

Git, Github e Workflow

Quem faz o que e como

Conteúdo

- Git na teoria e na história
 - O que é git
 - História de git
 - Cómo funciona
- Exemplos de uso do git
 - Casos de êxito
- Workflow
 - O que é?
 - Exemplos
- Hubs para git
- Com o que continuar?

O que é git?

- Git é um sistema de controle de versão distribuído
 - Tem outros CVS (Control Version System), como Subversion
 - Dropbox e GDrive usam coisas parecidas
- Basicamente, git é um programa que se instala no computador
- Serve para ajudar a controlar qualquer tipo de arquivo, mas funciona muito melhor se são arquivos não-binários (é usado para edição de livros e código)
- Funciona totalmente offline



Porque usar git?

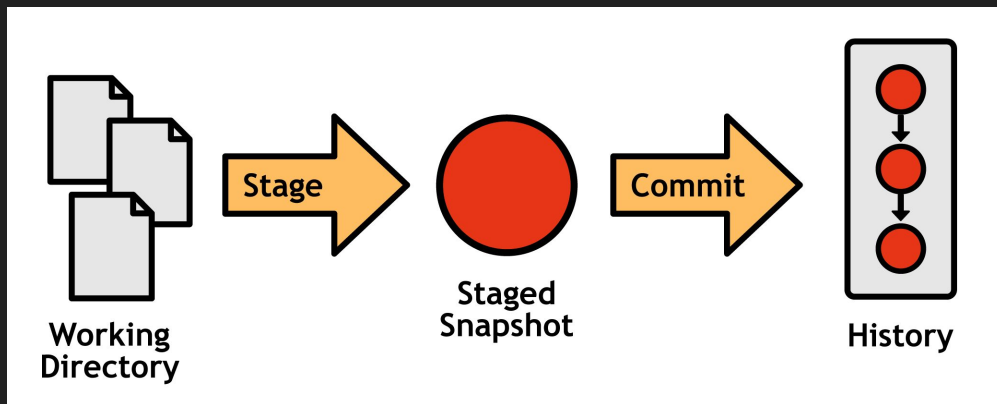
- Quando trabalhamos sozinhos:
 - Recuperar estados anteriores do trabalho
 - Testar mudanças destrutivas sem medo
 - Não ter milhares de linhas comentadas pelo “vai que”
 - Sincronizar trabalho entre computadores
- Quando trabalhamos em grupo:
 - Todo o do trabalho sozinho, mais:
 - Manter diferentes “estados” ou versões do projeto e poder compartilhar elas
 - Não ter pastas com do tipo “projeto_victor”
 - Resolver conflitos destrutivos entre mudanças
 - Coordenar como distribuir o trabalho

História de git

- Foi criado por Linus Torvalds em 2005 para administrar o desenvolvimento do kernel de linux.
 - Foi criado por um gênio da programação e mantido por uma comunidade muito capacitada
 - Milhares de pessoas trabalhando ao tempo (100.000 colaboradores únicos por mês)
- SVN não dava conta
 - SVN era o melhor livre e não dava conta
 - Tinha alguns privados, mas com limitações
- É muito rápido e confiável
 - MUUUUUUITO rápido
- Standard de facto
- É software livre

Como funciona

- Cada diretório é um repositório.
 - Git NÃO diz como trabalhar, só fornece ferramentas.
 - As ferramentas são comandos que fazem coisas, muitas sem efeitos visíveis
- Cada coisa em git pode se mover em 3 estados:
 - Diretório de trabalho (Working directory)
 - Área de preparação (Staging Area)
 - Histórico (History ou Commits)



Como funciona

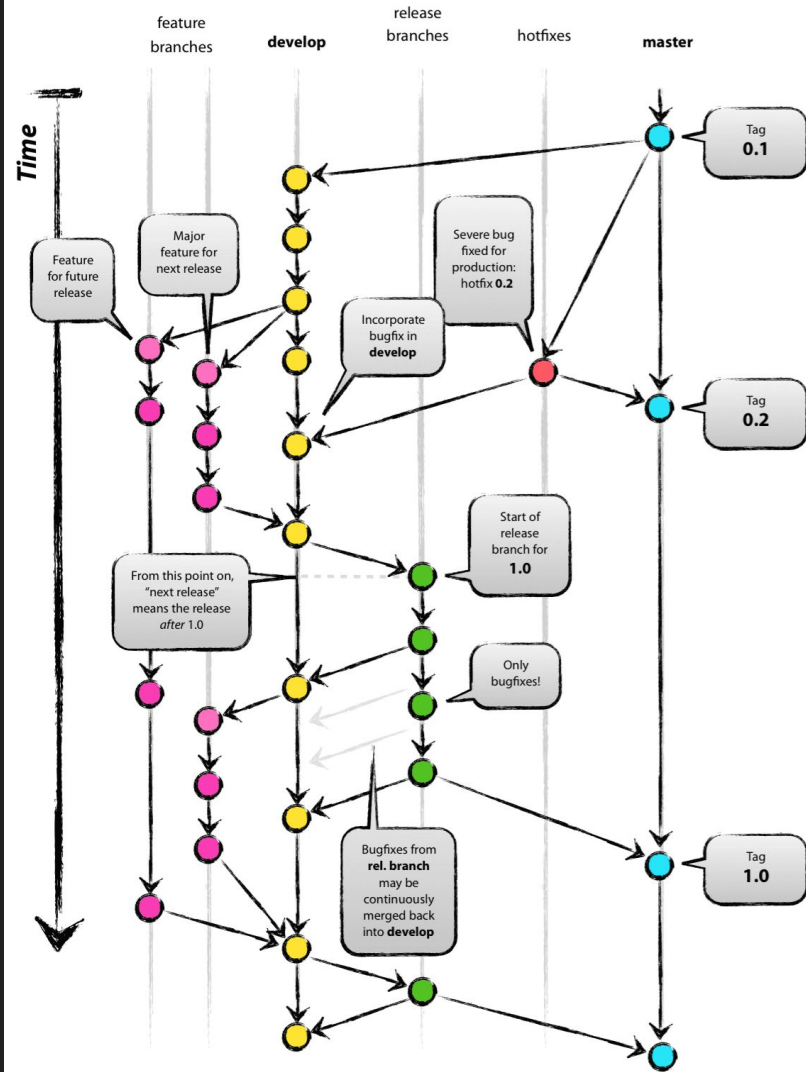
- Existem muitos comandos, mas vamos nos focar nos seguintes:
- Primeira parte:
 - log
 - status
 - commit
 - add
 - checkout
- Segunda parte:
 - branch
 - checkout
 - merge
 - rebase
 - tag
- Terceira parte:
 - push
 - pull
 - clone

