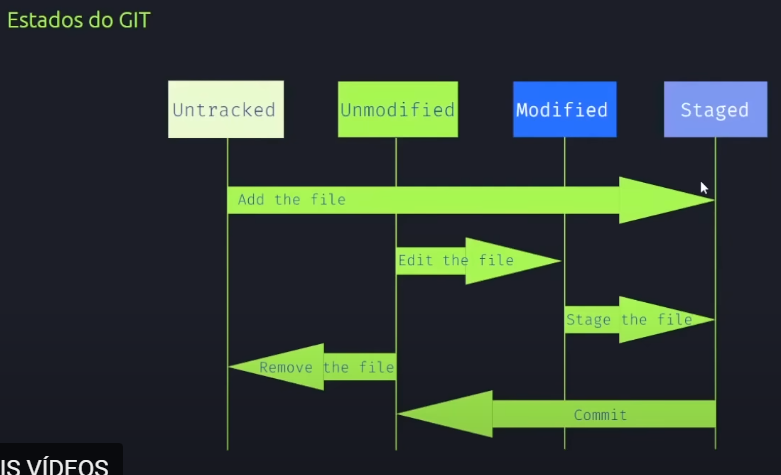
git -v #verificar se o git esta instalado e sua versao

git config --global user.name"" #configuracao de usuario

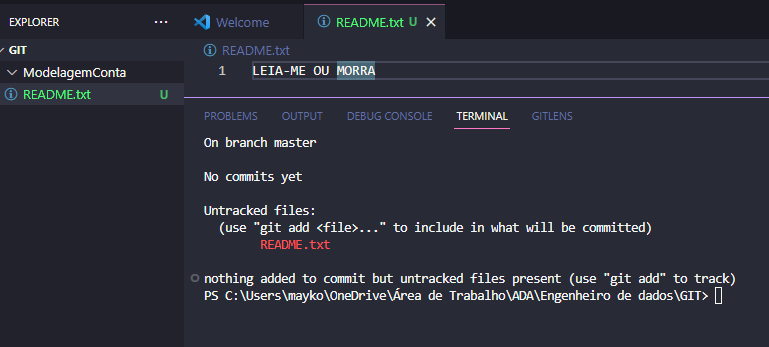
git config --global user.email"" #configuracao de email

git clone https://github.com/GuilhermeSsampaio/ModelagemConta.git #clonar e baixar um repositorio ja existente no github via HTTPS

git init #transformar a pasta / arquivo em um repositorio git dentro da pasta

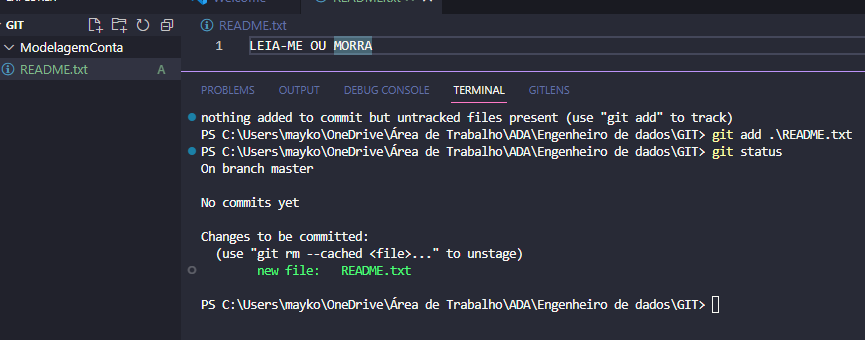


git status#informa o estado dos arquivos da pasta (se foram modificados ou nao)

