Git-Hub 27.03

       Finalidade:Compartilhar códigos

       repositório: onde guardamos os códigos (pasta ou diretório)

       git config--global [user.name](http://user.name/) seu nome(digitar comando assim)

       git config -- global user.email seu email (digitar também)

       criação de repositório git local:

       mkdir meu\_projeto, cd meu\_projeto,git init

       entra na pasta do VS, escreve o comando teste.py

       git add

       git commit -m "Mensagem descrevendo as alterações feitas"

       ex: git commit -m “estamos adicionando a idade de fulano"

       git remote add origin URL\_do\_seu\_repositório\_no\_GitHub(depois de criar git hub online)

       no git hub entrar no seu perfil, criar um novo repositório, dê nome ao repositório (nome:git hub)

       o repositório deve ser público

       add a README file(marcar)

       add gitignore(none)

       license(MTI license)

       agora o repositório está sendo criado

       Branch(versões do repositório) permitem trabalhar em várias funcionalidades ou correções simultaneamente sem interromper o desenvolvimento principal. A branch principal é normalmente chamada de main ou master, e outras branches são criadas para diferentes tarefas (como adicionar um novo recurso ou corrigir um bug).

       Merge(mesclar)

       Agora no VS digite git remote add origin URL\_do\_seu\_repositório\_no\_GitHub (ir em code no git hub online e colocar no lugar da URL)

       git push -u origin master(para enviar mudanças para o github)

       git checkout master

       git merge nome\_do\_branch

       “U” arquivo novo criado no repositório

       Git desketop, no commit USE titulo e descrição



Passo a Passo:

       alterar: git add.

       Git commit -m “...”

       Git push (-u origin)

       Git pull

01/04/2025