



NTT DATA

BECA MOBILE

ios

Git

**FUTURE
AT HEART**

Lucas Leitão / Jonathas Moraes/ Ricardo Santana
2022

Sumário

- O que é Git
- Status de uma alteração no Git
- Gerenciamento de Branches
- GITIGNORE
- Comandos úteis
- Links úteis

O que é Git?

O que é Git?

Git é um projeto open source de sistema de controle de versões distribuído, criado por Linus Torvalds.

Por que usar o git?

- Ferramenta simples
- Facilita a integração de Código criado em diferentes locais
- Padrão de mercado

Conceitos de git

Repository: É o local onde suas alterações serão adicionadas. Aqui temos repos remotos e locais. O repo local está apenas na sua máquina, enquanto o remoto está em alguma ferramenta de controle de versão online, por ex: GitHub, GitLab e BitBucket.

Branches: São ramificações criadas dentro dos repos, cada branch contém as alterações feitas apenas nela, e são independentes entre si.

Merge: É utilizado para incluir alterações de outras branches

Clone: É utilizado para clonar o repo remoto na máquina que você está utilizando

Fork: Utilizamos o Fork quando queremos criar um repo separado do original

Como funcionam alterações no git?

Alterações no git

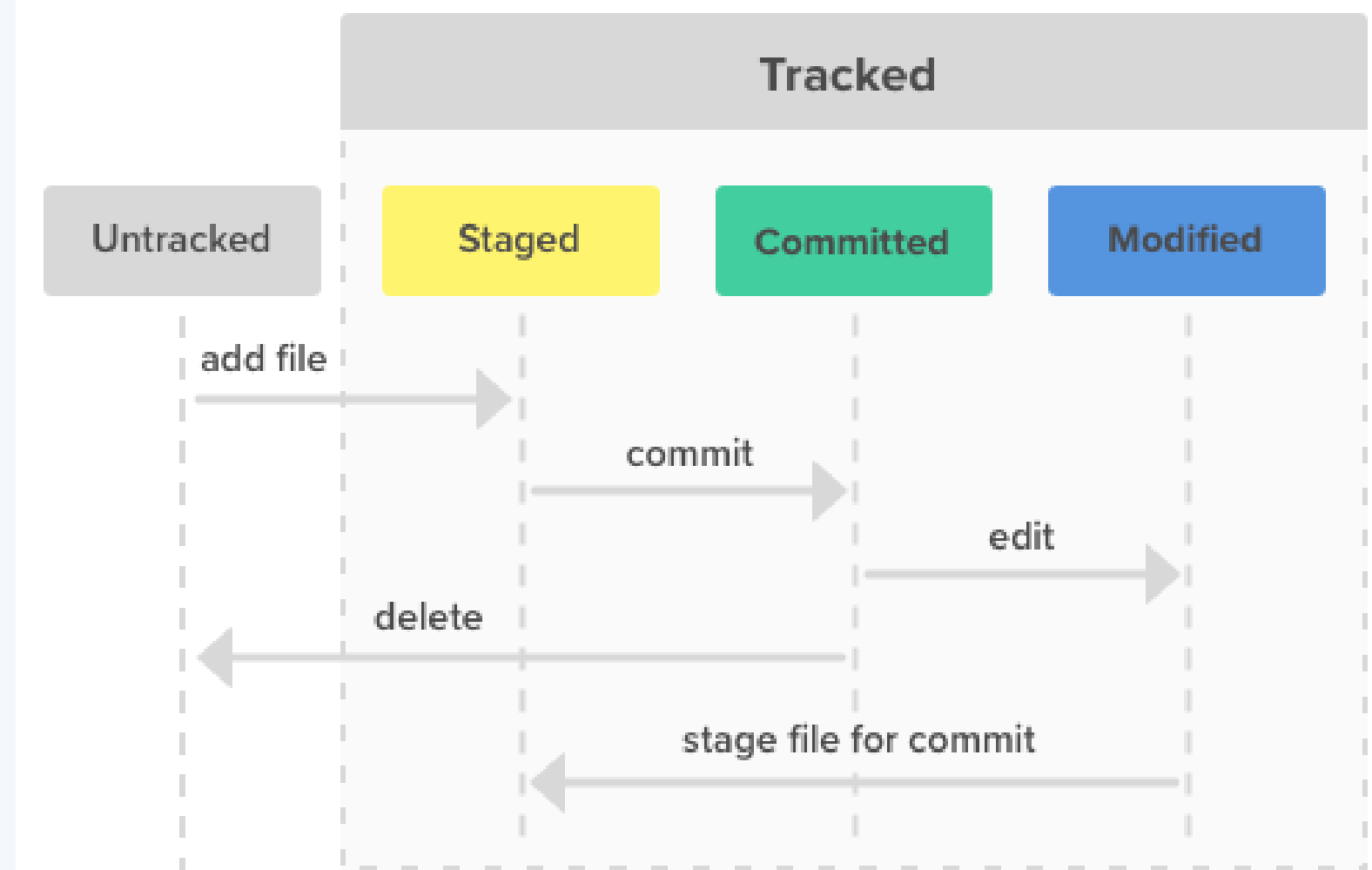
Dentro do git temos alguns estados de alterações, são eles: Untracked, Unmodified, Modified e Staged