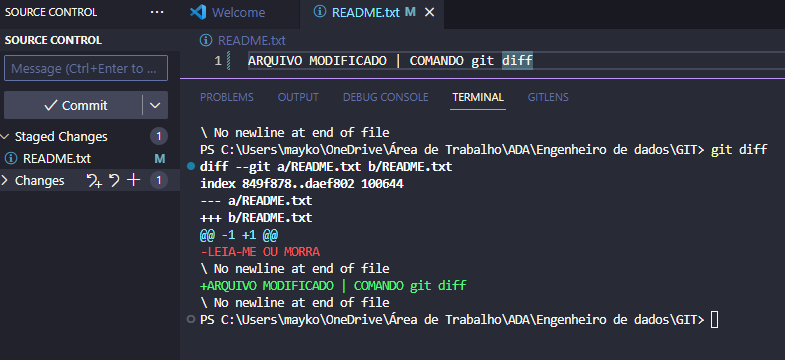


git add .\README.txt #adionando arquivo individual para comitar se for necessário (RE clique em TAB para preencher automaticamente) STAGED

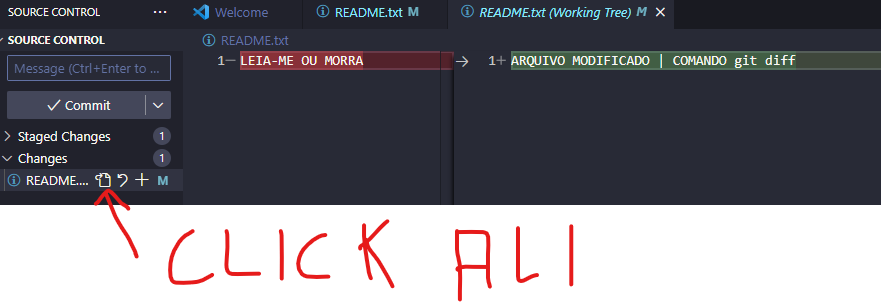
git add . #adionando todos os arquivos arquivos para comitar se for necessário STAGED



git diff #mostra as linhas em que o arquivo foi modificado depois de comitado

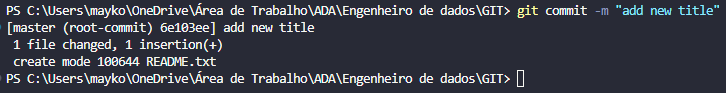


OU



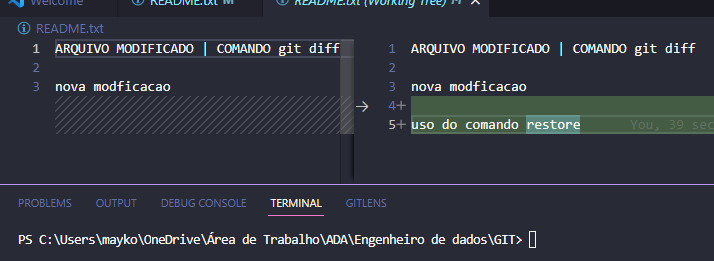
git commit -m "add new title" #comitando arquivo (dica: é legal comitar a cada alteracao do arquivo, mesmo se for pequena, como no exemplo: um novo titulo)

#COMMIT (salvar estado do projeto) SNAPSHO

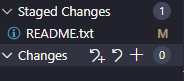


git log #mostra o histórico de todos os commits

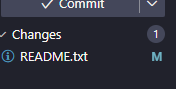




git restore .\README.txt #descarta as alteracoes feitas e volta ao arquivo cru nao modificado



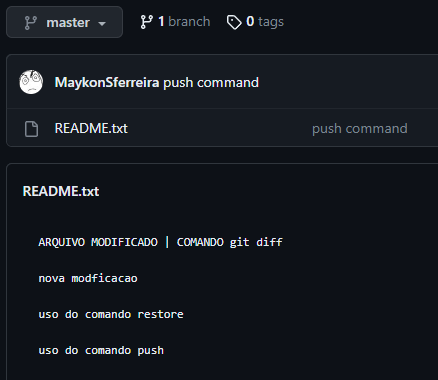
git restore --staged .\README.txt #usado quando se deseja retornar ao estado de apenas modificado (CHANGE) depois de comitado (add . STAGED)



