



Git & Github

O que é Git

Ferramenta que permite fazer o versionamento de arquivos, ou seja, salvar o histórico de evolução na construção dos arquivos

O que é GitHub

GitHub fornece uma hospedagem para guardar os arquivos(repositório)

Comandos básicos terminal

▼ avançar pastar

Windwos → `cd`

Linux → `cd`

▼ voltar um diretório

Windwos → `cd ..`

Linux → `cd ..`

▼ listar arquivos na pasta

Windwos → `dir`

Linux → `ls` ou `ls -a` (lista repositórios ocultos)

▼ criar arquivo/pasta

Windwos → `echo "texto" > "nome_arquivo".extensão`

Linux → `echo "texto" > "nome_arquivo".extensão`

▼ deletar arquivo/pasta

Windows → del(deleta arquivos) e rmdir (deleta pastas)

Linux → rm -rf (r → recursive, apaga todas as pastas que tiver dentro|| f → não pede confirmação)

▼ limpar tela

Windows → cls

Linux → clear ou ctrl + L

▼ autocomplete

Windows → TAB

Linux → TAB

▼ ler arquivos

Windows → cat nome_arquivo

Linux → cat nome_arquivo

▼ mover arquivos

Windows → mv "nome_arquivo".extensão ./"diretorio"

Linux → mv "nome_arquivo".extensão ./"diretorio"

Como o git funciona em baixo dos panos

Secure Hash Algorithm (SHA1)

| Algoritmo de criptografia, a partir de um conjunto de funções hash

- A encriptação gera um conjunto único de caracteres identificador de 40 dígitos (Forma curta de representar o arquivo)
- openssl sha1 "nome arquivo".extensão

```
eduardo@pop-os:~/Área de Trabalho$ openssl sha1 teste.txt
SHA1(teste.txt)= ea09ef093797db5044495be55c9a85cea2d1eb41
eduardo@pop-os:~/Área de Trabalho$
```

- É através desse algoritmo de encriptação que o git diferencia as versões dos arquivos, visto que qualquer alteração a chave gerada é diferente.

Objetos Internos do git

BLOBS

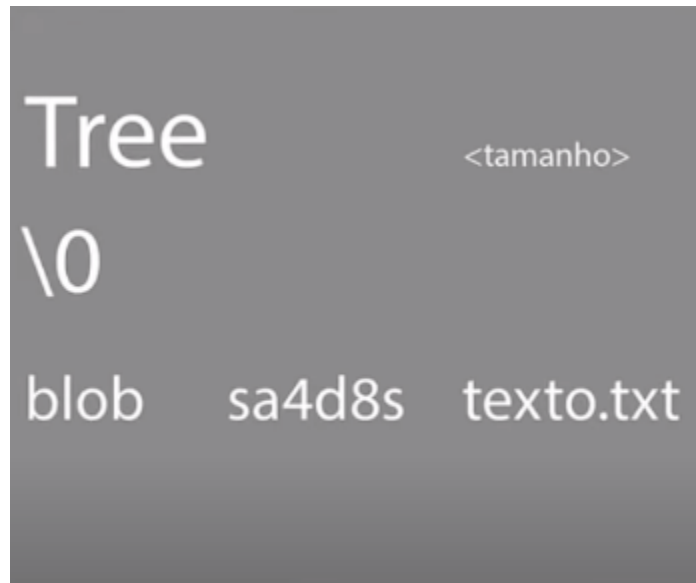
- Guarda os arquivos sha1
- Contém meta dados com o tipo, tamanho, \0 e o conteúdo

