**Introdução ao GIT e GitHub**

**O que é GitHub?**

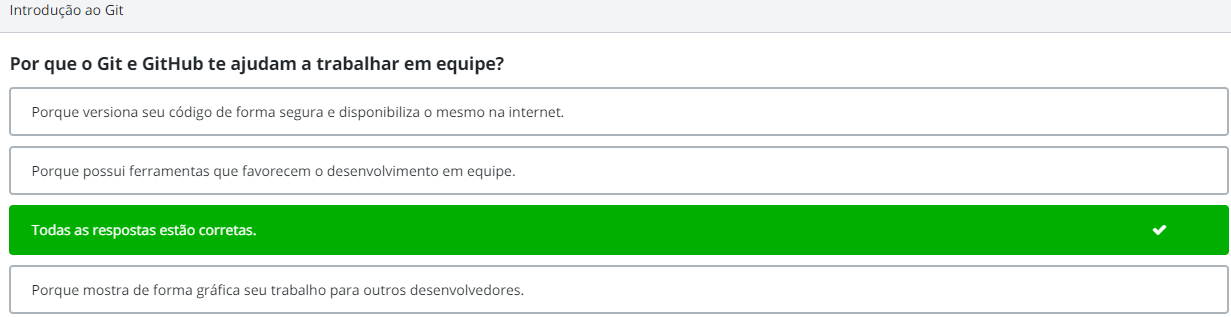
**Um servidor remoto para armazenamento de código.**

**O que o GIT é?**

**Um sistema de versionamento de código distribuído.**

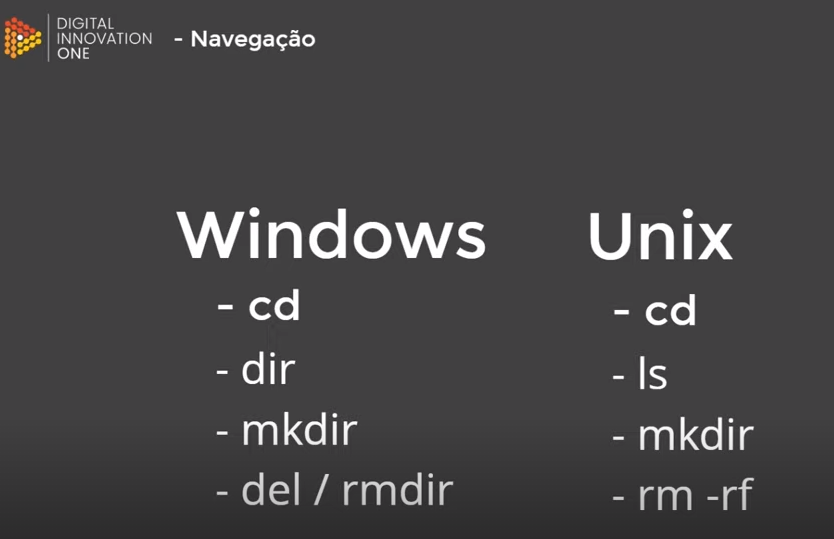
**Quem foi o criador do Git?**

**O mesmo criador do Linux, o Linus torvalds.**



**Navegação básica no terminal & instalação;**

**GIT –** È um CLI – comand line interface (Não tem interface gráfica, tudo por linha de comando).



Windows / Linux

Dir- utilizado para listar os diretórios no Windows. (Linux: ls)

Cd – utilizado para navegar nas pastas. (Linux:cd)

cd.. – retrocede para a pasta anterior que estava. (Linux: cd ..)

cls – utilizado para limpar o terminal. (Linux: clear ou ctrl+L)

OBS: Utilizando o botão “tab” no teclado ele da autocomplit.

