

# Git

---

É usado para fazer controle de versões de um arquivo, lembrar as versões específicas, ver quem modificou, etc.. Pode-se fazer um commit (viagem no tempo), onde cada etapa do projeto que é "commitada" para o git fica salva, sendo possível de ser recuperada; há, também, a branch, onde cria-se uma linha paralela, onde o usuário pode liberar a sua criatividade para fazer algo bem diferente no código, podendo unificar ele ao código central.

## Funcionamento básico do Git

---

### Working directory (ou working tree)

- todos os arquivos ficam alocados nesse diretório

### Stage are (ou stage tree)

- preparação dos arquivos escolhidos para serem commitados

### Repositorio local (ou histórico)

- Banco de dados dos arquivos selecionados onde ficarão os pontos da história, prontos para serem mandados para um servidor na nuvem

## Adicionar nome de usuário

---

```
git config --global user.name "Nome de usuário"
```

## Adicionar email

---

```
git config --global user.email "endereçoDeEmail@provedor.algumacoisa"
```

## Ver as informações cadastradas no git (email e senha)

---

```
git config --list
```

## Iniciar um projeto no git

---

```
git init
```

Será criado um arquivo `.git` contendo todas as informações necessárias para criar os commits, branches, logs de informações, etc.

## Conferir em qual branch e qual autor fez os commits

---

## Ter um panorama mais completo das commits

```
git log
```

## Ver todas as commits de uma forma mais simplificada

```
git log --oneline
```

## Ver as últimas $n$ alterações

```
git log -n 5 ##se quiser ver os últimos 5 commits
```

## Consultar a partir de uma data

```
git log --since=2020-09-10 ##coloque a data aqui
```

## Consultar antes de uma data

```
git log --until=2020-09-10 ##coloque a data aqui
```

## Consultar a partir de um nome de autor do commit

```
git log --author=nomeDoAutor
```

## Consultar a partir de um nome de arquivo do commit

```
git log --grep="o nome do arquivo que quer procurar"
```

---

## Marcar um ponto na história (fazer um commit)

Para salvar um arquivo local nos servidores da plataforma Git (seja ela Github, Gitlab, Bitbucket) é necessário antes definir os arquivos a serem enviados (*stage area*), e fazer um *commit* -- uma demarcação na linha do projeto.

## Preparar o(s) arquivo(s) para serem commitados (stage area)

```
git add . ##o ponto indica que está selecionando tudo dentro do diretório
```

## Fazer um commit

```
git commit -m "a mensagem que descreve o commit" ##o 'm' define que é uma de mensagem
```

## Verificar os status (quantidade de commits, qual branch está, etc.)

Deve-se executar o comando abaixo a partir da pasta onde o arquivo *.git* está gerado.

```
git status
```

## Remover algum arquivo que está na stage area (pronto para ser commitado)

```
git rm --cached nomeDoArquivo.extensao
```

---

## Comandos úteis

### Ver as alterações feitas desde o último commit

```
git diff
```

### Ver as alterações feitas dos arquivos que estão no *stage area*

```
git diff --staged
```

### Renomear um arquivo commitado

```
git mv nomeDoArquivo.txt novoNome.txt
```

### Voltar alterações antes da fase do staged

```
git restore nomeDoArquivo
```

### Voltar alterações Após a fase do staged

```
git restore --staged nomeDoArquivo
```

### Mudar a mensagem do último commit

```
git commit --amend -m "nova mensagem"
```

### Restaurar um arquivo já commitado

```
git checkout 000hashDoArquivofff -- nomeDoArquivo.md
```

## Voltar um arquivo já commitado e criar um novo ponto na história

```
git revert HEAD~5 ##voltará 5 commits antes do HEAD (o último commit feito)
```

## Alterar o nome da branch

```
git master -M nomeDaBranch
```

## Salvar no staged e commitar ao mesmo tempo (só funciona para o segundo+ commit)

```
git commit -am
```

## Subir / pegar um arquivo do Github

Para tal, antes será necessário criar um repositório local

## Subir os arquivos para a nuvem

```
git push
```

## Pegar os arquivos da branch da nuvem

```
git pull
```

## Fluxo de trabalho normal no Github

```
git init ##cria o arquivo .git
git add . ##para adicionar TUDO que está na pasta
git commit -m "first commit" ##cria a primeira linha do tempo
git branch -M main ##cria a branch main

##identifica a origem do repositório que será feito o commit:
git remote add origin git@github.com:Autor/nomeDoArquivo.git
## ou
git remote add origin https://github.com/Autor/nomeDoArquivo.git

git push -u origin main ##envia os arquivos para o servidor na nuvem
```