Cls ( Crtol + L) = limpar a tela

mkdir = criar uma pasta

para voltar um nível acima = cd ..

Criar um arquivo:

Texto

Descrição gerada automaticamente

O comando “del” deleta apenas os arquivos:

Ex: del workspace

Ele só vai deletar os arquivod dentro do workspace. Não deleta a pasta toda.

Para deletar a pasta toda: C:\>rmdir workspace / s / q

Para ler o que está no arquivo, eu digito “cat” “nome do arquivo”.

Para deletar eu posso usar o “rm -r” “nome da pasta ou arquivo”.

SHA 1 = é um conjunto de funções hash criptográficas projetadas pela NSA (agência de Segurança Nacional dos EUA). A encriptação gera um conjunto de caracteres de 40 dígitos.

Três tipos básicos de objetos do GIT: Blobs, Trees e Commits.

Os blobs armazenam conteúdos. A Tree armazena blobs.

Os commits têm um timestamp.

Todos têm um sha1. Se tu mudar algo na blob, isso muda o sha1 dela, que altera o sha1 da Tree, que altera o sha1 do Commit.

Chaves SSH e Tokens.

**Para acessar a chve pública (a gente sempre usa a chave pública):**

$ cat id\_ed25519.pub

**Resultado:**

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKOYfILPMw5iFW8S/GujChl/ie/2Qfi0Nr3F7MW3mOpA kellson7777@gmail.com

No GIT o crtl+L limpa = ou o clear.

Git add nome do arquivo (move para staged só o arquivo)

Git add \* e Git add . = move tudo daquela pasta que foi alterado para o staged.

Truque: se tu clica na seta para cima, ele cola o último comando (texto)

Para “comitar”, antes de encaminhar, eu preciso primeiro deixar staged:

* git add\*
* git status (para ver se não tem mais nada para stagear)
* git commit -m “digita alguma frase de committed, ex: adicionei uma receita a mais”

Encaminhar arquivos para o Github (os arquivos em md. Já aparecem na página inicial:

* Copio o link da URL (depois que abro um repositório novo)
* Digito no git: git remote add origin ‘ link’
* Clica “ok”
* Git push origin master
* Clica “enter”

Tem que estar tudo “commitado”. Tu olha com o “git status”.

Para ver o histórico de commits a gente digita “git log”.

Se eu quiser ver, por exemplo, só os dois úmtimos commits, a gente digita “git log -n 2”.

Se eu quiser ver de forma “resumida” eu digito “git log --oneline”

Para verificar a quais repositórios eu estou vinculado no git e github eu digito: git remote -v (e “enter” depois).

Quando der erro, primeiro tu puxa e depois tu empurra de novo. Para puchar = git pull origin main

Interface gráfica do usuário

Descrição gerada automaticamente com confiança baixa

Obs: para clonar um projeto para a tua máquina, se não funcionar o “Ctrl+v” tu pode usar o “shift+insert”.

Depois que eu commitei, eu posso modificar o arquivo com o “echo” também. Se eu colocar o nome do mesmo arquivo, ele tipo muda o texto que está dentro do arquivo e no “git status” aparece “modified”:

Texto

Descrição gerada automaticamente

Para “cancelar” essa modificação “e voltar ao texto original”, a gente digita “git checkout” e “nome do arquivo”.

A gente pode usar esse “checkout” para “passear” em novas branchs:

* Verificar as branchs: git Branch
* Criar uma nova Branch: git branch “nome da Branch que eu quero criar”
* Depois que criei, se quiser ir para ela, eu digito “git checkout “nome da Branch que eu quero ir”
* Se eu quiser já “criar a branch” e já ir para ela, tudo de uma vez: git checkout -b “nome da Branch que eu quero criar e ir para ela”.
* Ela já faz com base no que há atualmente na Branch que tu está: por exemplo, se tenho um arquivo na branch-a e faço outra branch(branch-b), a branch-b já nasce com o arquivo da branch-a.
* Para deletar branchs: git branch -D “nome da branch” (é “D” maiúsculo)