Exemplos de uso do git

- Primeiro vamos a fazer um uso simples, para isto vamos usar 3 ferramentas:
 - Git: pode baixar em <https://git-scm.com/downloads>
 - Gitkraken (opcional): pode baixar em <https://www.gitkraken.com/>
 - Um editor qualquer (vou usar gvim)
- As condições deste exemplo são:
 - Queremos fazer um código simples, com algumas funções
 - Vamos fazer código real
 - O Gitkraken não é necessário, mas vamos usar para poder ver o que está acontecendo
- O segundo vai ser um exemplo de uso de branch, para o qual vamos usar um simulador: [Visualizing Git](#)

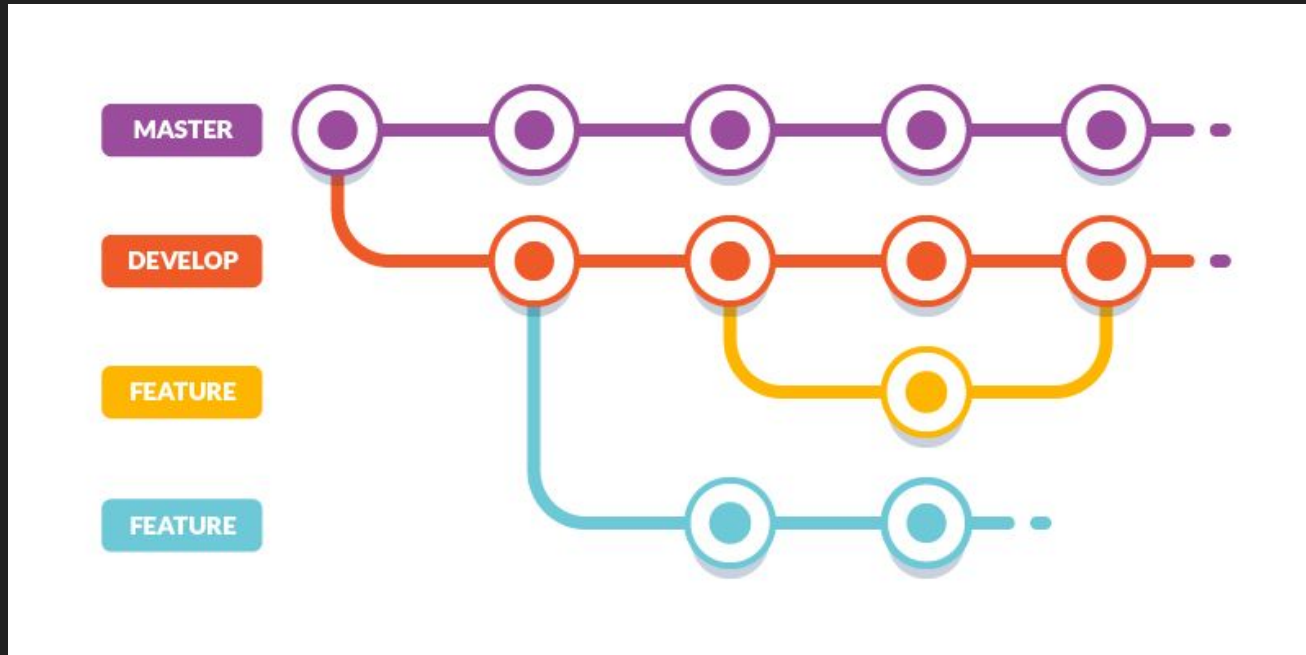
Workflow

- Git da ferramenta, não diz como trabalhar
- Existem muitos jeitos de trabalhar com git e é costume que cada empresa defina seu próprio jeito
- No 2010 alguém propôs o que ficou conhecido como *The Git Flow*: [A successful Git branching model](#)



Workflow

- Git flow simplificado:



Hubs para git

- Um dos melhores usos que podemos dar a git é fazer que um dos seus nós seja remoto
- github, bitbucket, gitlab
- Vamos a ver um exemplo de uso com servidor remoto

Com o que continuar?

- Um referência gráfica de git: [A Visual Git Reference](#)
- Um livro excelente: [Home // Think Like \(a\) Git](#)
- Um aplicativo para treinar: [Learn Git Branching](#)