**Instalação e configuração do Git:**

<https://dev.to/womakerscode/instalando-configurando-e-inicializando-o-git-no-linux-2m96>

**git config –global –unset user.name/user.e-mail**

**Comandos de utilização:**

É possível criar um repositório Git em qualquer pasta do computador. Após entrar na pasta desejada, basta usar o comando: **git init**

A partir de então, quaisquer arquivos presentes na pasta poderão ser indexados ao git, usando o comando **git add nome do arquivo**. Para adicionar todos, basta usar: **git add .** (com o ponto).

Para checar quais arquivos já estão indexados ao Git, usa-se **git status**. Geralmente, os indexados aparecerão em verde, enquanto os que estiverem fora do *stage*, serão mostrados em vermelho.

É possível dar *commit* em todos os arquivos indexados. Para isso, o comando é: **git commit -m “Mensagem sobre o commit”**. A mensagem servirá para identificar o commit.

Por ocasião do *commit*, será feita uma “snapshot” dos arquivos comitados. Será registrada uma versão daqueles arquivos.

Só aparecerão no **git status** arquivos indexados antes do *commit*, arquivos novos (untracked) ou arquivos outrora comitados que sofreram alguma modificação após o *commit* (de conteúdo, caminho de armazenamento, etc), que aparecerão também como *untracked*.

**Trabalhando com branches:**

A utilização de *branches* (ramos, ramificações) ao editar o código está entre as boas práticas do desenvolvimento. Dessa forma, o código inicial não será alterado. Cada branch é uma nova versão daquele código que poderá juntar-se a ele posteriormente. Seguem alguns comandos utilizados nessa prática:

Criar um branch:

**git branch nome\_do\_branch**

Acessar determinado branch:

**git checkout nome\_do\_branch**

Criar um branch e já acessá-lo :

**git checkout -b nome\_do\_branch**

Saber qual branch está sendo utilizado:

**git branch**

Listar branches

**git branch -a**

Remover um branch

**git branch -d nome\_do\_branch**

Remover um branch de um servidor remoto:

**git push origin -d nome\_do\_branch**

**ENVIANDO REPOSITÓRIO PARA O GITHUB**

Antes de enviar o repositório para o servidor remoto (Github) é necessário escolher uma forma de se conectar a ele a partir do Git. Essa conexão pode se dar através de um token gerado pelo Github ou de uma chave SSH, sendo esta última alternativa a mais prática, pois dispensará autenticação toda vez que for dar um **push** (enviar o repositório até o servidor online). Para isso, seguir as instruções desde artigo: <https://dev.to/womakerscode/tutorial-git-acesso-ao-github-com-ssh-4kg9>

É essencial também ter um e-mail e um username, que aparecerão na assinatura de cada *commit*. O username tb aparecerá no repositório do Github. Para isso, basta adicionar o mesmo username e e-mail utilizados no Github ao git, através dos comandos:

**git config --global user.name Seu username**

**git config –global email** [**seuemail@aqui.com**](mailto:seuemail@aqui.com)

Também é essencial apontar o Git para o repositório online do Github. Lembrando que, como a conexão se dará via chave SSH, ao copiar o endereço do repositório online, escolher a opção de endereço SSH, que iniciará com “git@github.com”.

**git remote add origin** [**git@github.com**](mailto:git@github.com)**:JD-Rodrigues/teste-push.git**