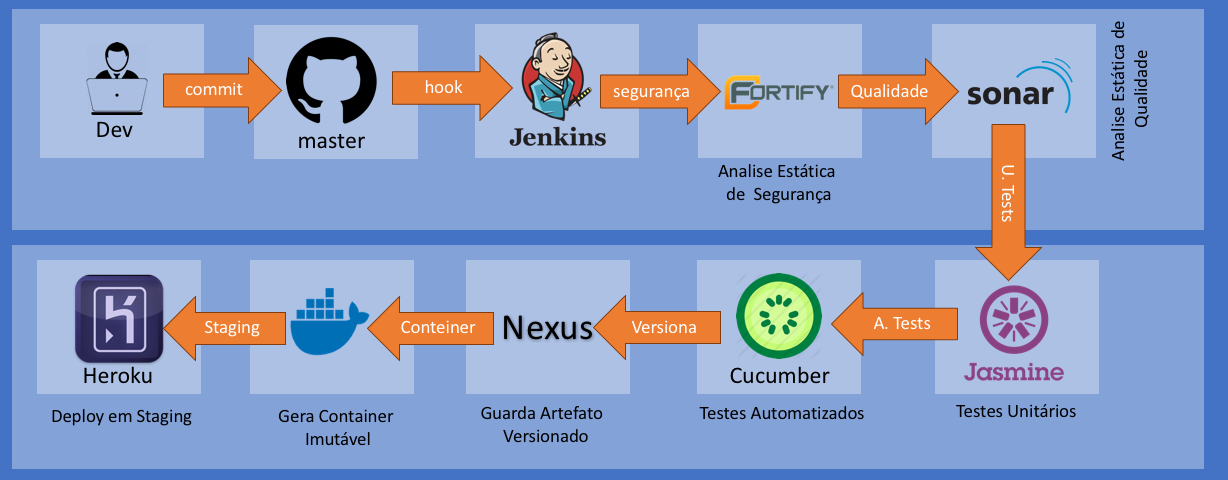
# Pipelines

Fizemos um estudo sobre qual sistemas de deployment usaríamos, o Blue-green ou Canary.

Tínhamos como premissa deixar mais automatizado possível, pois nosso time é pequeno e devemos dedicar nossos esforços no desenvolvimento e não no gerenciamento da infraestrutura.

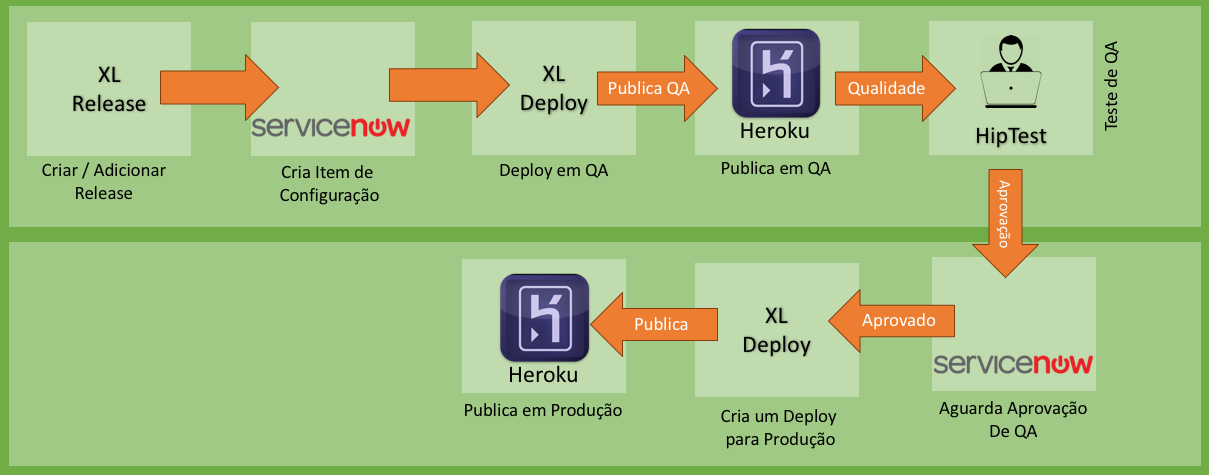
Criamos os dois pipelines e concluímos que o que mais nos atenderia é o blue-green.