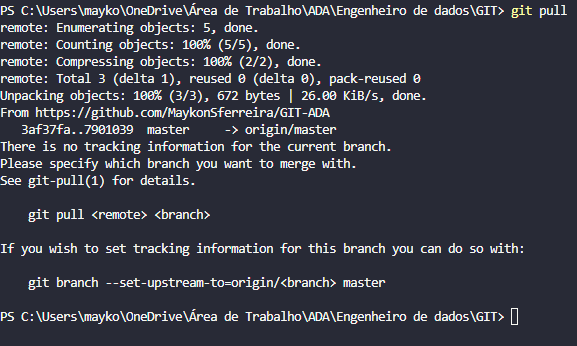
git remote add origin https://github.com/MaykonSferreira/GIT-ADA.git #Adiciona uma conecao remota ao repositorio do github

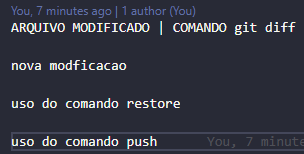
git remote #verifica o nome da conexao remota (ORIGIN)

git push origin master #empurra o arquivo local(ORIGIN) para a branch principal (MASTER) do github



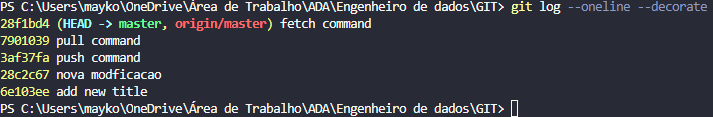
git pull origin master #puxa as alteracoes feitas na branch principal (MASTER) E ADICIONA para o meu repositorio local (ORIGIN) || Ex: QUANDO OUTRAS PESSOAS COMITAM O ARQUIVO DA BRANCH PRINCIPAL (MASTER)





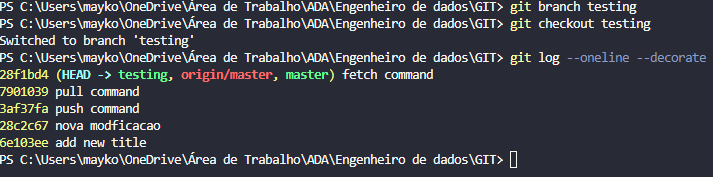
git log #mostra a localidade de onde estou nas ramificacoes HEAD

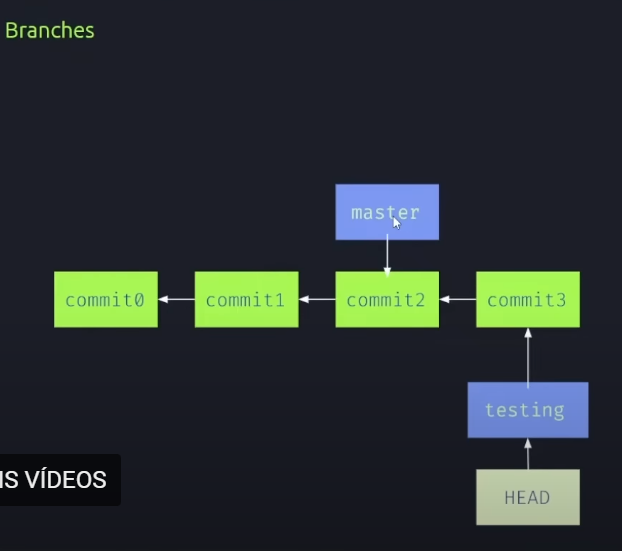
git log --oneline --decorate #mostra a localidade de onde esta nas ramificacoes HEAD em uma linha(--oneline) e com cores(--decorate)



git chekout testing #muda a localidade do HEAD para a branch testing q eu criei

git log --oneline --decorate #mostra a nova localidade de onde estou nas ramificacoes HEAD em uma linha(--oneline) e com cores(--decorate)





COMITEI UM NOVO ARQUIVO NA BRANCH TESTING

DEPOIS MUDEI PARA A BRANCH MASTER

O NOVO ARQUIVO CRIADO NA BRANCH TESTING NAO APARECE NA BRANCH MASTER PERMANECENDO NO ÚLTIMO COMMIT (PASSADO).