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

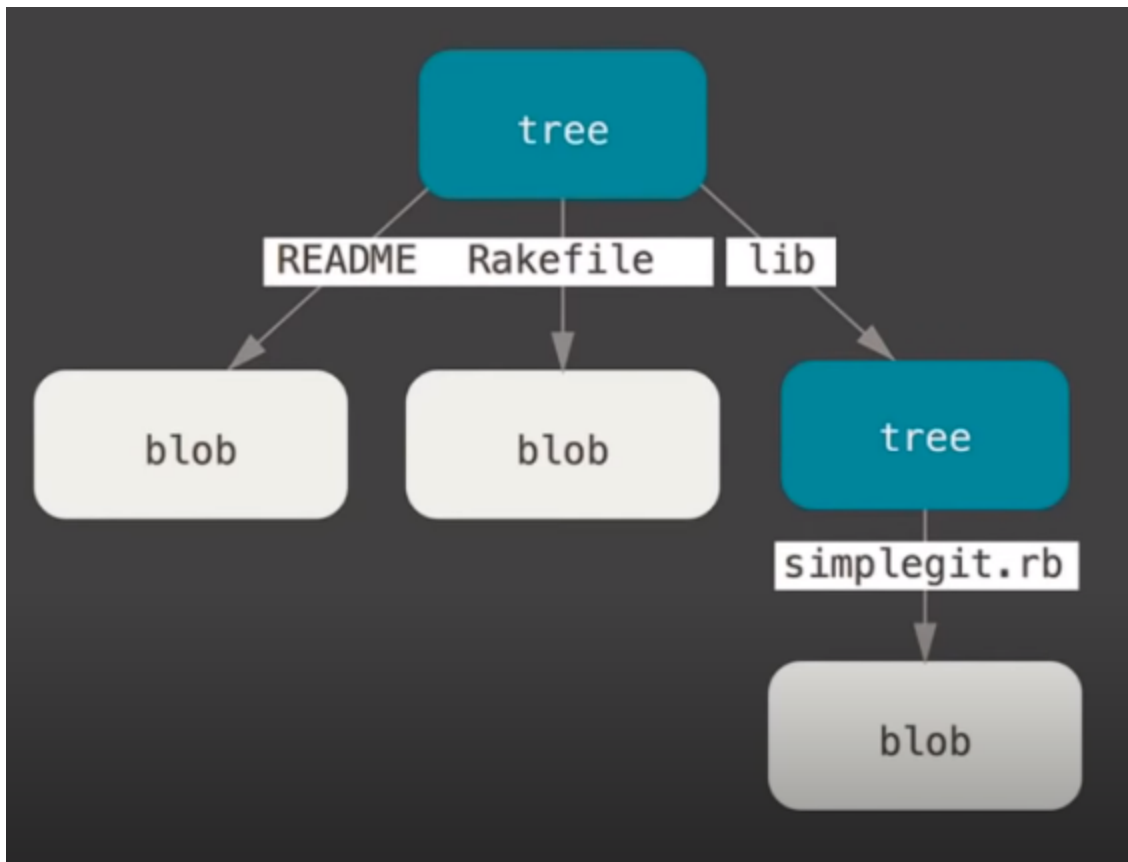


TREES

- Armazena as BLOBS
- Contém um sha1 da árvore
- Contém meta dados com o tipo, tamanho, \0, aponta para um BLOBS(que tem um sha1 do arquivo) e o nome do arquivo



- Monta as estruturas de onde estão os arquivos
- Pode apontar para BLOBS ou para outras TREES
- Qualquer alteração no arquivo o sha1 da BLOBS se modifica o conseqüentemente o sha1 da TREES se modifica também