Mkdir – utilizado para criar pastas. (Linux:mkdir)

OBS:‘>’ redirecionado de fluxo.

del – utilizado apenas para deletar arquivos. (Com o comando abaixo deleta todos os arquivos dentro de um direotiro)

EX: **del nomedodiretorio**

**Criando arquivo pelo cmd do Windows.**

**Echo –** Comando utilizado para

**EX: echo hello > hello.txt**

**Apagando diretórios junto com seus arquivos**

**Windows(CMD):**

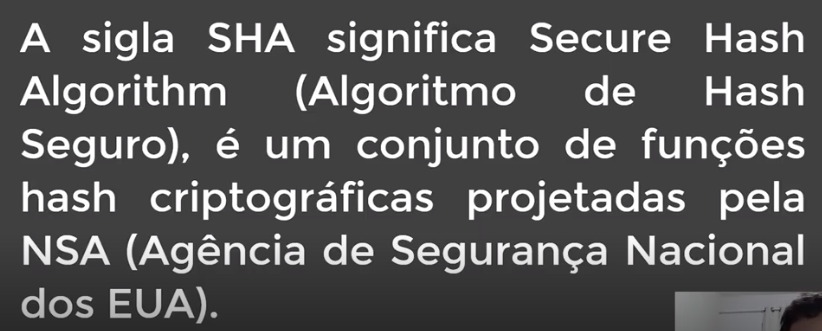
**Rmdir nomedodiretorio /S /Q**

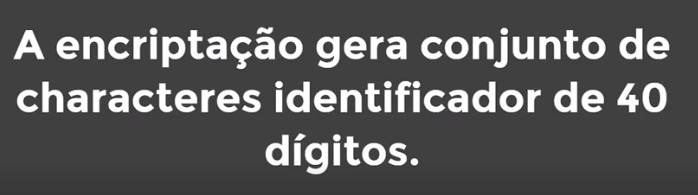
**Linux (Terminal): rm -rf nomedodiretorio/**

**Entendendo como o GIT funciona por baixo dos panos**

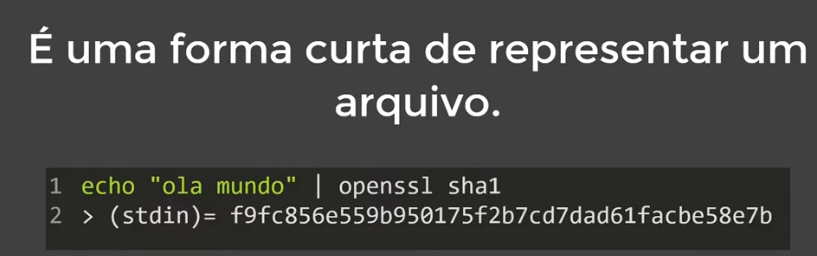
**GIT Bash – Você pode utilizá-lo para usar comando que vão servir tanto para o Linux quando para o Windows.**

**- SHA1**





**Exemplo:**



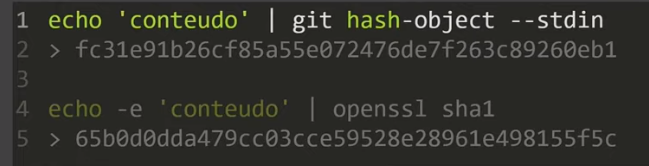
**Exemplo 2: openssl sha1 arquivo.txt**

**OBS: Com esse comando podemos criptografar arquivos e objetos.**

**Objetos esses que são:**

**BLOBS:**

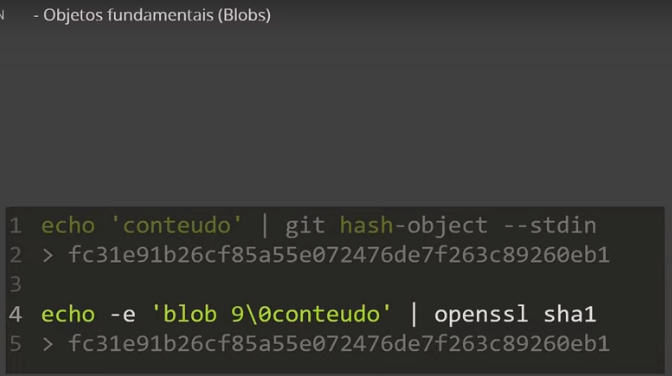
**Ex:** A função “echo” para poder imprimir uma string junto do ‘|’ para executar mais de um comando junto e passar essa string para uma função do GIT que é a “hash-object --stdin” o ‘--stdin’ é para passar para a função que o que vai passar por ela é uma string.



