



Serviço Nacional de Aprendizagem Industrial

PELO FUTURO DO TRABALHO

Aula 03 - Git flow e Trabalho prático

Professor: Rafael Ventura

O que iremos aprender:

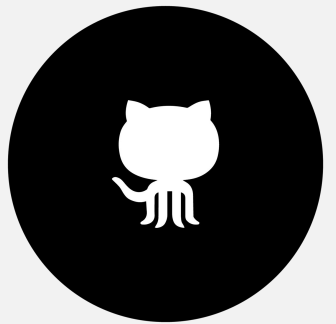
- ❖ Padrão de projeto Gitflow

Referências

- ❖ Git: <https://git-scm.com/doc>
- ❖ GitHub: <https://docs.github.com/pt>
- ❖ Gitflow :
<https://datasift.github.io/gitflow/IntroducingGitFlow.html>

O que é GitFlow

- ❖ É um modelo, uma estratégia, ou ainda um fluxo de trabalho muito utilizado por equipes de desenvolvimento de software.
- ❖ Ela se destaca por auxiliar na organização do versionamento de códigos.



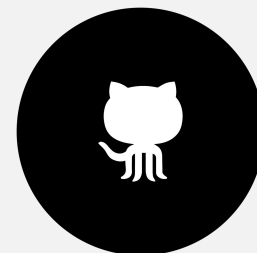
Quando utilizar o Git Flow?

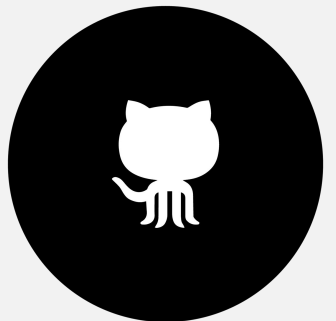
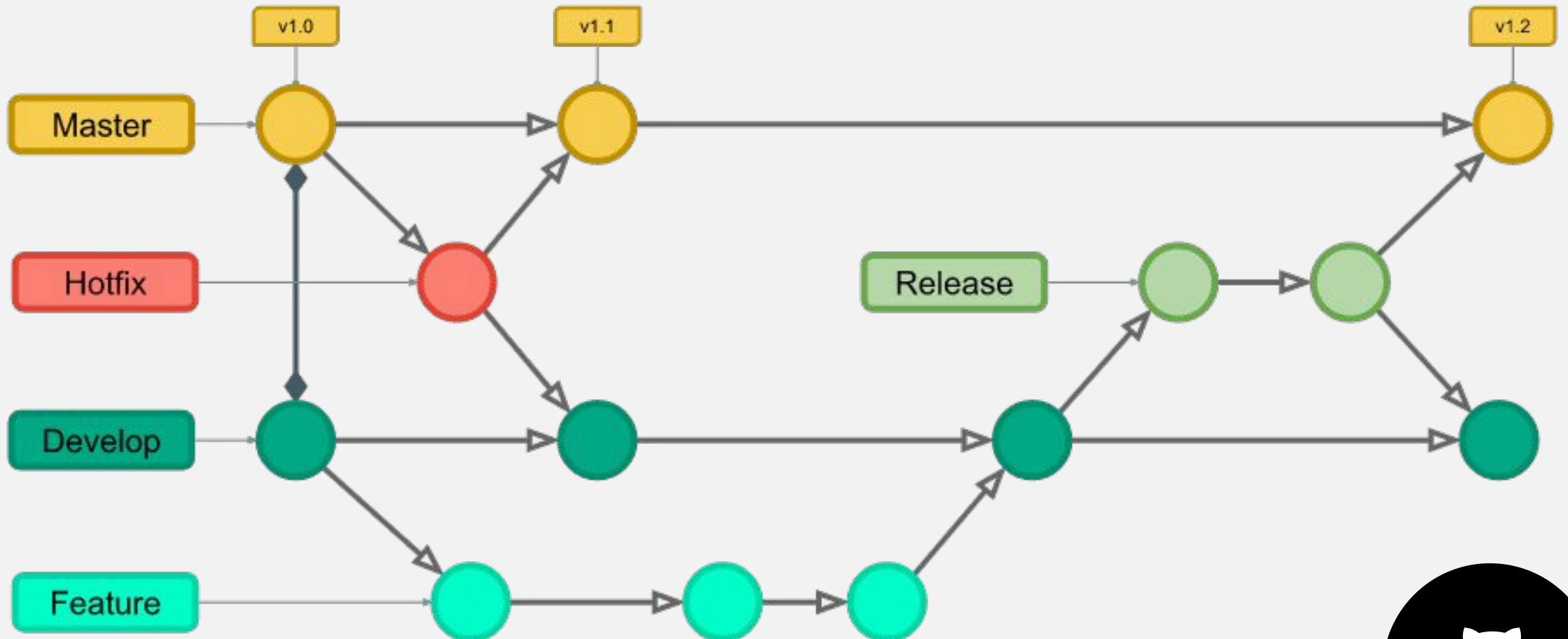
- ❖ Projetos que possuem uma grande quantidade de colaboradores “commitando” dentro de um repositório.
- ❖ Projetos que utilizam o versionamento semântico



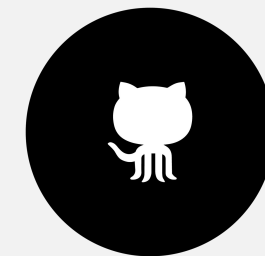
Como funciona o Git Flow?

