Aula (1)

(I) Abri o VSCode

(II) Abri a PASTA (Repositório no meu PC)

(III) Criei um Arquivo.py

(IV) No Terminal dentro do VS Code, Navegue até a PASTA que criando o repositório e dei o comando git Init.

(V) Criei Qualquer coisa dentro do arquivo e Apliquei o comando git add (nome do arquivo) [NESTE ponto o arquivo está num stage, aguardando Commit p/ ser salvo

(VI) Aplique o comando git Commit -m "qualquer coisa" (ou seja: salvei o arquivo)

Aula (2)

GitHub → Repositório Remoto. (CENTRAL)

↳ FUNCIONA COMO UMA PASTA DE ARQUIVOS

Os comandos são basicamente para enviar e buscar os arquivos no repositório central.

(IV) Criar o repositório central (My-first-Repository)

(II) clonar o repositório central

(I) clone or download

(II) copiar o url

(III) No cmder.exe navegar até a pasta em que queremos clonar o repositório e usar o comando

git clone (LINK) (NOME DA PASTA CRIADA)

(III) No VS Code, abri a pasta (NOME DA PASTA CRIADA) e criei um novo programa.

(IV) Apliquei os passos (IV) e (III) da aula 1.

=>) Voltei ao gitHub, atualizei a página e chequei se o repositório foi realmente atualizado. (remoto)

(II)

-> 2) Abri o meu Repositório Remoto e criei outro programa.

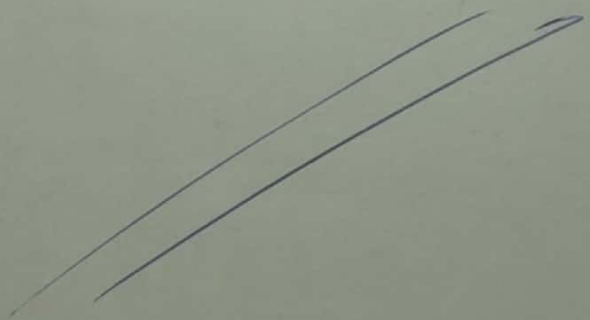
-> 3) Voltei ao meu Vscode, editei meu primeiro programa e segui os passos p/ subi-lo para meu Repositório Remoto (salvar, git ADD, git commit, git push)

Contudo, ao git push o git me alertou: "Seu Repositório local está desatualizado".

(VI) Realizei um git pull -> Atualizei meu Repositório local

(VII) Realizei um git push

(VIII) Repet o passo -> 1)



AULA (3)

QUANDO DUAS PESSOAS TRABALHAM NO MESMO FRAME
↳ mesmo que código, programa etc.

ALGUMAS CONFLITOS PODEM OCORRER, E ENTÃO TEMOS

QUE DECIDIR QUAIS LINHAS DO CÓDIGO FICAM E OUTRAS DIVERGÊNCIAS

(I) CRIEI UM NOVO FRAME -> merge.py (NO REPOSITÓRIO REMOTO)

(II) FIZ UM git pull, ATUALIZEI MEU REPOSITÓRIO local

(III) EDITEI MEU REPOSITÓRIO Remoto, E FIZ commit

(IV) EDITEI " " local

(V) FIZ git PUSH (DEU ERRO), Programa ≠ DO REPOSITÓRIO Remoto

(VI) FIZ git pull (Programa ME mostrou AS DIFERENÇAS ENTRE OS REPOSITÓRIOS)

(VII) EDITEI O QUE PRECISAVA, E SALVE (NO VS Code)

(VIII) APLIQUE git ADD., git commit, git PUSH.

AULA 48

Git Branches

→ Uma Branch é um Ramo do projeto

Utilizamos uma Branch, por exemplo, quando queremos testar alguma coisa em paralelo ao Branch principal, que é o (Branch Master.)

(I) Naveguei até meu Repositório Local (R.L)

I.I) git Init

I.II) Criei um novo Frame

I.III) git add, git commit, git Push

I.IV) Simulei um projeto, Repeti (I.III)

(II) Criei e acessei uma nova Branch através do comando

git checkout -b nome da Branch

(III) Criei um novo Frame e alterei o Frame que já existia no Branch master.

(IV) salvei o arquivo no vscode e apliquei git add. ; git commit

(V) Voltei para Branch master git checkout master

(V)

Ao retornar p/ branch master foi possível verificar que tudo aquilo que havíamos criado na nova branch, ainda não existia, é como se realmente houvessemos avançado no tempo, e depois retornado.

(VI) Adicionei a branch que havia criado a branch master.

