



Comandos

GITs

Catu
2022

Comandos no git

Iniciar o git:	git init
Iniciar o versionamento	git add
Criar um commit	git commit
Criar pastas:	mkdir
Mostra arquivos ocultos	ls -a

*Quando utiliza-se o comando: init cria-se um repositório dentro da pasta.

Configuração básica do git

1º: Comando que faz com que os arquivos tenham um autor atrelado ao objeto (pode ser do repositório ou do git todo):

```
git config --global user.email "email_do_usuario.123@gmail.com "
```

2º: Definição do usuário:

```
git config --global user.name nome_usuario
```

Adicionando arquivo:

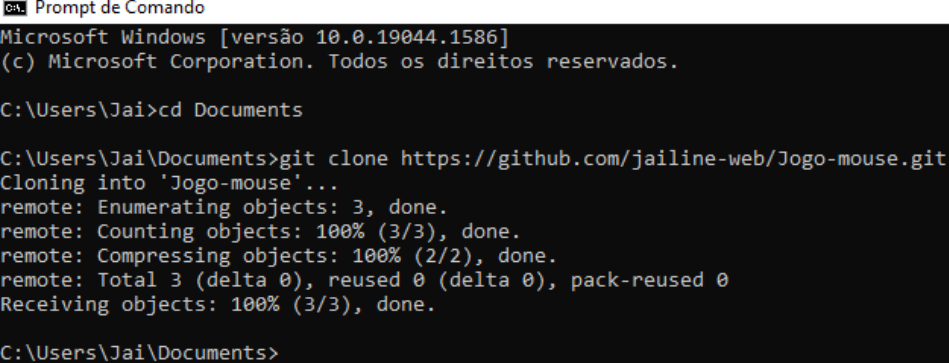
Joga todos os arquivos para a “sala de espera”(Staged)	git add * ou git .add
Mandar todos os arquivos da “sala de espera” para o repositório(github)	git commit -m "commit inicial"
[master (root-commit) 0b3bc3f] commit inicial 1 file changed, 22 insertions(+) create mode 100644 strogonoff.md sha1	← Output/Saída:

Como baixar um repositório direto no git

1º Copie a url do código no github clicando no campo “code”

2º No terminal bash insira o comando:

```
git clone url_repositório
```



```

C:\> Prompt de Comando
Microsoft Windows [versão 10.0.19044.1586]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Jai>cd Documents

C:\Users\Jai\Documents>git clone https://github.com/jailine-web/Jogo-mouse.git
Cloning into 'Jogo-mouse'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

C:\Users\Jai\Documents>
```

Com isso o repositório já estará disponível, tendo todo o versionamento de código e também pra onde o este repositório está apontando

Comando de envio do PC para o Github

Comando para enviar todos os commits para a nuvem ,empurrando para o github alterações realizadas no git local pelo protocolo HTTPS:

```
git push origin main
```

Comando para sincronizar às mudanças locais e remotas e enviar atualizar os repositórios sem erro(Puxa o que tá no repositório remoto)

```
git pull origin main
```

Objetos do git

Processos de git

* Por que do git ser um sd e seguro:

Blobs

1º Comando: envia o conteúdo e recebe o sha 1 da própria.

função: `echo 'conteudo' | git hash-object --stdin`
outra forma: `echo -e 'conteudo' | openssl sha1`

Porém cada um gera um “tipo” de sha diferente estrutura do blob, ou seja, serão feitos dois SHAs1 para um mesmo arquivo caso ele seja modificado e retornado ao seu estado inicial.

Blob tamanho: 23 \0 conteúdo	O blob insere metadados na string, com isso o mesmo sha1 é atrelado ao segundo jeito de encriptar um arquivo, retornando o mesmo sha1 para o mesmo arquivo de maneiras diferentes de encriptar.
---	---

`echo -e 'blob 9\0conteudo' | openssl sha1`: este comando gera o mesmo sha1 do comando 1.

Trees

Armazena blobs, também contém metadados e sha1, além de guardar o nome do arquivo, e o blob não guarda árvores, e elas apontam para um blob.

```
Tree <tamanho>
\0
blob sa4d68  texto.txt
```

Commits

O objeto que junta tudo, o commit aponta pra todos os metadados abaixo (a esquerda)

```
Commit <tamanho>
```

Itens do commit

tree	s3trye56
parente	76tygdu
autor	perkles

Ex: `git commit -m "commit Inicial"`
timestamp

O sha1 desse commit é o hash de toda essa informação.

Chave ssh: meio seguro de conexão do git com as máquinas dos usuários.

Processo para ativar a chave SSH

- 1º Clica na foto do perfil → configurações → chaves ssh e cpg
- 2º Abre o terminal para que a chave seja acionada/gerada pelo bash
- 3º Comando para visualizar o conteúdo da chave pública: `cat id_ed25519.pub` // apresenta a chave pública.

Tracked / Untracked

Tracked: arquivos que são rastreados pelo git(ele tem ciência da existência deles) e tem três estágios diferentes.

Modified : o arquivo sofreu alteração (comparando o sha1 para saber se foi modificado) se rodar o comando `add` de novo no modified ele vai para o stage(“sala de espera” aguardando para entrar em ação, se preparando para o commit).

Quando o processo é finalizado(commitado) ele sempre volta para unmodified até a próxima modificação e próximo commit.

Unmodified : é o arquivo que ainda não foi modificado (comparando o sha1 para saber se foi modificado)

Staged: seria os arquivos que estão se preparando para participar de outro tipo de agrupamento `add`(moveu o arquivo direto para o stage)

Untracked: o git não sabe da existência dele ao criar um determinado arquivo (ao remover um arquivo recém criado ele volta a ser um untrecked) git init (acabou de ser criado)

Demais comandos

Comando para ver o estado dos processos	git status
Comando para mover arquivos	mv strogonoff.md ./receitas/ nome do arquivo ou nome da pasta
Adicionando os arquivos ao stage	git add strogonoff.md receitas/
Comando que pega tudo que foi modificado no diretório de trabalho e envia para a área de stage para comitar	git add * ou git add .
Lista de todas as configurações realizadas no git	git config --list

*Arquivos que as pessoas vão ver: readme.md

Comandos para alterar configurações cadastradas

Altera o campo email	git config -- global --unset user.email
Altera o campo username	git config -- global --unset user.name

Enviar os arquivos para o github pela chave SSH:

Apontar o repositório remoto para a máquina local	git remote add origin https://github.com/web/livro-receitas
	git remote add livro-receitas https://github.com/web/livro-receitas https://github.com/web/livro_teste

Lista os repositórios	git remote -v
Empurra o arquivo da máquina local para o repositório remoto github	git push origin master
Puxa o que tá no repositório remoto	git pull origin master

Erro de merge, conflito entre as versões na mesma linha.

*Este erro se resolve manualmente

modifica o arquivo e commita o arquivo novamente

`git commit -m "resolve conflitos"`

por fim empurra o arquivo atual para o git com o `git push origin master`