- ❖ O Git Flow trabalha com duas branches: DEVELOP e MASTER (MAIN) que tem duração contínua.
- ❖ Ainda existe três branches de suporte: FEATURE, RELEASE e HOTFIX que são temporários e duram até realizar o merge com as branches principais.





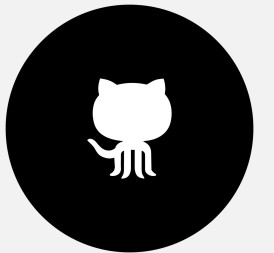
Branch Master/Main

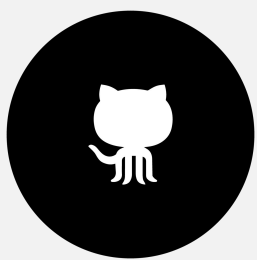


- ❖ Principal branch, aqui é onde temos o código em produção.
- ❖ Todas as novas funcionalidades que estão sendo desenvolvidas, em algum momento, serão mescladas ou associadas a Master (Main).
- ❖ Só interagimos com essa branch através de hotfix ou uma release

Branch Develop

- ❖ É branch onde fica o código do próximo deploy.
- ❖ Ela serve como uma linha do tempo com os últimos desenvolvimentos.
- ❖ Ela possui funcionalidades que ainda não foram publicadas.



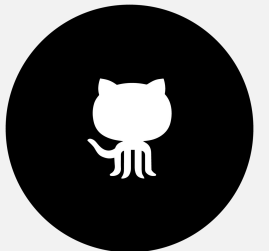


Branch Feature

- ❖ São branches utilizadas para o desenvolvimento de funcionalidades específicas,
- ❖ Geralmente usam uma convenção para o nome feature/nome-da-feature.
- ❖ São criadas sempre a partir da branch Develop.
- ❖ Quando finalizadas são removidas após o merge com a Branch Develop.

Branch Hotfix

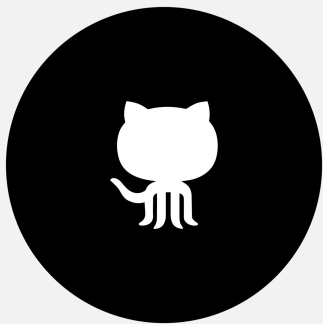
- ❖ É uma branch criada a partir da Master para realizar correções imediatas encontradas no sistema de produção.
- ❖ Quando concluída, ela é excluída após realizar o merge com as branches Master e Develop.



Branch Release

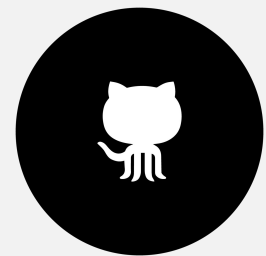


- ❖ Uma vez que a etapa de desenvolvimento esteja concluída, teremos em nossa Branch Develop todas as features e Hotfix mesclados.
- ❖ A branch Release serve como ponte para fazer o merge da Develop para a Master.
- ❖ Ela funciona como um ambiente de homologação e é removida após realizar os testes do merge com a Master.
- ❖ Caso seja encontrado algum bug e haja alguma alteração, ela também deve ser sincronizada com a Develop.



Trabalho Prático

- ❖ Separados em grupos crie uma aplicação em HTML, CSS e JS simples.
- ❖ Utiliza como padrão de versionamento o modelo git flow
- ❖ Crie um repositório remoto e gerencie as versões da sua aplicação.
- ❖ Crie um Readme com a documentação da sua aplicação.
- ❖ Compartilhe o link como entrega do trabalho.



Observações

- ❖ Se atente ao modelo gitflow
- ❖ Pode tirar dúvidas pontuais com o professor
- ❖ Caso prefiram trabalhar com vários repositórios locais, não se esqueçam de sincronizar os locais com o repositório remoto, usando o `git remote add origin <link>`
- ❖ Caso seja preciso adicione os colaboradores ao seu repositório remoto.
- ❖ Não esqueça de fazer as Tags sempre que for mergear na Master
- ❖ Não se esqueça que só existem duas branches que mergeia com a Master (hotfix e release)





Serviço Nacional de Aprendizagem Industrial

PELO FUTURO DO TRABALHO

0800 048 1212     **sc.senai.br**

Rodovia Admar Gonzaga, 2765 - Itacorubi - 88034-001 - Florianópolis, SC