VI.I) Verifiquei quais branches existiam com o git branch

VI.II) realizei o git merge (nome da branch)

Neste ponto foi possível perceber que tudo aquilo que havia criado na nova branch foi incluído na branch master

(VII) Excluí a branch auxiliar com o git branch -d (nome da branch)

AULA 5

NESTA AULA APRENDEREMOS A DESFAZER AS COISAS NO git. É IMPORTANTE RESSALTAR QUE NÃO É VIÁVEL APLICAR ESSE MÉTODO QUANDO ESTIVERMOS TRABALHANDO EM EQUIPE E/OU UTILIZANDO UM REPOSITÓRIO REMOTO.

OU SEJA, PODEMOS APLICÁ-LOS SOMENTE ATÉ O git PUSH

(I) CRIEI UM ARQUIVO E INICIALIZEI O REPOSITÓRIO (git INIT)

ADICIONEI O ARQUIVO NA ÁREA DE STAGE (git ADD .)

E FIZ O SEU COMMIT (git COMMIT -m " ")

(II) APLIQUEI git status P/ CHECAR SE FOI REALMENTE "COMMITADO"

" git log " → PARA SAIR: TECLAR q. " COMO SE DEU O COMMIT.

(III) FIZ UMA ALTERAÇÃO NO FRAME E SALVEI

(IV) APLIQUEI O git STATUS { AQUI FOI POSSÍVEL VER QUE A ALTERAÇÃO AINDA NÃO ESTAVA SENDO RASTREADA, PELO git

→ PARA O git RASTREAR O ARQUIVO PRECISAMOS APLICAR O git ADD. (git ADD)

(VIII)

-> ISSO SIGNIFICA QUE PODEMOS DESFAZER ESSAS ALTERAÇÕES

(V) APLIQUEI O .git CHECKOUT (NOME DO ARQUIVO) → FRASE.

FOI POSSÍVEL PERCEBER QUE A ALTERAÇÃO QUE HAVIA FEITO (MESMO SALVANDO O ARQUIVO) FOI DESFEITA.

(VI) FIZ UMA NOVA ALTERAÇÃO NO ARQUIVO E A ADICIONEI NA ÁREA DE STAGE ATRAVÉS DO git add. OBSERVE QUE ESTE CASO É DIFERENTE DO ANTERIOR, POIS NO ANTERIOR A ALTERAÇÃO FOI APENAS SALVA, NÃO FOI ADICIONADA AO STAGE.

(VII) PARA REMOVER A ALTERAÇÃO DO DO STAGE UTILIZEI O

git reset HEAD NOME DO ARQUIVO

→ A PALAVRA HEAD INDICA QUE ESTAMOS "NESSE PONTO DO commit NESTE BRANCH, PODEMOS MOVER A HEAD NO HISTÓRICO DOS Commits, COMO NESTE momento "HEAD" ESTÁ NO ÚLTIMO Commit, SE DERMOS UM RESET, TUDO QUE ESTÁ NESSE STAGE

(VIII)

(VIII) APLIQUEI UM NOVO git status, E A MENSAGEM QUE O COMANDO RETORNOU, É QUE HAVIAM ALTERAÇÕES SALVAS QUE AINDA NÃO ESTAVAM NO STAGE (COMO SE TIVESSE DESFEITO O ULTIMO git ADD)

(IX) APLIQUEI O PASSO (V).

(X) FIZ UMA NOVA ALTERAÇÃO NO ARQUIVO, SALVEI E APLIQUEI git ADD; git commit; OBSERVE DE IMEDIATO QUE ESTA SITUAÇÃO É DIFERENTE DA (VI), POIS COMMITAMOS O ARQUIVO.

(XI) AFIM DE VERIFICAR ONDE ESTAVA A MINHA HEAD APLIQUEI UM git log --decorate --ONE LINE

↪ MOSTRA TUDO EM UMA LINHA.

↳ ME DIZ EFETIVAMENTE ONDE ESTA A MINHA HEAD

(IX)

APENAS P/ UMA FINALIDADE DIDÁTICA Realizei mais um Commit, A IDEIA AGORA é colocar minha HEAD no Commit ANTERIOR AO ÚLTIMO, OU SEJA O ÚLTIMO commit SERÁ EXCLUÍDO

OBS: P/. DESFAZER um commit SERÁ NECESSÁRIO ABRIR O TERMINAL NO próprio VScode. No Cmder ou Cmd NÃO FUNCIONARÁ OS passos A Seguir.

(VII) Saí Do git log e apliquei o git reset HEAD^ O NÚMERO DE (^) INDICAM QUANTOS commits queremos desfazer NESTE CASO QUEREMOS DESFAZER um commit, então Utilizaremos Apenas UM (^).