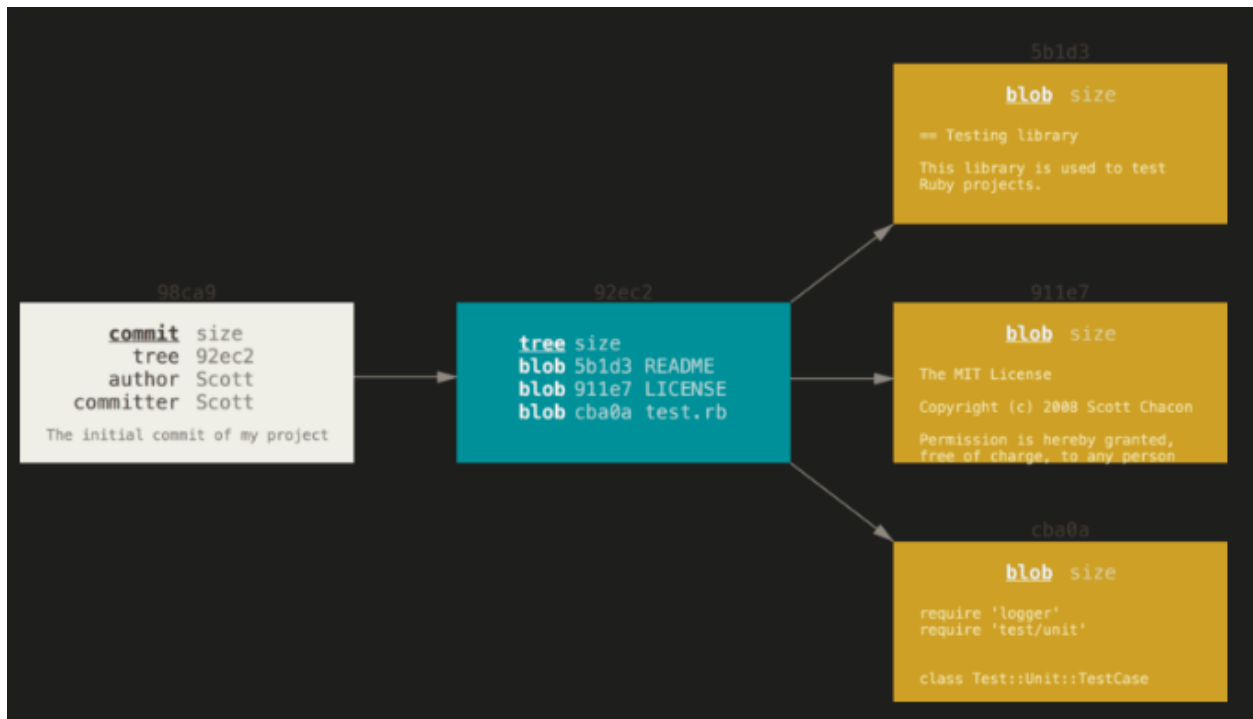


COMMITTS

- Junta tudo que foi feito
- Os commits possuem um sha1
- Aponta para uma TREES, aponta para o commit realizado antes dele, aponta para o autor e aponta para uma mensagem
- Possui um carimbo de tempo (timetamp), levando a data que o commit foi realizado



- Da o significado para os arquivos e todas as alterações
- Qualquer alteração no arquivo o sha1 da BLOBS se modifica o conseqüentemente o sha1 da TREES, que modifica também o sha1 do commit



Chaves SSH e Tokens

Chave SSH

Forma de fornecer uma conexão segura e encriptada entre duas máquinas, com ela faz o gitHub conhecer a máquina, que não pede senha.

- `ssh-keygen -t ed25519 -C email_GitHub`
- `eval $(ssh-agent -s)`
- `ssh-add ~/.ssh/id_ed25519`

Token de acesso pessoal

- Gerar o tokens no GitHub
- Na hora de clonar um repositório ele vai pedir usuário e senha, na hora de colocar a senha deve-se colocar o token gerado no github

Comandos git

- ▼ `git config --global user.name "name"`

Configura o nome do usuário global

▼ `git config --global user.email "email"`

Configura o nome do email global

▼ `git config --global --unset user.email /name "email/name1"`

modifica o nome e o email

▼ `git config --list`

lista as configurações globais

▼ `git init`

cria um repositório git

▼ `git add`

adiciona na staged os comando git

▼ `git restore --staged <file> ou`

▼ `git commit`

Dão significado as alterações

▼ `git status`

Vê os status dos arquivos no diretório

▼ `git remote add <apelido> https/ssg`

conecta a um repositório remoto

▼ `git remote -v`

Lista os repositórios conectados

▼ `git pull origin main/master ou git pull --rebase=merges`

buscar e baixar conteúdo de repositórios remotos e fazer a atualização imediata ao repositório local para que os conteúdos sejam iguais

`--rebase=merges` fala para o git criar um commit de merge para que possamos solucionar manualmente conflito

