

História do Git

- Desenvolvimento do linux
- Bitkeeper
- Necessidade de um controle de versão gratuito
- Criado por Linus Torvalds

Motivo para usar Git

- Melhora do processo
- Desenvolvimento em etapas
- Trabalho em equipe

Configuração inicial

`git config --global user.name "seu nome"`

`git config --global user.email "seuemail@emai.com"`

Iniciando um repositório local

`git init`

a pasta `.git`

Os 3 estágios

git status - verifica em qual estágios os arquivos estão

1º untracked files:

Arquivos que não estão sob o controle de versão do **git**

2º Changes to be committed:

Nesse estágio os arquivos estão pronto para entrar no controle de versão

git add .

Muda todos os arquivos para o próximo estágios

git add nome_arquivo.txt

Muda apenas o arquivo escolhido para o próximo estágios

3º Committed:

git commit

git commit -m ""

git commit -am ""

Arquivos já fazem parte do controle de versão

Estágio extra

modificando um arquivo já *commitado*, o arquivo terá o seguinte estágio :

Changes not staged for commit:

ou seja, ele está no controle de versão, porém não está pronto para ser *commitado* e aí, repetido o estágio 2

Subindo arquivos para o repositório remoto

git push origin master/main

Criando um repositório no github

repositories > New

colocar nome do repositório

Create repository

git init

git add .

git commit -m "primeiro commit"

git branch -M main

git remote add origin <https://github.com/NomeUsuario/nomeRepositorio.git>

git push origin main

A branch master/main

Adicionei um arquivo, porém não quero *commitar* ele mais

git reset HEAD nome_arquivo.txt

usando o comando acima, o arquivo selecionado voltará um estágio

Fiz algumas modificações, porém quero descartá-las

Com o arquivo no 1º estágio dar o comando :

```
git checkout nome_arquivo.txt
```

Vendo histórico de commits

```
git log
```

```
git log --pretty=oneline
```

.gitignore

nesse arquivo podemos colocar arquivos que serão ignorados pelo GIT

exemplo arquivo .vscode

Voltando versões

```
git reset HEAD~1 --soft
```

Volta a versão com os arquivos modificado, podendo fazer o *commit* novamente deles

```
git reset HEAD~1 --hard
```

Volta a versão porém, sem os arquivos dos *commits* mais recentes

*****Muito cuidado quando for voltar versões(analise a necessidade)*****

clone - faz um cópia na sua máquina de um repositório

```
git clone link_repositorio
```

fork - copia o repositório escolhido para a sua conta do github

Branchs

criar

```
git checkout -b nomeBranch
```

deletar

```
git branch -D nomeBranch
```

mudar de branch

```
git checkout nomeBranch
```

merge/pull

resolvendo conflitos