Esse arquivos ficam guardando no neste objeto chamado Blob

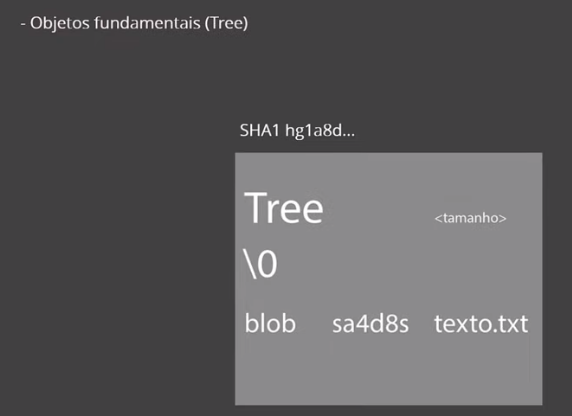


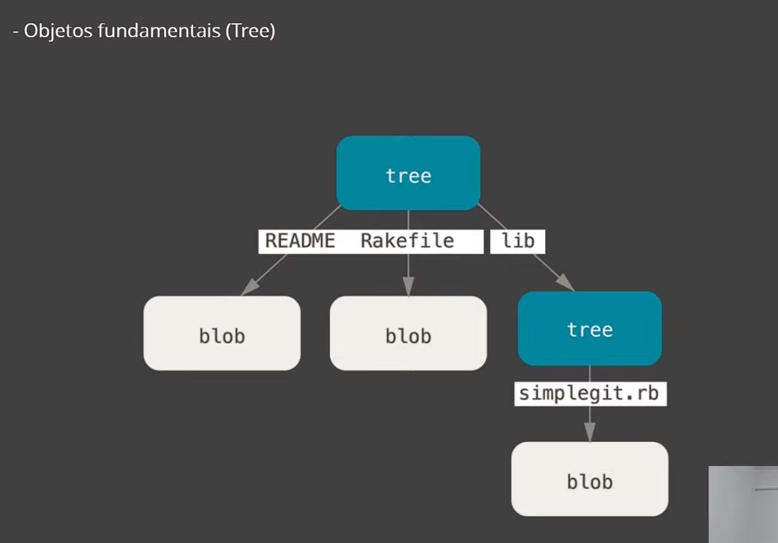
Usando o “openssl sha1” passando os metadados para um string



