Chegit coConfigurando o Git com dados do usuário :

- git config --global user.name “Fabrício”

- git config --global user.email [fabriciocmh@outlook.com](mailto:fabriciocmh@outlook.com)

- git config --global core.editor pycharm

Para ler as informações que eu cadastrei:

-git config user.name

Ou

-git config user.email

Ou para ver todas as infos :

-git config –list

Para iniciar o repositório do git, dentro da pasta onde esta o projeto digitar o seguinte comando:

-git init

Para verificar o que foi modificado :

-git status

Para adicionar os arquivos listados em vermelho ao track:

-git add nomedoarquivo

Ou

-git add -A

Quando der status novamente, os arquivo vão estar em verde..

Para o commit :

-git commit -m “comentário a respeito do arquivo”

Quando eu não dou commit, os arquivos ficam disponíveis somente no meu computador.

Para ver os commits feitos :

-git log

Para verificar qual branch estou:

-git branch

O que estiver com \* é o que estou no momento.

No próprio commit, posso fazer já o add.

-git commit -**am** “Comentario a respeito do arquivo”

Para voltar o arquivo, antes do ultimo commit.

SOFT, volta para um estado antes do commit, mantendo os arquivos alterados.

-git reset --**soft** (numero do commit que eu quero voltar)

MIXED, volta para um estado antes do commit, contudo vai ter que dar um add

HARD, vai apagar tudo o que foi feito os commits e os arquivos.

-git reset --**hard** 1b1771651b19d3e33094ef9c802d0d2a9b9bfa5a (esse é o número do commit para o qual eu quero voltar.)

Dessa forma eu apago os commits depois desse.

Trabalhando com branchs.

Para criar um novo branch

-git branch (nome da nova branch)

Para trocar de branch:

-git checkout teste

-git status (vejo o q tem para comitar)

Para verificar o que mudou no arquivo.

-git diff

Para saber quais arquivos foram modificados

-git diff --name-only

Para voltar um arquivo que não quero comitar a mudança, desfaz as alterações no arquivo.

-git checkout HEAD -- (nome do arquivo)

Para transferir informações do git no laptop(local) para o remoto.

-Criada uma chave SSH para o git Hub

Para add um repositório remoto:

-git remote add origin [https://”nome](NULL) do repositório criado dentro do git”, este é pego dentro do próprio git, após a criação do repositório.

Para ver se a conexão deu certo:

-git remote ou -git remote -v

(fetch) – a capacidade de puxar o conteúdo do repositório remoto para o local

(push) – a capacidade de levar as coisas do repositório local para o remoto.

**Para enviar do repositório local para o remoto**

- git push -u origin **master**(no caso o **master** é de onde estou enviando os arquivos.)

- - --------------------------------------

Como ignorar alguns arquivos:

Criar um arquivo com o nome

.gitignore

Nele coloco os nomes dos arquivos que quero ignorar.

Revert

Reverto as alterações do commit, eliminando a linhas de código que eu havia adicionado.

- git revert –no-edit (código do commit)

Como deletar branchs remotos:

-git push origin **:**teste

Deletar branch local

Primeiro tem q sair do branch que se deseja deletar.

-git branch -D nome do branch

Para **pegar** arquivos do repositório remoto.

-git pull origin master

Procurar fazer o PULL sempre antes do PUSH para sempre poder enviar novamente os arquivos atualizados.

Clonar projetos

-git clone **e colo o endereço https que desejo clonar.**

Isso dentro de uma pasta já criada para receber este clone.

Como contribuir com outros projetos que não são os meus.

Para isso utiliza-se o fork.

Dessa forma pego o projeto de outra pessoa e coloco no meu git.

Após faço um clone para meu repositório local.

Faço alteração subo para meu repositório e depois e clico em New pull request.

Verifica se o head sou eu e se o base é o destinatário. Após clico em create pull request