



# Introdução ao Git



Estes sistemas de controle possuem a função de registrar quaisquer alterações feitas em cima de um código, armazenando essas informações e permitindo que, caso seja necessário, um(a) programador(a) possa regredir a versões anteriores de uma aplicação de modo simples e rápido.

## GUI X CLI

**Git CLI:** CLI significa Command Line Interface. Você abre um terminal, digita comandos e diz ao Git o que fazer. Esta é a interface padrão e aquela que você obtém ao instalar o

**Git Git GUI:** GUI significa Graphical User Interface.

## Comandos básicos

### Windows

▼ cd

Para navegar entre pastas.

### Unix

▼ cd

Para navegar entre pastas.

#### ▼ dir

Listar todos diretórios contidos na pasta atual

#### ▼ mkdir / echo hello > hello.txt

Criar um diretório. Echo é uma função para criar um arquivo

#### ▼ del/ rmdir

Del deleta arquivos, rmdir deleta diretórios

#### ▼ ls

Listar todos diretórios contidos na pasta atual

#### ▼ mkdir

Criar um diretório

#### ▼ rm -rf

Deletar arquivos e todas pastas. Rf remove a pergunta antes de deletar

## SHA1



A sigla SHA significa Secure Hash Algorithm (Algoritmo de Hash Seguro), é um conjunto de funções hash criptográficas projetadas pela NSA (Agência de Segurança Nacional dos EUA).

```
E uma forma curta de representar um arquivo.  
1 echo "ola mundo" | openssl sha1  
2 (stdin)= f9fc856e559b950175f2b7cd7dad61face58e7b
```

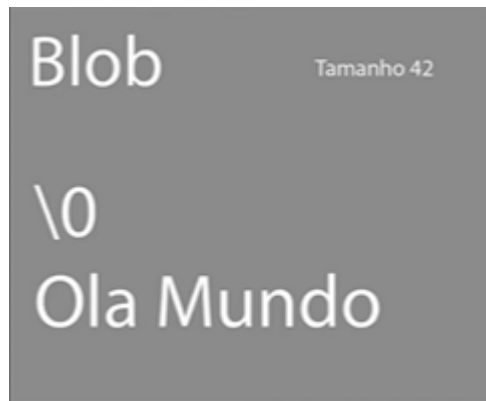
A cada cada alteração ele gera uma nova chave, mas se essa modificação for a mesma que o primeiro estado ele gera a primeira chave.

## Objetos dentro do Git

### Blob



Os blobs armazenam os arquivos como metadados, ou seja, obtendo tipo, tamanho, \0 e o conteúdo de fato.



função echo possui a string do arquivo 'conteudo' | função do próprio git

```
1 echo 'conteudo' | git hash-object --stdin
2 > fc31e91b26cf85a55e072476de7f263c89260eb1
3
4 echo -e 'conteudo' | openssl sha1
5 > 65b0d0dda479cc03cce59528e28961e498155f5c
```

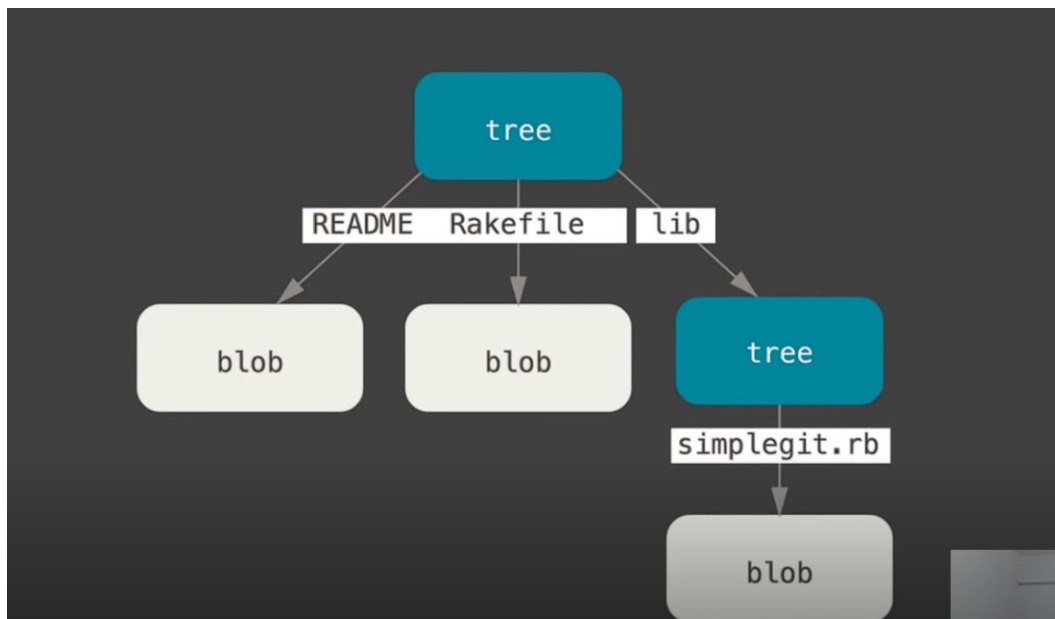
Maneira correta da função acima

```
1 echo 'conteudo' | git hash-object --stdin
2 > fc31e91b26cf85a55e072476de7f263c89260eb1
3
4 echo -e 'blob 9\0conteudo' | openssl sha1
5 > fc31e91b26cf85a55e072476de7f263c89260eb1
```

