

**MANUAL GIT E GITHUB**

O Git é um sistema de controle de versão usado para gerenciar alterações em projetos de software. Ele permite que várias pessoas trabalhem em um projeto ao mesmo tempo, rastreando as alterações feitas em arquivos ao longo do tempo.

O GitHub é uma plataforma web que hospeda repositórios Git, facilitando a colaboração entre desenvolvedores. Ele fornece recursos adicionais, como rastreamento de problemas e gerenciamento de projetos, tornando-o um hub central para o desenvolvimento de software em equipe.

**O que é um repositório?**

É um local onde os arquivos do seu projeto e as informações são armazenados. Em relação ao desenvolvimento de software, um repositório é um diretório especial que contém todos os arquivos e histórico de alterações relacionados a um projeto específico. Ele é usado para controlar e gerenciar as versões do código-fonte, permitindo que várias pessoas trabalhem em conjunto, registrem alterações e acompanhem o progresso do projeto. Um repositório pode ser local, armazenado em um computador, ou remoto, hospedado em plataformas como o GitHub, onde várias pessoas podem acessá-lo e colaborar.

**COMANDOS GIT**

**Mkdir MeuProjeto**

Cria um novo diretório.

**Cd MeuProjeto**

Muda para o diretório especificado.

**git init**

Inicializa um novo repositório Git no diretório atual.

**touch arquivo.txt**

cria um novo arquivo vazio.

**git add** **arquivo.txt**

Adiciona as mudanças ao staging área (área de preparação) para o próximo commit.

**git commit -m "Primeiro commit - Adicionando arquivo.txt"**

git commit: Salva as mudanças no repositório local.

-m: Permite adicionar uma mensagem de commit diretamente na linha de comando

**git push -u origin master**

git push: Envia os commits locais para o repositório remoto.

-u origin master: Configura o Branch local master para acompanhar o Branch remoto origin/master.

**git clone https://github.com/seu-usuario/MeuProjeto.git**

Cria uma cópia local de um repositório remoto.

**git status**

Pergunta ao Git o que está acontecendo. Ele te diz se há algo pronto para ser salvo, algo que você esqueceu de colocar na lista, ou se tudo está bem.

**git log**

Este comando exibe o histórico de commits, do mais recente ao mais antigo. Você pode percorrer os commits pressionando a tecla "q" para sair do log.

**git pull origin nome-do-branch**Este comando busca as alterações do repositório remoto chamado origin e mescla-as no branch local especificado (nome-do-branch). Se você estiver no branch que deseja atualizar, o nome do branch remoto pode ser omitido.

git branch nome-do-novo-branch

Cria um novo branch, mas não muda para ele automaticamente. Você permanecerá no branch atual.

**git checkout nome-do-branch**

Este comando permite que você mude para um branch existente.

**git checkout branch-destino**

**git merge branch-fonte**

Este comando faz o merge das alterações do branch-fonte no branch-destino. Certifique-se de estar no branch de destino antes de realizar o merge.