everis
an NTT DATA Company

Treinamento
Git / Git Flow

Uberlândia, Fevereiro 2019

O que é o git?

- GIT é um de controle de versão de código aberto e usado inicialmente no kernel do Linux e criado em 2005, por Linus Torvalds e mantido atualmente por Junio C. Hamano. Hoje o GIT é usado por milhões de projetos no mundo inteiro, tanto proprietários ou open-source.

- Com o GIT, diversas pessoas podem contribuir simultaneamente editando e criando novos arquivos, e ainda permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas.

Primeiros passos - Uma Breve História do Git

O que é o git?

- É sistema de controle de versão distribuído, baseado em peer to peer, já os Controles de Versões Tradicionais segue o modelo baseado em cliente-servidor.
- GIT segue a estrutura de ramificação (branch) e isso deixa o código totalmente independente. O gerenciamento de ramificação (branch) é simples.
- No GIT, todas as operações são atômicas. Evitando código instável.
- No GIT, tudo é armazenado dentro da pasta .git. Isso não é o mesmo em outros VCS como SVN e CVS onde os metadados de arquivos são armazenados em pastas ocultas .cvs, .svn, etc.

O que é o git?

- GIT usa um modelo de dados que ajuda a garantir a integridade criptográfica de qualquer coisa presente dentro de um repositório. Cada vez que um arquivo é adicionado ou um commit é feito, suas somas de verificação (checksum) são geradas. Da mesma forma, eles são recuperados através de suas somas de verificação (checksum).

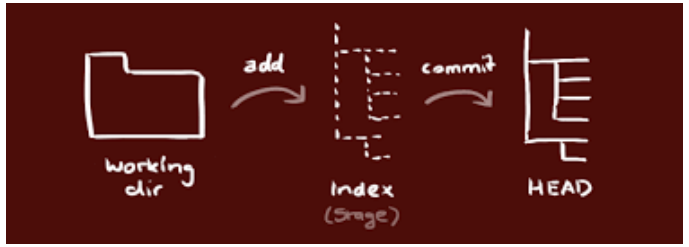
- Outra característica presente no GIT é sua área de teste ou índice. Na área de preparação, os desenvolvedores podem formatar commits e receber feedback antes de aplicá-los.

Primeiros-passos-Noções-Básicas-de-Git

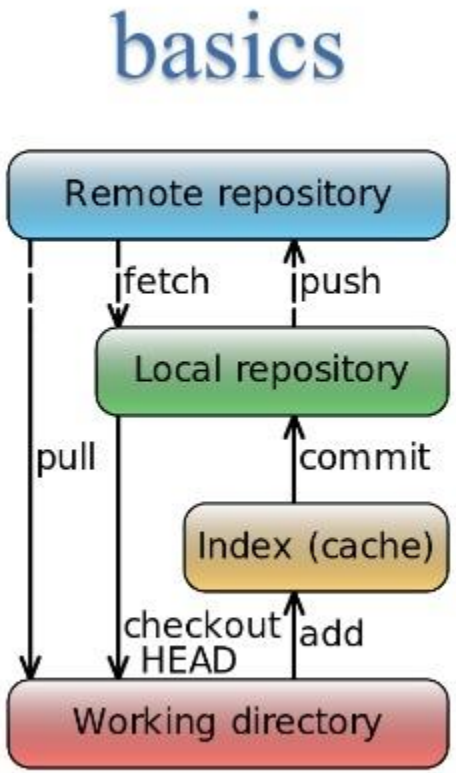
Fluxo de trabalho

Os repositórios locais do git são estruturados em 3 árvores.

- A primeira é sua Working Directory ou Repositório local, que contém os arquivos que o desenvolvedor está atuando.
- A segunda é a Index que funciona como uma área temporária, quando usa o comando add (Stage Area).
- A terceira é a HEAD que aponta para o último commit (confirmação) que você fez.



O que é o git?


basics
Remote repository
fetch
push
Local repository
pull
commit
Index (cache)
checkout HEAD
add
Working directory

O que é o git?

- git fetch baixa os HEADs com nomes ou tags de um ou mais repositórios, junto com os objetos necessários para completá-los. Basicamente ele atualiza as referências locais com relações às remotas, mas não faz o merge com o branch local.

- git pull incorpora mudanças de um repositório remoto para o branch local. É equivalente a git fetch seguido de git merge FETCH_HEAD

Fluxo de commit do projeto

```
$ echo "# curso-everis" >> README.md
$ git init
$ git add README.md
$ git commit -m "first commit"
$ git remote add origin https://github.com/rogeriofontes/curso-everis.git
$ git push -u origin master
```

Fluxo de commit do projeto existente

```
$ git remote add origin https://github.com/rogeriofontes/curso-everis.git
$ git push -u origin master
```

Problemas com CRLF

No Windows o checkout padrão é CRLF, e no Linux é LF

Se você quiser, pode desativar esse recurso em sua configuração do git core usando:

```
$ git config core.autocrlf false
$ git config core.autocrlf true
```

https://help.github.com/articles/dealing-with-line-endings/

everis
an NTT DATA Company

Problemas com CRLF

O comando deve ser dado de dentro do diretório, para que as alterações fiquem guardadas no repo. Para vê-las basta rodar um git config -e, ou exibir o arquivo config dentro de .git:

cat .git/config

Você verá o valor setado (entro outros, limitado para exibição):

```
[core]
  autocrlf = true
```

Ao fazer o commit agora, o warning não vai aparecer.

O que é o git?


```
$ git add file.txt

$ git commit -m "add file.txt"

$ git log --stat

$ git log --pretty=oneline

$ git checkout -b feature_n_v01

$ git branch -a -- lista todas as branchs.
```

```
$ git branch -d feature_n_v01

$ git show-ref --heads –s

$ git branch -v --all

$ git tag v1.1.0 -m "tags“

$ git push origin --tags

$ git fetch origin

$ git reset --hard origin/master
```

everis

an NTT DATA Company