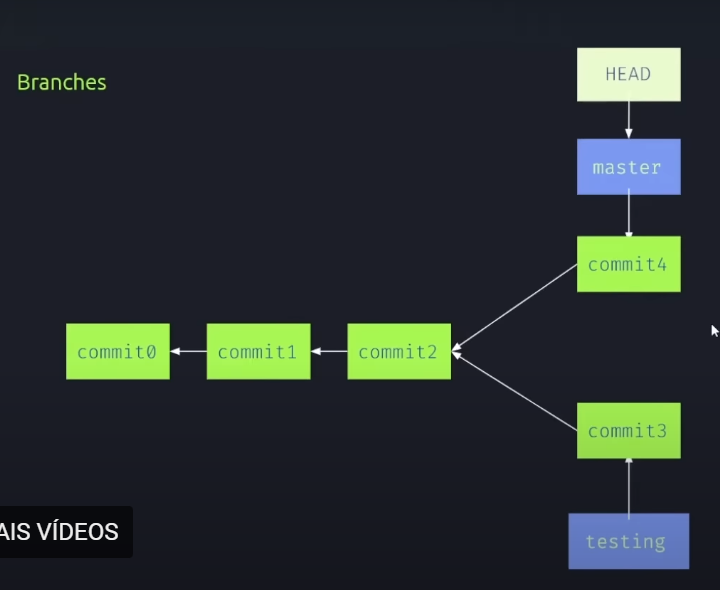
VER A FIGURA ACIMA

O QUE OCORREU FOI UMA RAMIFICAÇÃO/DIVISÃO DOS COMMITS

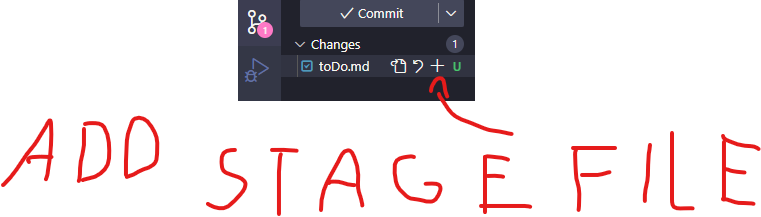
COMUMENTE USADO PARA QUE VÁRIOS DEV TRABALHEM AO MESMO TEMPO EM UM PROJETO EM DIFERENTES PARTES DELE

EM UM CERTO PONTO DO DESENVOLVIMENTO ESSAS RAMIFICAÇÕES PODERAO SE ENCONTRARAO NA LINHA PRINCIPAL DO CÓDIGO (COMMITS)

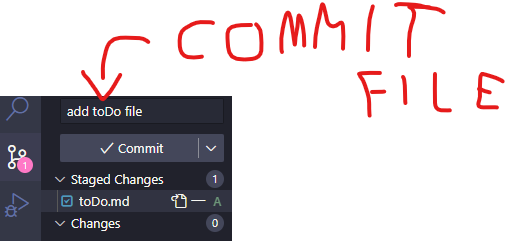
VER A FIGURA ABAIXO



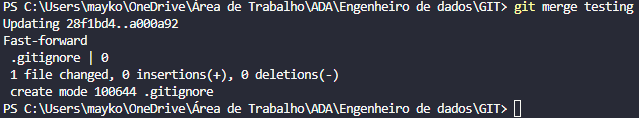
ADICIONAR ARQUIVO PARA STAGED ATRAVÉS DA EXTENSAO GIT LENS



COMMITAR ARQUIVO ATRAVÉS DA EXTENSAO GIT LENS



git merge testing #as alteracoes feitas na branch (TESTING) É FUNDIDA com o meu repositorio (MASTER)



VER A FIGURA ABAIXO