Untracked : Estado inicial, quando algum arquivo não foi incluído no git ele será untracked. Este arquivo precisa ser adicionado ao git para ser enviado ao repo local

Unmodified : Neste estado não há alterações em relação ao que está no repo local.

Modified: Quando algo foi alterado em um arquivo no projeto, este irá aparecer como modified, suas mudanças precisam ser adicionadas para serem enviadas num commit

Staged: Neste momento as alterações feitas estão prontas para serem enviadas para o repo local através de um commit.



Alguns comandos

`git status`: Ver o estado de todos arquivos, exibe em vermelho os arquivos untracked e modified, em verde os arquivos staged

`git add <Nome do arquivo>`: Adiciona o arquivo se estiver untracked, se o arquivo já estiver presente no git, e contiver mudanças (modified), as mudanças feitas serão levadas para o estado de staged

`git commit -m "Mensagem de commit"`: Envia as alterações de staged para o repo local com uma mensagem, as alterações incluídas devem ser descritas nessa mensagem

`git push`: Envia os commits do repo local para o repo remoto

Gerenciamento de branches

Gerenciamento de branches no git

Normalmente temos mais do que uma branch, e mais de uma pessoa trabalhando num repo de projeto, por isso é importante sabermos como gerenciar nossa branch e como manter nosso trabalho sincronizado com outras pessoas.

Alguns comandos

`git pull`: Trazer as alterações feitas no remoto para o local e fazer o merge dessas alterações na branch atual

