
GIT E GITHUB - USANDO O TERMINAL CMD (PROMPT DE COMANDO) E VSCODE

Diretório Principal (pasta no computador) ::: Área Temporária (Stage) ::: Repositório (.git)

- **git --version** (identificar se há instalação do git e qual a versão instalada);

Entrar no diretório usando DOS (CMD);

- Windows + R;
- CMD (Enter);
- *cd nome do diretório (DIR permite ver as pastas);*
- *é possível entrar no diretório diretamente, bastando digitar cd e arrastar a pasta para dentro do CMD pra ele identificar o caminho;*

-
- **git init** (aqui, o git será inicializado na pasta que estiver aberta no DOS, criando um arquivo git oculto na pasta);

Abra a pasta no vscode, seja pelo menu file > open folder ou arrastando a pasta direto pro vscode; e crie um arquivo HTML, para começarmos a entender melhor os comandos do github e o repositório local;

Criar um arquivo html usando o vscode

-
- **git status** (atual estado do repositório, quais e quantas branches, quantos commits, se há algum arquivo que precisa ser enviado pro git...);
 - **git add <nomedoarquivo>** (ele irá adicionar o arquivo na área temporária, deixando-o preparado para ser enviado pro repositório);
 - **git add .** (ne caso, não envia apenas um arquivo para área temporária, e sim todos os arquivos do diretório);

Usando o git status novamente, é possível ver que o arquivo foi adicionado para área temporária, ficando apto a subir pro repositório

-
- **git config --global user.name SEU NOME** (cadastrando o autor dos commits no repositório);
 - **git config --global user.email SEU EMAIL** (cadastrando o e-mail do autor dos commits no repositório);

-
- **git commit -m "MENSAGEM QUE REPRESENTA A MUDANÇA OU ENVIO"** (aqui você irá dar uma identificação as mudanças feitas);
-

fazer algumas mudanças no arquivo html (importante lembrar que, a partir de agora é sempre bom lembrar de trackear as mudanças usando git add . e subir os commits no repositório através do git commit -m "mensagem";

- **git add .**
 - **git commit -m "MENSAGEM"**
-

Crie novas páginas para testar e fixar o conhecimento usando sempre o git status, git add . e git commit...

- **git log** (mostrar um relatório de commits e branches do repositório; caso fique preso dentro do log, basta usar o atalho Q, pra voltar pra área de digitação do prompt);
 - **git checkout SEISDIGITOSDOCOMMIT** (volta no tempo do repositório para um commit específico; é possível a troca de branch também; mas lembre-se, é necessário antes de fazer um checkout, ter realizado um commit anteriormente);
 - **git checkout NOME DA BRANCH** (ele volta pro commit mais atual);
-

Criando uma nova branch

- **git branch NOME DA BRANCH** (cria uma nova branch);
 - **git log** (mostra em qual branch estou e quais branches existem);
 - **git status** (mostra a branch que estou e se há algo novo na branch);
-

Crie novas páginas para testar commits na nova branch

- **git add .**
 - **git commit -m "MENSAGEM"**
 - **git log** (mostrará em qual branch estamos, qual commit e em qual commit ficou a branch master);
-

Dica: não mude de branch sem fazer commit antes

- **git checkout NOME DA BRANCH** (troca de branches);
-

Caso haja necessidade de alterar o nome da branch, basta usar o comando **git branch -m NOME**

- **git merge NOME DA BRANCH** (com esse comando eu vou poder mesclar a branch com a minha branch principal; mas para isso é necessário que eu esteja na branch principal);
-

Lembrando que, a cada merge feito, é criado um commit automaticamente;

GITHUB REMOTO

Repositórios Públicos e Privados

- **git remote add origin URL-DO-GIT** (levando o seu repositório local para um repositório remoto);
- **git branch -m main** (renomeando a branch);
- **git push -u origin main** (enviando os dados para o repositório remoto principal);

Lembrando que, caso não esteja logado no github no seu computador, basta logar e vincular (sincronizar) a conta;

- **git clone URL-DO-GIT** (permite que você faça uma cópia de um repositório na internet);

Lembrando que, caso não esteja logado no github no seu computador, basta logar e vincular (sincronizar) a conta;