## PipeLine Blue



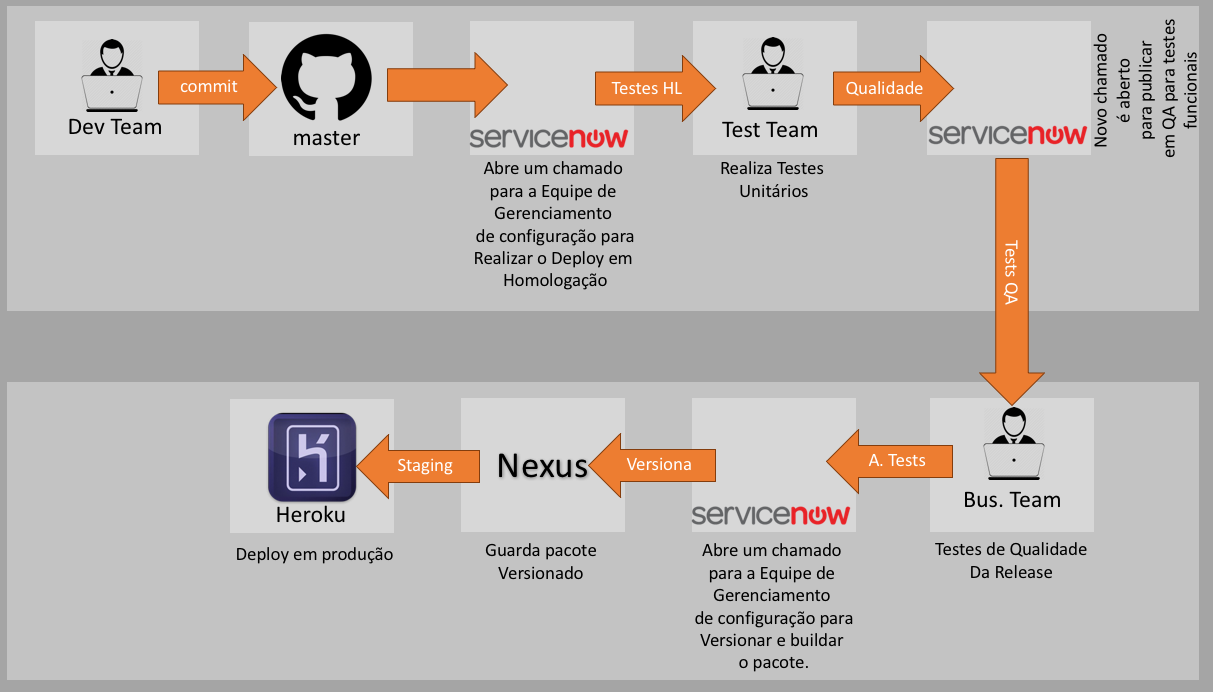
Qualquer falha no pipeline deverá interromper o mesmo e aguardar a correção do desenvolvedor, neste momento não é criado um bug em caso de erro.

## Pipeline Green



O pipeline Green vem logo após o final do pipeline Blue.   
No Green, ainda temos um passo de aprovação para entrar em produção.

## Canary Deploy



No Canary deploy, toda a interação com o ambiente de homologação, Qualidade e Produção é humana e depende de um fluxo de aprovação e transição entre áreas de desenvolvimento e infraestrutura.

# Conclusão

A estratégia de que adotamos foi a de blue/green por permitir uma estratégia de entrega continua em produção reduzindo o tempo do ciclo de aprovações para a publicação de release e um rápido.