\*Tópicos Importantes

1 – Estudar mais a fundo comandos via cmd ou afins. Pelo que pude identificar, usar esses comandos é mais “boa prática” do que usar uma interface.

2 – Tentar listar abaixo os principais comandos e lógicas do GIT

Em essência o git consiste em fazer com que os arquivos que são utilizados em um programa ou projeto, tenham sua estrutura e modificações realizadas pelo time, centralizados em servidores na nuvem (tudo isso se chama Version Control).

A lógica disso é relativamente simples, os files ficam no seu PC, vão para uma área intermediária (staging área) e depois vão para nuvem.

MAPA do GIT

FILES -> STAGING AREA -> Commit History  
*Isso acontece no seu PC*

MAPA do GITHUB

Nada mais do que um lugar para guardar repositórios git na nuvem.

Os commits que ficam no commit history representam um safe point, onde cada um deles representa uma funcionalidade adicionada a sua estrutura

Comandos:

**MUNDO DA UTILIDADE (GERAL)**

#No git o uso do “.” quer dizer TUDO no diretório!  
#Se eu usar -A quer dizer que TUDO na pasta inteira!

git --version -> Verifica a versão do git , além de confirmar se foi instalado corretamente.

git --config user.name OR user.email OR color.ui -> Configura nome de usuário, e-mail e ui, para o nome colocar entre “ “.

git config -l -> Exibe as configs

git status -> verifica se os itens de um diretório estão na staging area.

**MUNDO DO DESKTOP (GIT)**

init . -> Inicia a possibilidade de utilizar comandos git para um diretório (apenas para novos projetos)

touch .extension -> Cria files dentro de uma pasta inicializada pelo git.

Add -> Leva os Files para a staging area, devo dizer o nome do file, porém se eu usar “ . “ estou mandando tudo.

rm –cached -> Remove um arquivo da staging area .

Git commit -> Leva os files na staging area para o commit history.

git commit -m “descrição” -> Faz um commit para e aloca uma descrição.

git show hash-do-commit -> Informa o que foi feito no commit, os arquivos e outro importantes.

vi <file> -> modifica um arquivo pelo cmd, achei complicado KKK, para sair aperta esq e digita :q, para salvar, :wq

cat <file> -> exibe o que está escrito no arquivo.

git diff -> diz a diferença entre o arquivo comitado antes e o arquivo alterado.

**MUNDO DA NUVEM (GITHUB)**

Push -> Leva os commits do “commit history” para os servidores na nuvem.

Pull -> Leva os files na nuvem, que foram atualizados via commits de algum usuário, para o seu desktop.

git remote add origin <https://github.com/CarlosmvBarbosa/Aprendendo_GIT.git> (Da o push para o repositório citado)

git branch -M main (Comando que me tira da Master para a Main)

git push -u origin main (Leva o que está comitado para a nuvem)

**BRANCHES (Parei nisso, continuar daqui)**

Uma branch é uma linha que guarda todos os commits e alterações de código e arquivos, uma linha do tempo. A linha central é a MAIN, e pelas boas práticas quando se adiciona algo novo, deve-se usar uma branch diferente (outra linha do tempo) e adicionar esta a MAIN (via pull request)