**Termos básicos de um repositório Git*:***

* ***master:*** *branch* principal do repositório.
* ***clone:*** cópia de um repositório git existente, para o seu ambiente local.
* ***commit****:*envio de arquivos para o repositório local.
* **fetch** ou **pull:** pega as últimas atualizações de um repositório remoto. A diferença entre fetch e pull é que pull combina as duas coisas: pega as atualizações de código do repositório remoto e efetua sua mesclagem com o repositório local.
* ***push:***nós utilizamos para enviar os nossos arquivos para um repositório remoto.
* ***remote:***são os locais remotos do seu repositório, normalmente em algum servidor central, como o [GitHub](https://github.com/).
* ***SHA:***cada commit ou *node* na árvore Git é identificado por uma chave ***SHA***única. Pode-se utilizá-las em vários comandos para manipular um *node* específico.
* ***head****:*é uma referência a um *node* para o qual nosso espaço de trabalho no repositório está referenciando.
* ***branch:***é uma *branch* no Git nada mais é do que um rótulo particular em determinado *node*.

**Comandos Básicos para o uso do Git/GitHub**

* git init

Para iniciar um novo repositório

* dir  
  mostra o que tem na pasta
* git add .  
  git commit -m “mensagem indicando a alteração efetuada”  
  git push

enviar o nosso arquivo para nossa origin master.

### Git clone é um comando para baixar o código-fonte existente de um repositório remoto (como o Github, por exemplo).

git clone <https://url-do-link>

Para criar uma nova branch local, digite:

git branch <nome-da-branch>

Este comando criará uma branch local. Para upar a nova branch para o repositório remoto, você precisa usar o seguinte comando:

git push -u <remote> <nome-da-branch>

Para ver as ramificações

git branch

ou

git branch --list

Deletar uma branch a partir do comando:

git branch -d <nome-da-branch>

O git checkout é ajudar você a mudar de uma branch para outra ou então verificar arquivos e commits:

git checkout <nome-da-ramificação>

Um comando de atalho que te permite criar e ir para um branch de uma vez só:

git checkout -b <nome-da-branch>

Para fornecer algumas informações importantes sobre a branch em que você estiver no momento, incluindo se ela está atualizada em relação à master e quais arquivos foram alterados.

git status

Uma maneira segura de desfazer os commits é usando **git revert**.

git revert 'número do hash'

esse comando vai mesclar, no seu repositório local, todas as alterações feitas.

git merge <nome-da-branch>

>cd (nome da pasta)

Entrar em uma pasta

>cd ..

Sair de uma pasta

Crtl l

Limpar a tela do terminal

Del (nome da pasta)

Limpa os arquivos que estão na pasta, mas a pasta continua lá

echo “texto” > (nome do arquivo)

Criar um arquivo

>rmdir (nome do repositório) /S /Q

Remover o repositório e tudo que tem dentro

Resumo

Cd mudar diretório

Dir listar

Mkdir criar diretório

Del/rmdir remover diretório

Se é a primeira vez que utiliza configurar:

Git config --global user.email “email a utilizar”

Git config –global user.name “nome ou apelido”