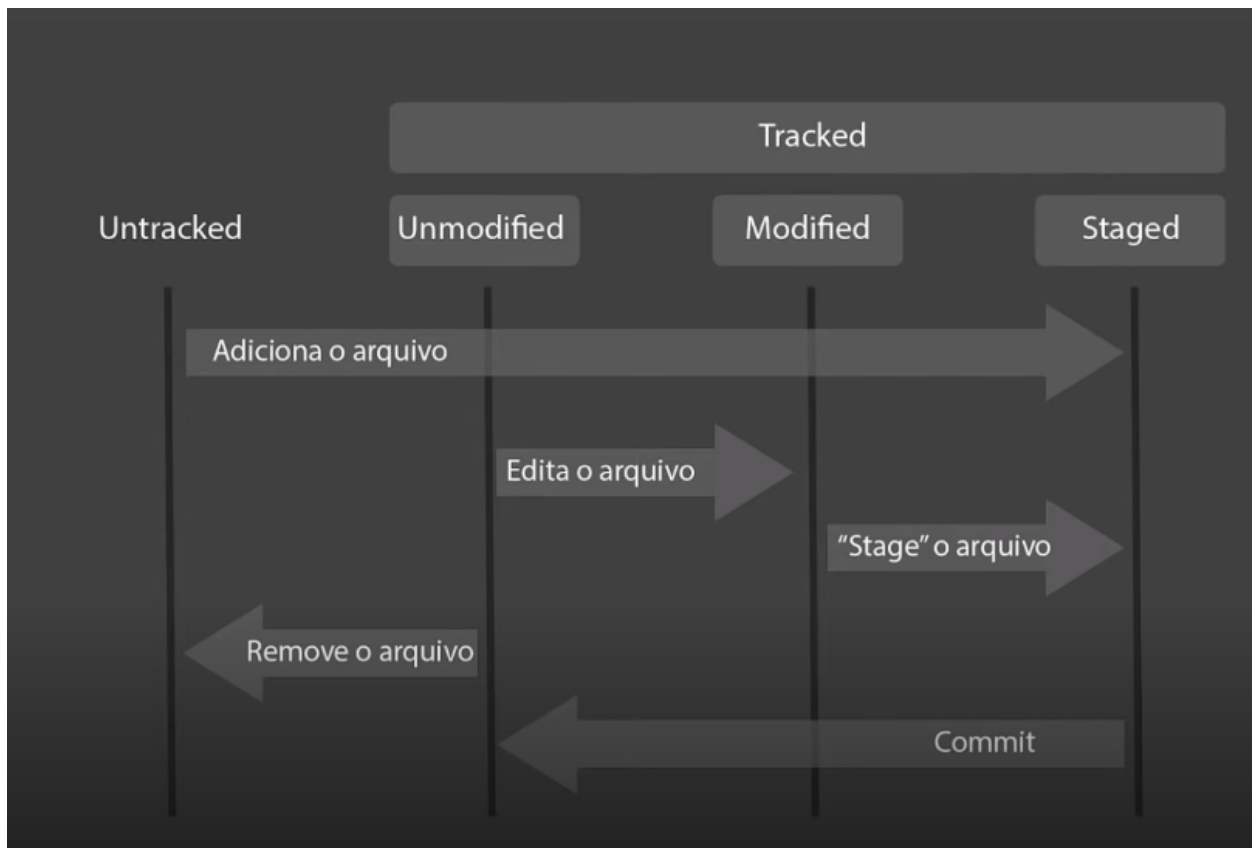
▼ `git rebase --continue`

Em encerra depois que você resolve um conflito no merge para que possa ser commitado

Tracked

Arquivos que o git tem sabem que existem

- Untracked → não sabe da existência (acabou de ser criado/excluídos)
- Unmodified → arquivo não modificado
- Modified → arquivo modificado (compara o sha1 do arquivo para vê se há alterações)
- Stage → Fica os arquivos que estão se preparando para serem comitados (git add)



Repositórios

São áreas que guardam o código que podem ser enviados para um servidor, ou local que está sendo armazenado



Resolução de conflitos

Manualmente vê qual a versão importante e faz o commit