

GIT / GITHUB

COMANDOS

• Git init	Inicializa um novo repositório
• Git Add .	Adiciona os arquivos no commit
• Git ignore	
• Git Status	Verifica o status atual do repositório
• Git Commit -m "mensagem do commit"	Cria um novo commit com mensagem
• Git push	Envia as atualizações para a nuvem na branch atual
• Git branch	Permite listar e ver qual branch está ativa
• Git Branch nome-da-branch	Cria uma nova branch
• Git checkout nome-da-branch	Permite mudar para uma nova branch
• Git checkout -b "nome da branch de origem" "nome da nova branch"	Permite mudar para uma nova branch e criar uma nova com base em outra
• Git merge "branch a receber merge"	Permite fazer o merge da branch atual com outra branch
• Git pull	Atualiza a branch atualmente ativa

git init inicializa um novo repositório	inicializa um novo repositório
1 git init	adiciona os arquivos atuais ao próximo commit
2 git add .	verificar o status atual dos repositório git
3 git status	cria um novo commit com uma mensagem
4 git commit -m "mensagem do commit"	envia as atualizações para a nuvem na branch atualmente ativa
5 git push	permite listar e ver qual branch está ativa atualmente
6 git branch	permite mudar para uma nova branch
7 git checkout nome-da-branch	permite mudar e criar uma nova branch com base em outra
8 git checkout -b "nome da branch de origem" "nome da nova branch"	permite fazer o merge da branch ativa atualmente com outra branch
9 git merge "branch a receber merge"	atualiza a branch atualmente ativa
10 git pull	

PASSO A PASSO

***SEMPRE PUXE 1* OQUE ESTIVER NO GITHUB PARA ANTES DAR UMA MERGE NO PC DE CASA, PODE TER UMA ATUALIZAÇÃO E VOCÊ NÃO SABE E IRA SUBIR UM ARQUIVO SEM AS ATUALIZAÇÕES.**

Para trazer arquivos para seu computador: **GIT PULL**

Para iniciar em uma pasta nova, basta clicar e selecionar "**Git Bash Here**"

- Iniciar o comando no git é "**GIT INIT**"

- Para verificar o status da pasta "**GIT STATUS**"

* SE O ARQUIVO ESTIVER EM **VERMELHO** ELE AINDA NÃO FOI "SUBIDO"

- Para adicionar arquivo basta digitar: **GIT ADD "NOME DO ARQUIVO"**

* Se quiser adicionar vários arquivos de uma pasta basta: **GIT ADD .**

- Para dar um commit de arquivos é: **GIT COMMIT -M "NOME DO COMMIT AQUI"**

- Para enviar do pc para nuvem (GitHub) tem que dar um "PUSH": **GIT PUSH**

Antes você precisa escolher o local que irá subir: **GIT REMOTE ADD ORIGIN**

Copie o link do repositório e adicione no comando

GIT REMOTE ADD ORIGIN ([HTTPS://GITHUB.COM/JOZEFERNANDO/CMD-TESTE](https://github.com/JOZEFERNANDO/CMD-TESTE))

-Para verificar o histórico de versões: **GIT REFLOG**

- Para restaurar uma versão anterior: **GIT RESET --HARD "NÚMERO DO ID"**

O Número do ID aparece no [git reflog](#)

- Alterar da Master para main: **GIT BRANCH -M MAIN**

- Verificar qual BRANCH está rodando: **GIT BRANCH**

* Criar uma nova BRANCH

`git branch "nome da branch"`

`git branch Desing`

- Alterar entre BRANCH: **GIT CHECKOUT "NOME DA BRANCH"**

SEMPRE QUE CRIAR UMA BRANCH NOVA

HAVERA UM COMANDO PARA VOCÊ SUBIR A BRANCH PARA NUVEM !

BASTA COPIAR E COLAR

Para unir códigos de diferentes BRANCH , basta dar um merge

Escolha a BRANCH principal (que irá receber o código)

`git checkout "master"`

E então merge: **GIT MERGE DESING**

(Você vai selecionar a BRANCH que ira retirar o codigo e subir para a master)

- Para criar uma BRANCH de acordo com uma já existente: **GIT CHECKOUT -B SISTEMA-DE-VISÃO MASTER**

A nova BRANCH será sistema de visão e ela terá tudo que a master tem.

- É necessário criar um arquivo que ignore outros arquivos, pode ser que tenha uma pasta com senhas e você não quer subir taul
arquivo então: **TOUCH .GITIGNORE**

Esse comando cria um arquivo na pasta e a partir dai você adiciona o que não quer subir no git

Exemplo : foto.png/

senhas.txt/

Após isso você sobe o arquivo para o github pois é ele que vai bloquear oque é ou não permitido

PASSO A PASSO

- Git pull da branch principal
- Gerar uma nova branch a partir da principal
- Trabalhar e adicionar novas funcionalidades na nova branch que criou
- Finalizar o trabalho na branch temporária
- Git checkout na branch principal
- Git pull
- Mergiar o código da branch temporária com a branch principal(Depois de testar)
- Git push da branch principal