Comandos Básicos de Git

1. **Instalação do Git:** Abra o terminal e execute os seguintes comandos para instalar o Git:

   ```
   sudo apt-get update
   sudo apt-get install git
   ```

   Verifique se a instalação ocorreu com sucesso usando `git --version`.

2. **Configuração do Git:** Configure seu e-mail e nome de usuário que serão associados à sua conta GIT:

   ```
   git config --global user.name "Seu Nome"
   git config --global user.email "seuemail@exemplo.com"
   ```

   Substitua `"Seu Nome"` e `"seuemail@exemplo.com"` pelos seus dados.

3. **Inicialização do Git:** Dentro da pasta do projeto, digite: `git init`. Isso irá criar toda a estrutura básica do repositório.

4. **Adicionando arquivos ao Git:** Para adicionar todos os arquivos alterados à fila de atualizações do repositório, execute o comando: `git add .`.

5. **Commit das alterações:** Para confirmar as alterações feitas, use o comando: `git commit -m "mensagem do commit"`. Substitua `"mensagem do commit"` por uma descrição breve do que foi alterado.

6. **Push das alterações:** Para enviar as alterações para o repositório remoto, use o comando: `git push origin master`. Aqui, `origin` é o nome padrão do repositório remoto e `master` é o branch principal.

7. **Pull das alterações:** Para obter as últimas alterações do repositório remoto, use o comando: `git pull origin master`.

8. **Clonando um repositório:** Para copiar um repositório remoto para sua máquina local, use o comando: `git clone url_do_repositório`.

9. **Criando e alternando branches:** Para criar um novo branch, use o comando: `git branch nome_do_branch`. Para alternar para este branch, use o comando: `git checkout nome_do_branch`.

10. **Merge de branches:** Para juntar as alterações de um branch no branch atual, use o comando: `git merge nome_do_branch`.

    Claro, aqui estão alguns comandos adicionais do Git que podem ser úteis:

11. **Verificar o status do repositório:**

    ```
    git status
    ```

    Este comando mostra o estado atual do repositório, incluindo quaisquer alterações não confirmadas

Outros Comandos que podem ser Úteis

1. **Ver o histórico de commits:**

   ```
   git log
   ```

   Este comando mostra o histórico de commits do repositório atual.

2. **Reverter para um commit anterior:**

   ```
   git checkout <hash_do_commit>
   ```

   Este comando permite que você reverta para um estado anterior do repositório, especificado pelo hash do commit.

3. **Remover arquivos do Git:**

   ```
   git rm <nome_do_arquivo>
   ```

   Este comando remove um arquivo do repositório Git.

4. **Renomear ou mover arquivos no Git:**

   ```
   git mv <nome_atual_do_arquivo> <novo_nome_do_arquivo>
   ```

   Este comando renomeia ou move um arquivo no repositório Git.

5. **Visualizar diferenças entre commits:**

   ```
   git diff <hash_do_commit1> <hash_do_commit2>
   ```

   Este comando mostra as diferenças entre dois commits.

6. **Stash de alterações:**

   ```
   git stash
   git stash apply
   ```

   O comando `git stash` permite que você salve alterações que ainda não deseja commitar. O comando `git stash apply` aplica as alterações salvas de volta ao seu diretório de trabalho.

7. **Rebase de commits:**

   ```
   git rebase <nome_do_branch>
   ```

   Este comando move ou combina uma sequência de commits para um novo base commit.

Lembre-se, a prática leva à perfeição. Então, pratique esses comandos e explore mais sobre o Git.