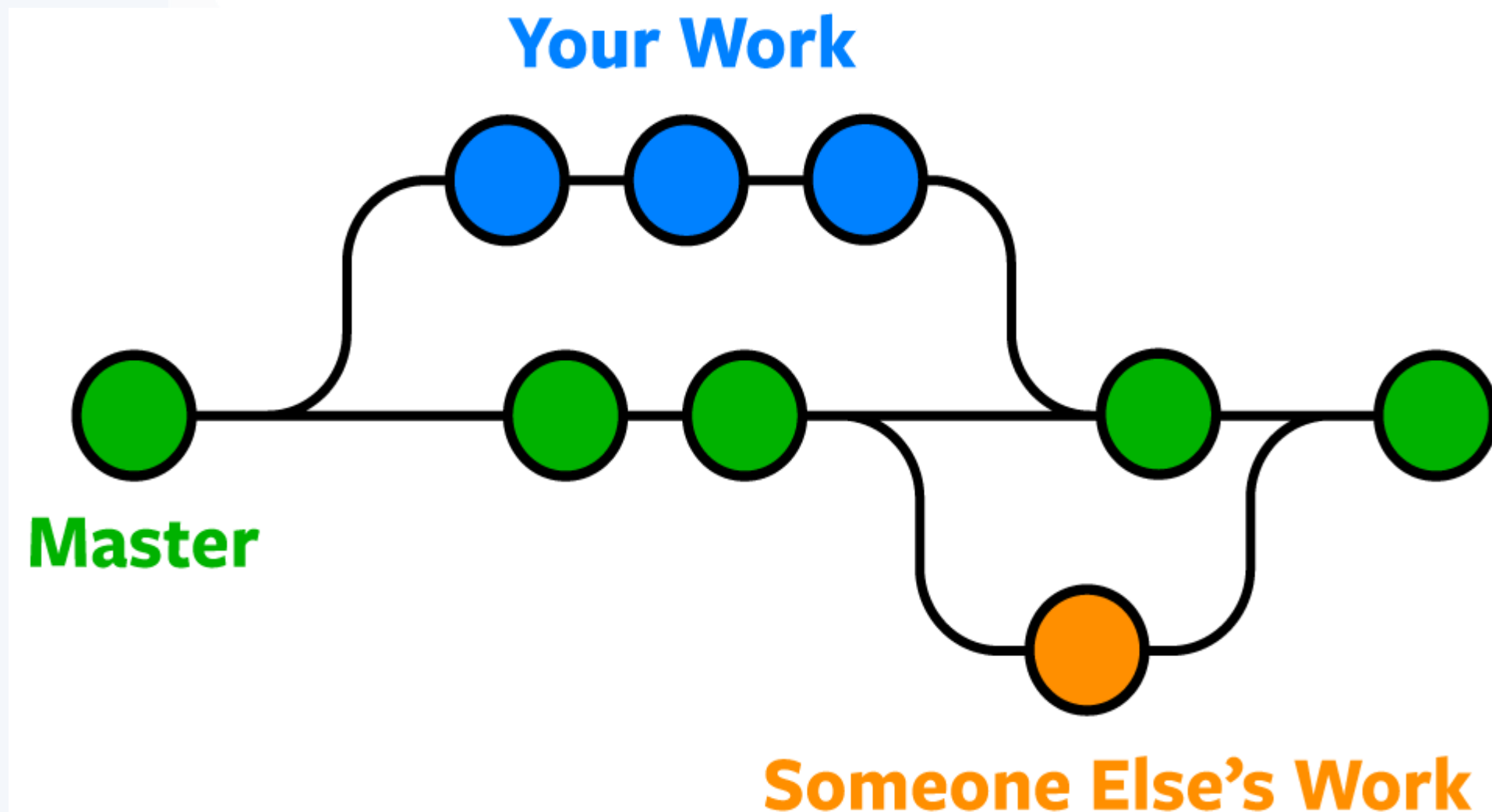
`git branch`: Ver outras branches

`git merge <outrabranh>`: Trazer alterações de outra branch para a atual

`git clone <repo>`: Criar um clone do repo remoto na máquina

`git fetch`: Trazer as alterações feitas no remoto para o local

`git checkout`: Mudar para outra branch



Git Flow

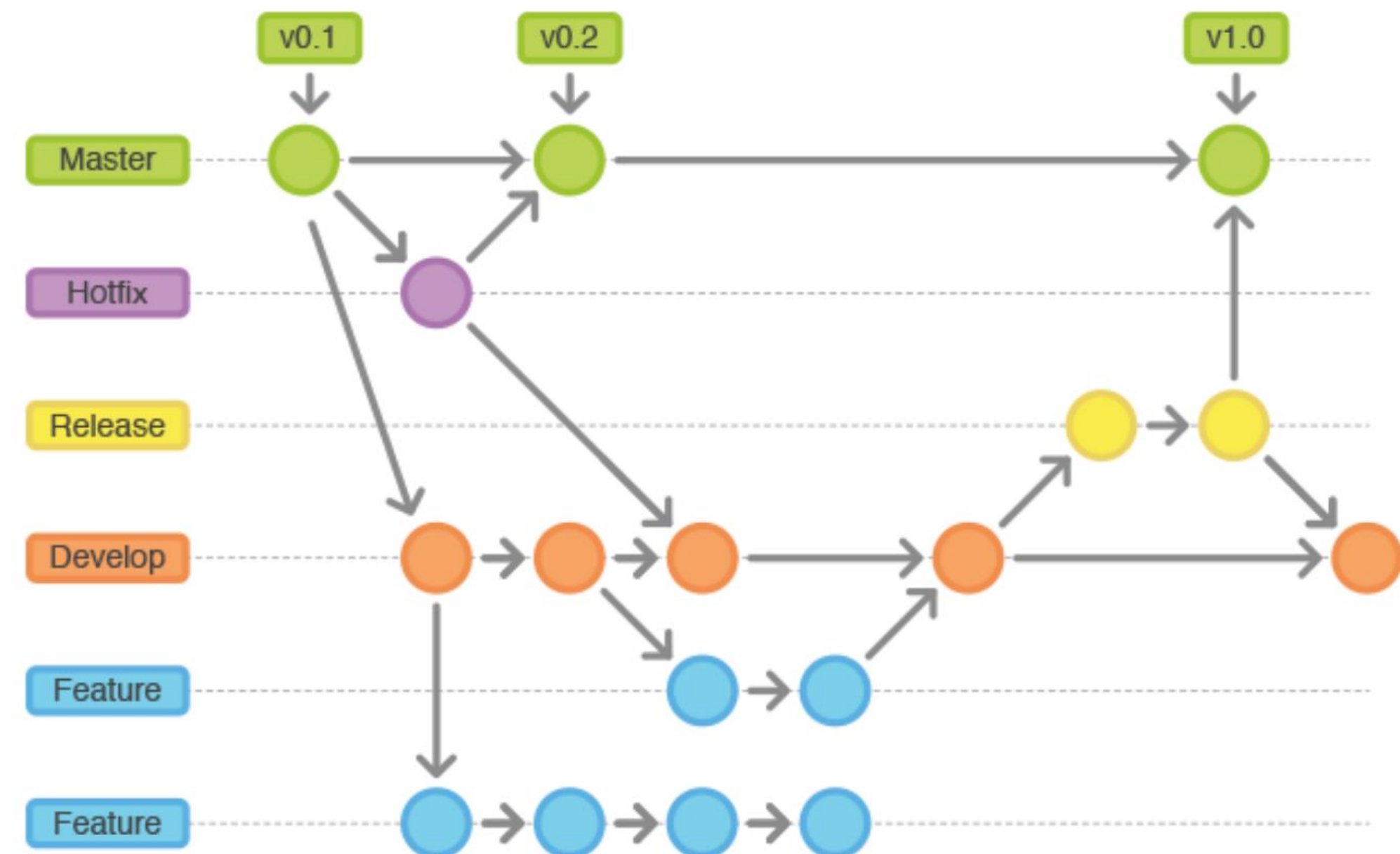
Divulgado pela primeira vez em 2010, pelo engenheiro de software, Vicent Driessen É um workflow implementado para auxiliar na organização do versionamento das Branches

Basicamente o git flow é composto em 2 grupos de Branches:

- Principal: Master e Develop
- Suporte: Feature, Release e Hotfix

Nota: Para fazer merge em branches nas ferramentas remotas utilizamos Pull Request(GitHub) ou Merge Request(GitLab), eles .

Para abrir um PR ou MR depende um pouco da plataforma, mas normalmente existe uma aba com “merge” ou “Pull requests” na qual é possível criar novos MRs e PRs. Eles são de extrema importância para organizar e gerenciar mudanças de código e é nesse ponto que o code review normalmente é feito



Para que serve?