## Tree



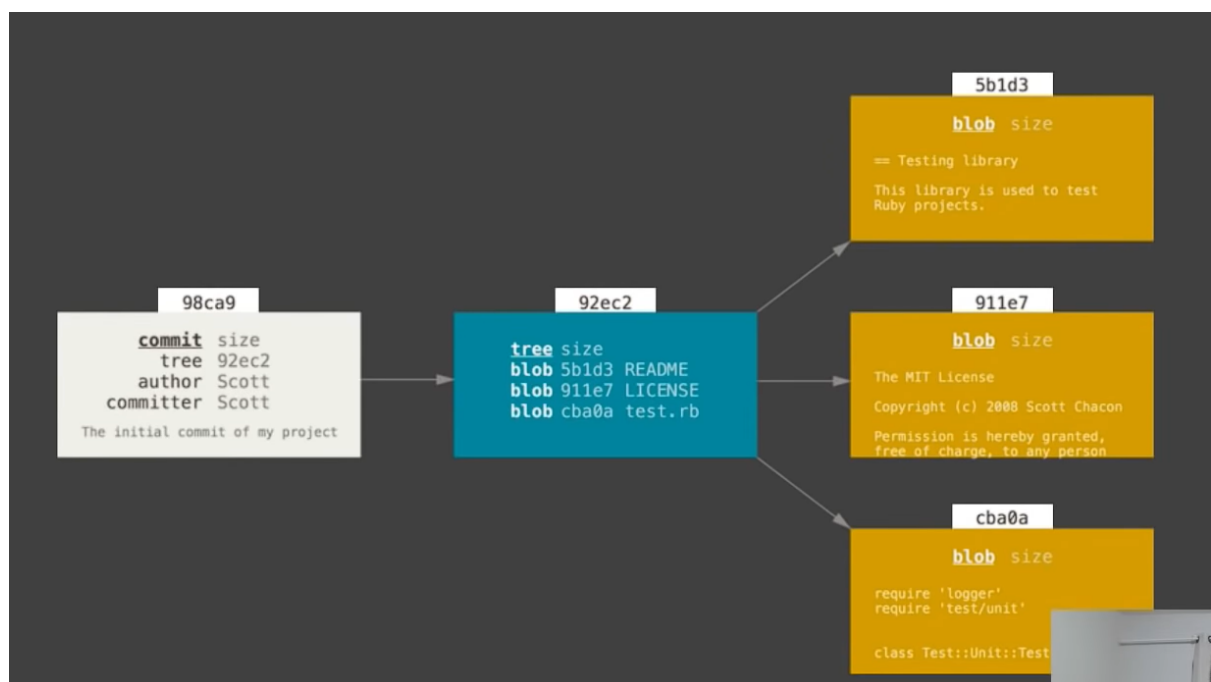
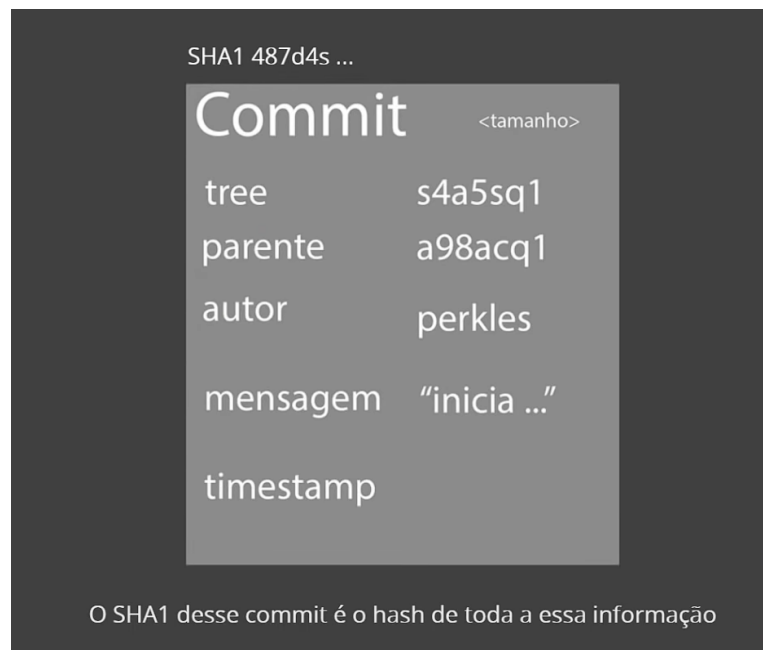
Árvores ou Trees armazenam os Blobs, é uma crescente sendo o blobs o bloco básico de composição e a Tree apontando e armazenando tipos de Blobs diferentes. Ou seja Trees realizam o encapsulamento o comportamento de diretórios



