

Clonar um repositório do GitHub:

```
git clone endereco_do_repositorio_na_web
```

Após o clone deve-se configurar a conta do GitHub na pasta local:

```
git config --global user.email "you@example.com"
```

```
git config --global user.name "Your Name"
```

ADICIONAR & CONFIRMAR

Você pode adicionar as mudanças de um único arquivo ou de todos os arquivos usando, respectivamente:

```
git add nome_do_arquivo
```

```
git add *
```

Este é o primeiro passo no fluxo de trabalho básico do git. Para realmente confirmar estas mudanças (isto é, fazer um commit), use

```
git commit -m "comentários das alterações"
```

Agora o arquivo é enviado para o HEAD, mas ainda não para o repositório remoto.

ENVIANDO ALTERAÇÕES

Suas alterações agora estão no HEAD da sua cópia de trabalho local. Para enviar estas alterações ao seu repositório remoto, execute

```
git push origin master
```

Altere master para qualquer ramo (branch) desejado, enviando suas alterações para ele.

Branches ("ramos") são utilizados para desenvolver funcionalidades isoladas umas das outras. O branch **master** é o branch "padrão" quando você cria um repositório. Use outros branches para desenvolver e mescle-os (merge) ao branch master após a conclusão.

Para verificar branches existentes:

```
git branch
```

Para adicionar um novo branch:

```
git branch nome_do_branch
```

Para trazer um branch chamado "funcionalidade_x", por exemplo:

```
git checkout -b funcionalidade_x
```

Retornar para o master usando:

```
git checkout máster
```

Remova um branch da seguinte forma

```
git branch -d funcionalidade_x
```

Um branch não está disponível a outros a menos que você envie o branch para seu repositório remoto

```
git push origin funcionalidade_x
```

ATUALIZAR & MESCLAR

Para atualizar seu repositório local com a mais nova versão, execute

```
git pull
```

Para fazer merge de um outro branch ao seu branch ativo (ex. master), use

```
git merge branch
```

ROTULANDO

É recomendado criar rótulos para releases de software. Este é um conhecido conceito, que também existe no SVN. Você pode criar um novo rótulo chamado 1.0.0 executando o comando

```
git tag 1.0.0 1b2e1d63ff
```

O **1b2e1d63ff** representa os 10 primeiros caracteres do id de commit que você quer referenciar com seu rótulo. Você pode obter o id de commit com

```
git log
```

Você pode também usar menos caracteres do id de commit, ele somente precisa ser único.

FONTE: https://rogerdudler.github.io/git-guide/index.pt_BR.html