Apresentação Git + Git flow

1. O que é o Git?
2. O que é Git Flow?
3. Exemplos

Por onde começar?

Quando estamos trabalhando de forma **colaborativa** em projetos de desenvolvimento de software, parte do desafio é **gerenciar as versões do código** desenvolvido entre os programadores e podemos concluir que eventualmente problemas podem ocorrer se nenhuma iniciativa for tomada para solucionar isso.

"É como tentar pintar um quadro com várias pessoas ao mesmo tempo, se feito de forma desordenada, podemos ter um resultado nada satisfatório."

1 - O que é Git?

Git é o sistema de controle de versão mais utilizado no mundo. É um projeto de código aberto desenvolvido por Linus Torvalds (Criador do kernel do Linux) que é utilizado para controlar as versões localmente ou de forma distribuída, garantindo outros pontos como:

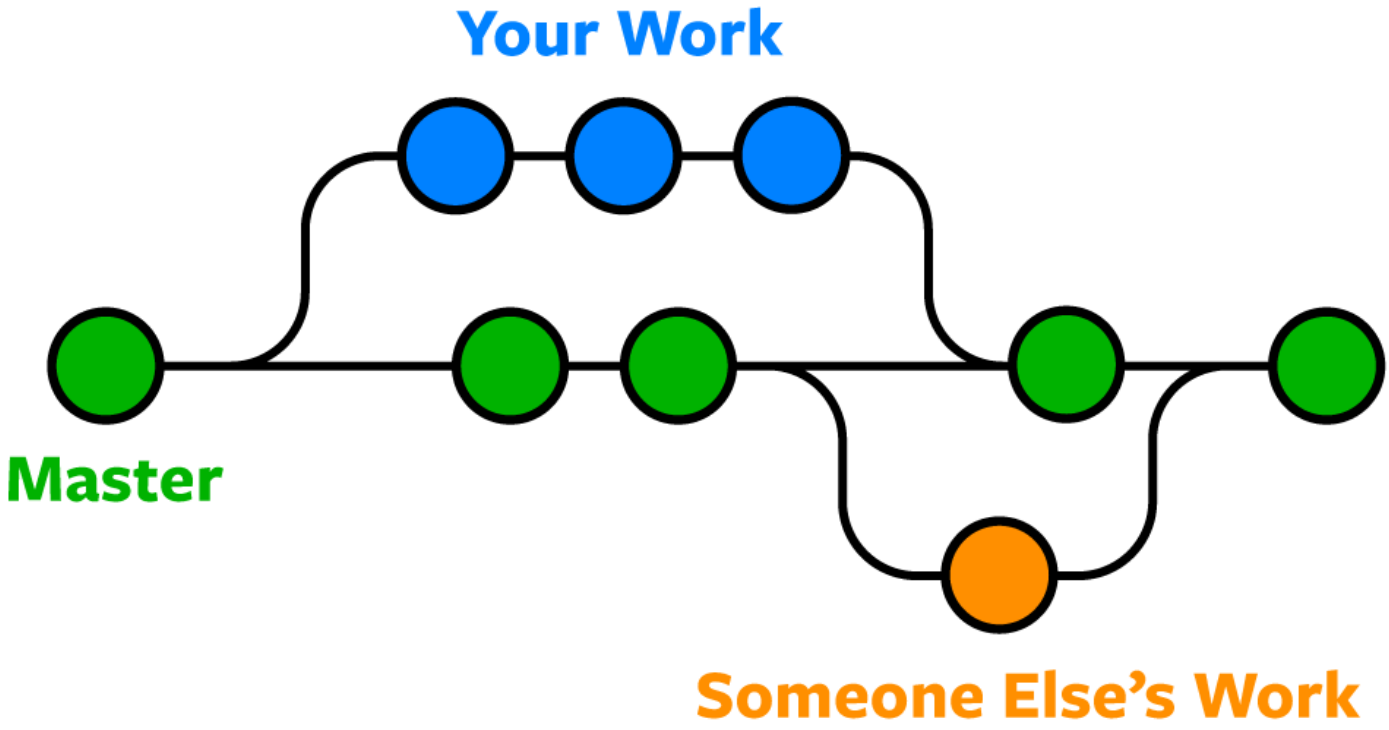
- Desempenho
- Segurança
- Flexibilidade

Controle de versões com Git

Hoje, o Git é a melhor escolha para a maioria das equipes de software. Embora cada equipe seja diferente e deva fazer a própria análise, aqui estão os principais motivos pelos quais o controle de versão com Git é preferido em vez de alternativas:

1. **Git é bom** - Em comparação com as opções de mercado, o git se destaca
2. **Git é um padrão de fato** - É a ferramenta mais adotada da categoria e tem muitas integrações prontas
3. **Git é um projeto de código aberto e de qualidade** - Excelente suporte da comunidade e uma administração sólida, documentação abundante, entre outros beneficios.