Serve para ignorar alterações em arquivos contidos no repositório local. Obs: Uma vez que um arquivo já esteja presente em um commit anterior e você queira o adicionar ao gitignore, é necessário deletar o arquivo, adicionar ao git ignore e incluir o arquivo novamente dentro da pasta do projeto

Como usar?

Para usar o gitignore é necessário apenas abrir o arquivo do gitignore e incluir os arquivos desejados dentro para que sejam ignorados.

Comandos úteis

git stash: Guardar as alterações feitas num hash local e limpar o status.

git diff: Usado para ver as alterações feitas.

git cherry-pick: Trazer alterações de commits específicos para a branch local.

git reset: Volta as alterações de staged para unstaged (entre outras coisas)

git log: Exibe os commits em ordem cronológica, indicando o nome do usuário que executou o commit, mensagem e o hash do commit.

Links úteis

Treinando git: https://learngitbranching.js.org/?locale=pt_BR

Documentação do git: <https://git-scm.com/docs>

Erros comuns: <https://www.campuscode.com.br/conteudos/erros-mais-comuns-com-git-e-como-corrigi-los>

NTT DATA

Obrigado!

NTT DATA Corporation

Lucas Leitão / Jonathas Moraes/ Ricardo Santana 2022

**FUTURE
AT HEART**