**Comandos básicos do Git**

**git init**

Esse comando inicia um novo repositório Git em um diretório. Aqui está como usar o **git init** de forma básica:

git init

Para criar um novo repositório enquanto especifica o nome do projeto, use o seguinte comando:

git init [nome do projeto]

**git add**

Esse comando é usado para preparar alterações em arquivos, preparando-os para o próximo commit:

git add nome\_do\_arquivo

**git commit**

Use esse comando para criar uma mensagem de commit para as alterações, tornando-as parte do histórico do seu projeto:

git commit -m "Adicionar novo recurso"

**git status**

Esse comando exibe informações importantes sobre as modificações e o status de preparação de seus arquivos.

git status

**git log**

Em sua forma básica, o **git log**permite visualizar uma lista cronológica do histórico de commits:

git log

**git diff**

Esse comando permite comparar as alterações entre o diretório de trabalho e o commit mais recente. Por exemplo, esse uso do**git diff**identifica as diferenças em um arquivo específico:

git diff arquivo1.txt

Para comparar as alterações entre dois commits, use o seguinte:

git diff commit1 commit2

**git rm**

Esse comando remove arquivos do seu diretório de trabalho e prepara a remoção para o próximo commit.

git rm arquivo1.txt

**git mv**

Use esse comando para renomear e mover arquivos em seu diretório de trabalho. Aqui está o comando do Git para renomear um arquivo:

git mv arquivo1.txt arquivo2.txt

Para mover um arquivo para um diretório diferente, digite:

git mv arquivo1.txt nova\_pasta/

**git config**

Esse comando configura vários aspectos do Git, incluindo informações e preferências do usuário. Por exemplo, digite esse comando para definir seu endereço de e-mail para os commits:

git config --global user.email [seu-email@exemplo.com](mailto:seu-email@exemplo.com)

O sinalizador **-global**aplica as configurações universalmente, afetando seu repositório local.