### **Como instalar e utilizar o Git**

Ok, você já entendeu o conceito de Git e GitHub. Então, vamos por a mão na massa e mostrar na prática como criar um repositório Git e fazer suas primeiras alterações.

O primeiro passo é fazer o download do Git em sua máquina e realizar a instalação da ferramenta. Ele tem uma interface gráfica, além da integração com as principais IDEs do mercado. Porém, vamos fazer tudo em interface de comando.

Após instalar o Git, abra o prompt e utilize o seguinte comando:

git --version

Caso tudo tenha ocorrido conforme o esperado, o Git responderá mostrando a versão instalada em sua máquina.

Agora vamos criar a sua assinatura Git:

git config --global user.name "Seu usuario"

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

Pronto! Sempre que você salvar alguma alteração sua assinatura será incluída.

Para utilizar o Git, precisamos de uma pasta de projeto. Então, você pode entrar em seu diretório e criar uma pasta com o nome “ProjetoTeste” e, dentro dela, um arquivo txt com o nome “teste.txt”. Escreva qualquer coisa nele.

No prompt de comando, navegue até dentro da pasta “ProjetoTeste” e dê o seguinte comando para a criação do repositório:

git init

A saída esperada é:

Initialized empty Git repository **in** C:/Users/SEU\_USUARIO/ProjetoTeste/.git/

Já temos o nosso repositório Git, mas ele ainda está vazio, pois não salvamos nenhuma cópia. Você pode verificar isso realizando o seguinte comando:

git **status**

A resposta em um repositório vazio como esse será:

On branch master

No commits yet

Untracked files:

(**use** "git add <file>..." to include **in** what will be committed)

teste.txt

nothing added to commit but untracked files present (**use** "git add" to track)

Ou seja, estamos no branch master, ramo principal. Não há alterações salvas, existe um arquivo que ainda não tem cópia e não há nada pronto para ser arquivado.

Para realizar a cópia desse novo arquivo, vamos utilizar dois comandos. O primeiro é:

git add teste.txt ou git add . Para adicionar vários arquivos

Depois verificamos com:

git **status**

Sendo que a saída esperada é:

Changes to be committed:

(**use** "git rm --cached <file>..." to unstage)

new file: teste.txt

Significa que nosso arquivo já está na staged area, pronto para a criação de uma nova versão de código. O segundo comando, para confirmar e salvar essa alteração é:

git commit teste.txt ou git commit -m “alguma coisa” para colocar uma mensagem

[master (root-commit) Qualquer coisa escrita no arquivo

1 file changed, 1 insertion(+)

create mode teste.txt

Assim, caso utilizemos o comando de Git status novamente para verificar nossos arquivos, nossa resposta será um pouco diferente, já que o repositório não é mais vazio.

Agora, nesse ponto temos que fazer o vínculo com o GitHub. Feito somente uma vez.

Já tendo uma conta criada no GitHub vamos associa usuário e email com os comandos abaixo:

git config --global user.name “usuário no github”

git config --global user.email “email no github”

Precisamos agora associar o repositório remoto.   
No github crie um novo repositório, e copie a URL. Volte ao Git CMD ou bash e digite o comando:

git remote add origin COLE AQUI A URL DO NOVO REPOSITORIO do github

Em seguida enviamos os commits dos arquivos do repositório local para o repositório no github

git push origin master

Voltamos no github, atualizamos a página e verificamos se consta nosso arquivo.

Na próxima vez não será necessário fazer o vínculo do usuário e email e nem associar os repositórios.

Basta fazer add, commit ou merge e commit e push