## Commit



Commit ele é o objeto que agrupa e traz sentido para as alterações em um projeto. Ele aponta para Trees > Parente (último commit) > Autor > Mensagem > TimeStamp



## Chaves SSH e Tokens



Chave SSH é uma forma de estabelecer uma conexão segura e encriptada entre duas máquinas.

## Comandos



Com o terminal antes de todo comando é colocado na frente o nome do programa

### ▼ git init

Iniciar um repositório

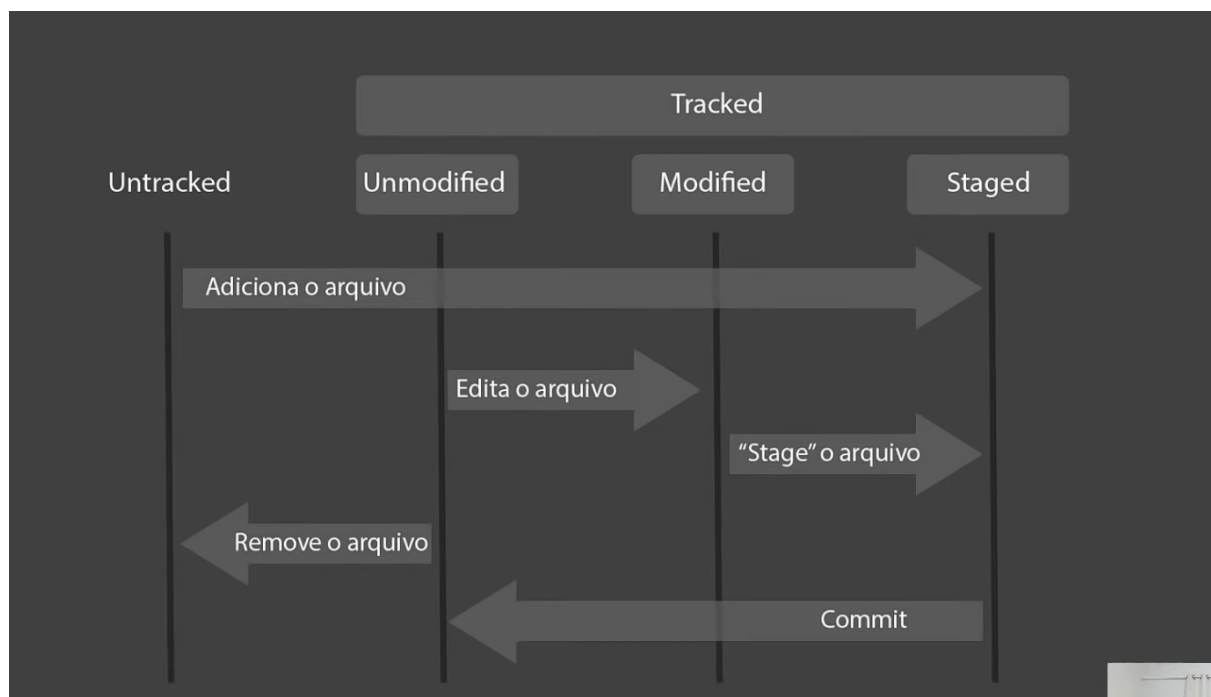
### ▼ git add

Mover arquivos

### ▼ git commit

Criar um commit

## Trajetória dentro do git

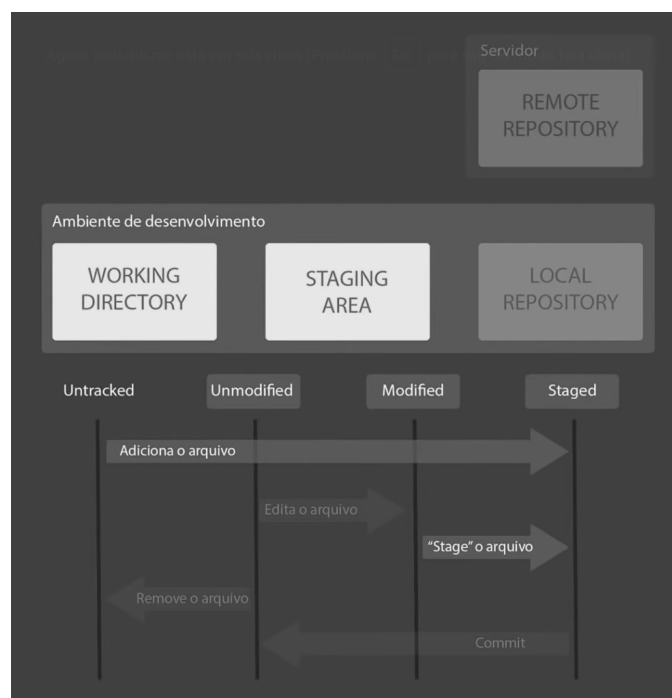