Ao utilizar o Git, percebemos a sua estruturação em ***branchs*** (ramos, filial), seria algo muito parecido como uma árvore, conforma o exemplo abaixo:



2 - O que é Git Flow

O Git Flow é uma abordagem, uma ideia abstrata do fluxo de trabalho utilizando o Git, ele dita como utilizar e controlar *branchs*, sendo esse método divulgado pelo seu criador Vincent Driessen apenas em 2010. Sua ideia é criar várias *branchs* cada uma com uma função:

Branchs principais

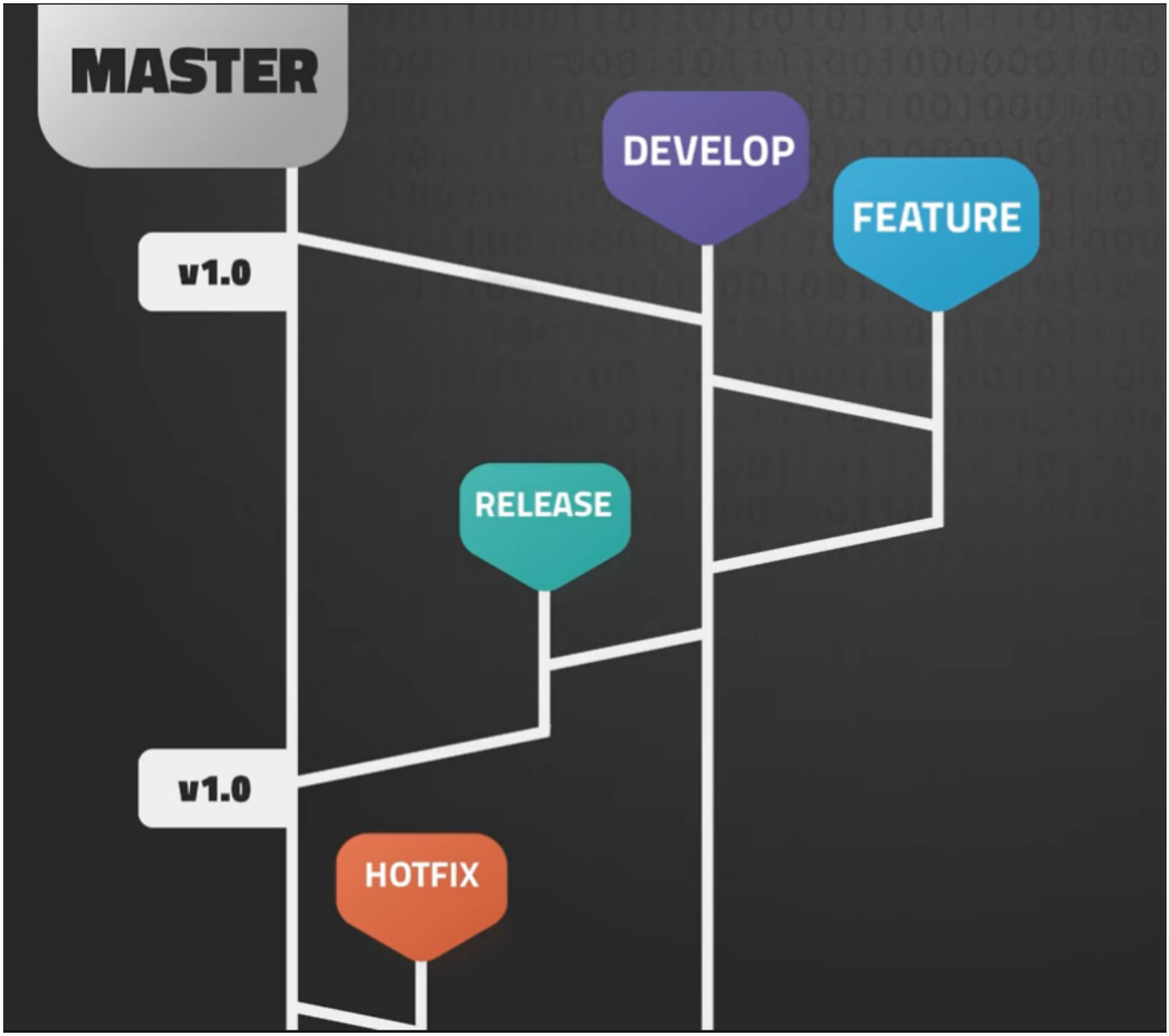
- *branch* '**master**' e ela vai conter todo código já testado, versionado e que será entregue ao cliente/produto.
- *branch* '**develop**' e é nela que todo trabalho deve ocorrer

Branchs suporte

- *branch* '**release**' que deve ser utilizada como um ponto antes da master
- *branch* '**feature**' que deve ser utilizada como um ponto antes do merge com a develop
- *branch* '**hotfix**' e é nela que as correções emergenciais devem acontecer

Neste modelo, os desenvolvedores criam uma ramificação de recurso e retardam o merge com a ramificação de tronco principal (**master**) até que o recurso esteja completo. As ramificações **feature** usam a ramificação **develop** como pai. Quando um recurso é concluído, ele passa por merge de

volta para a ramificação de desenvolvimento. Os recursos não devem nunca interagir direto com a ramificação **master**.



O fluxo geral do Gitflow é:

1. Uma ramificação develop é criada a partir da main
2. Uma ramificação de lançamento é criada a partir da ramificação de desenvolvimento
3. As ramificações de recurso são criadas a partir da ramificação de desenvolvimento
4. Quando um recurso é concluído, ele é mesclado na ramificação de desenvolvimento
5. Quando a ramificação release é feita, é feito o merge dela na ramificação develop e na principal
6. Se for detectado um item na main, uma ramificação de hotfix vai ser criada a partir da main
7. Depois que o hotfix for concluído, ele passa por merge para a ramificação develop e à main

Exploração
https://db1global.visualstudio.com/Plaenge/_git/developer-guide

https://db1group.github.io/gitflowanimated/

Referências:

https://www.atlassian.com/br/git/tutorials/comparing-workflows/gitflow-workflow
https://www.atlassian.com/br/git/tutorials/what-is-git
https://imasters.com.br/agile/fluxo-de-desenvolvimento-com-gitflow
https://www.youtube.com/watch?v=oweffeS8TRc
https://nvie.com/posts/a-successful-git-branching-model/