QUANDO O commit é DESFEITO, O arquivo VOLTA P/ ÁREA DE Stage, PARA RETIRAR O commit DA ÁREA DE Stage Podemos prosseguir como no Item (VIII)

(X)

(XIII) Apliquei o passo (VI) Novamente.

-> É possível fazer com que o git deixe de rastrear um arquivo.

(XIV) Apliquei o git rm --cached (nome do arquivo)

(XV) Apliquei o git status a mensagem diz que podemos desfazer a operação (XIV), ou Realizar um commit, caso esta seja a opção seguida, o git Deixará De rastrear o arquivo passado no git CMSH.

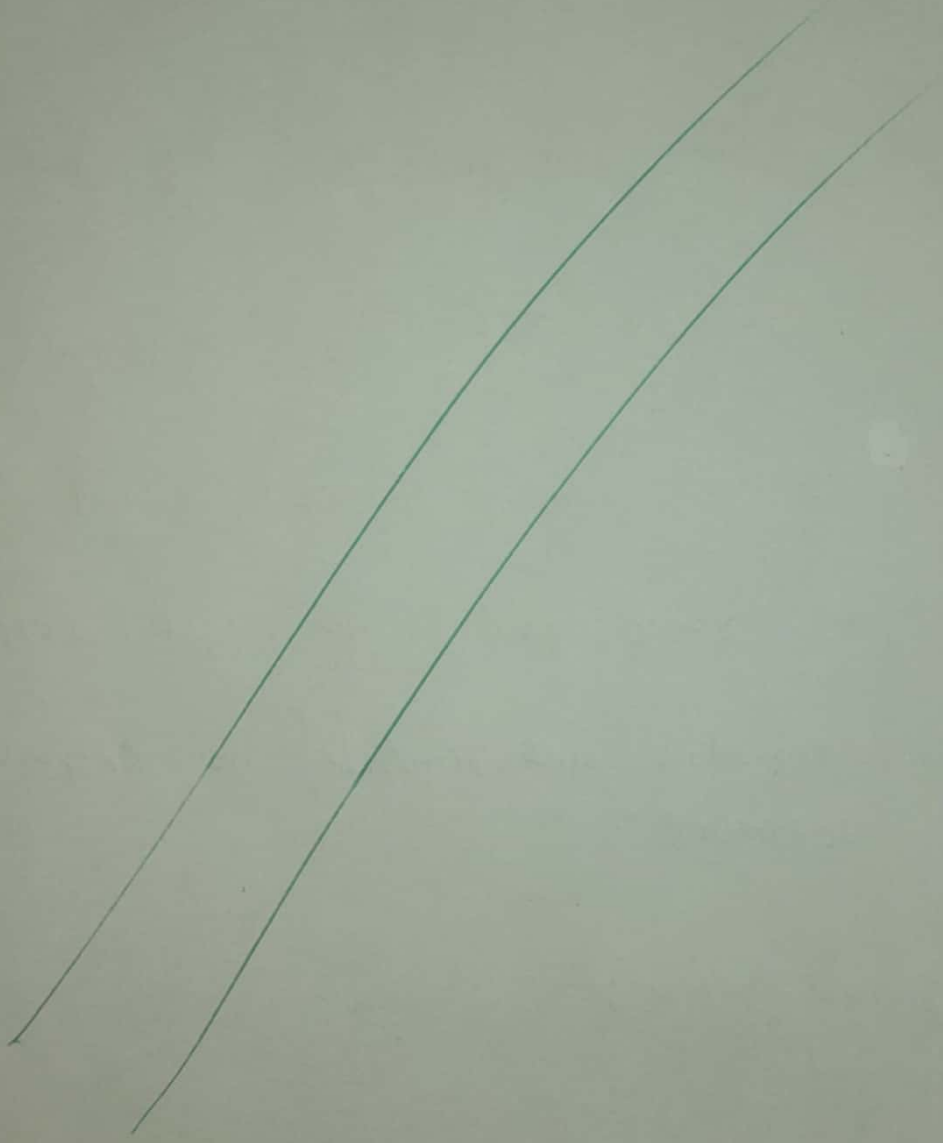
→ As operações realizadas anteriormente no arquivo ainda podem ser acessadas.

(XVII) Apaguei o arquivo (deletei)

(XVII) apliquei o git log copiei o endereço do commit no qual o arquivo ainda existia, apliquei o git checkout (Nome do arquivo) e voltei ao commit no qual o arquivo ainda existia

(XI)

XVIII ADICIONEI O ARQUIVO NOVAMENTE, COMMITEI E REALIZEI UM git PUSH.



XII