## Significado dos repositórios



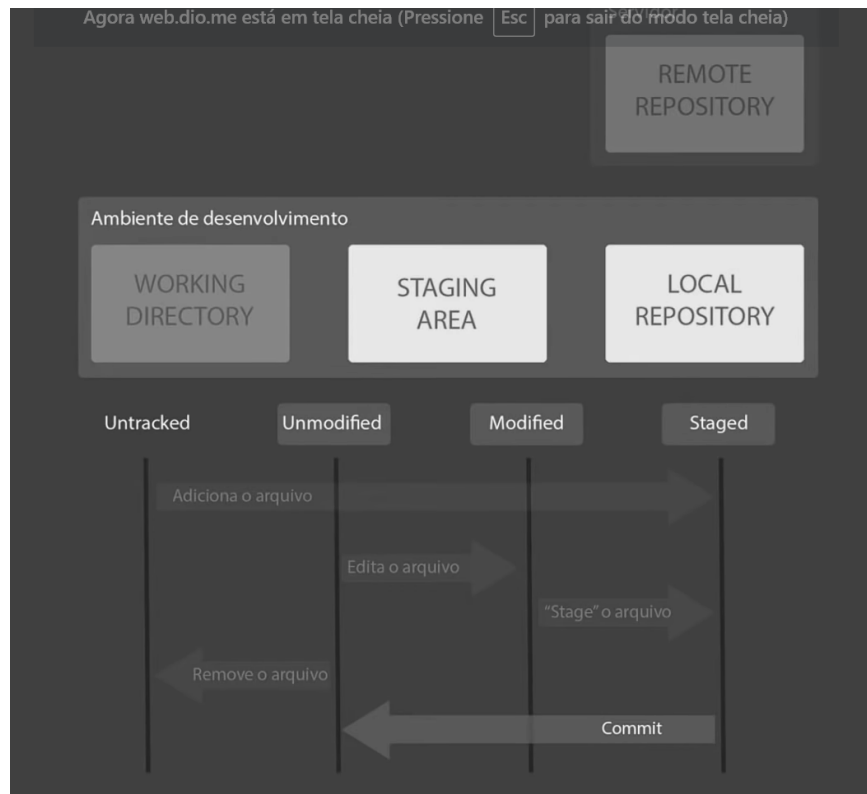
Quando é utilizado o `git add` ele vai para o estado staged

```
git add nomeArquivo
git add*
git add
```



## Quando é utilizado o commit

```
git commit -m "msg."
```



## Adicionar um repositório ao GitHub

```
git remote add origin git@github.com:Emily-Sousa/livro-de-receitas.git  
$ git remote -v
```

## Ver repositórios ocultos

```
ls -a
```

## Ver quais repositórios estão sendo apontados



```
git remote -V  
origin https://github.com/python/cpython. git (fetch)  
origin https://github.com/python/cpython.git (push)
```