**TREES (Arvores): É responsável em montar as estruturas de onde está os arquivos, ele guarda o char do arquivo.**

**Elas podem apontar para o blob ou outras arvores de forma recursiva.**

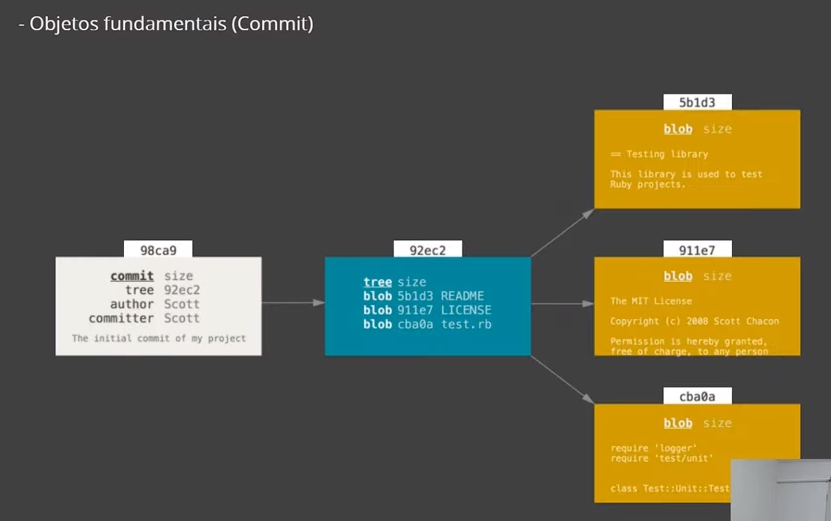
**As arvores tem um SHA1 do arquivo.**



**COMMITS: Junta todo os arquivos.**

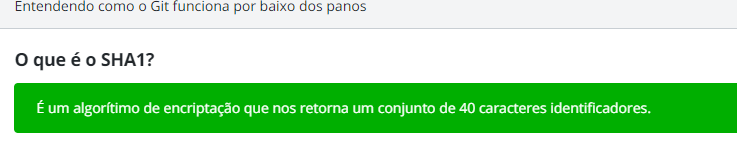
**O commit é único para cada autor**

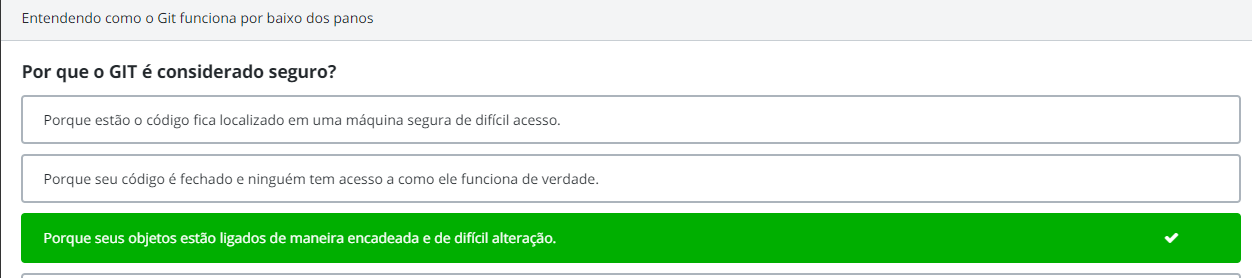




**O QUE É UM SISTEMA DITRIBUÍDO?**

**Um sistema que possui múltiplas cópias de si mesmo em diferentes locais.**





**Primeiros comandos com GIT**

**-git init;** Utilizado para iniciar o git e possibilitar que o git possa versionar e gerenciar o código.

Para visualizar o arquivo “.git” utilize o comando **ls -a**

**Configurações iniciais:**

1º - primeiro um e-mail com o comando: **git config --global user.email “**[**djovanibloomfield@yahoo.com.br**](mailto:djovanibloomfield@yahoo.com.br)**”**

2º - criar um usuário com o comando: **git config --global user.name dbzardo**

OBS: caso precise alterar alguma configuração utilize o comando:  **git config --global --unset user.name (Com esse comando você remove a configuração acima e configura novamente)**

**Adicionando um arquivo:**



**Iremos criar um arquivo tipo Markdown “.md”**

**Markdown: Uma forma mais simples de utilizar html**

**Ex: no editor typora para por o titulo igual a imagem acima utilize a “#”.**

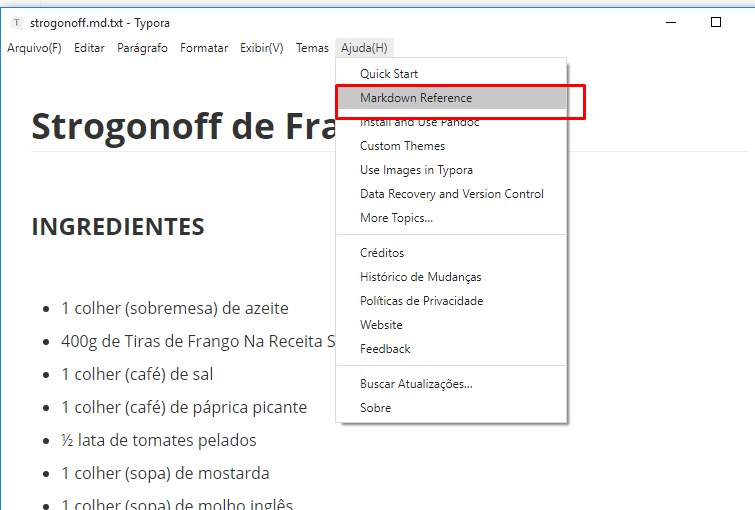
**# Strogonoff de Frango**

**OBS: Para por algum emociton do próprio editor basta utilizar o “:”**

**EX: # Strogonoff de Frango:chiken**

**EX:** Colocando uma palavra ou frase em negrito utilize entre a frase ou a palavra “\*\* escreva aqui\*\*” para por em itálico utiliza o \_escrevaaqui\_

Para verificar outros comandos no typora basta seguir os passos da imagem abaixo.



**git add \***

Exemplo de um comiit : **git commit -m "comit inicial"**

**Resultado de saída**

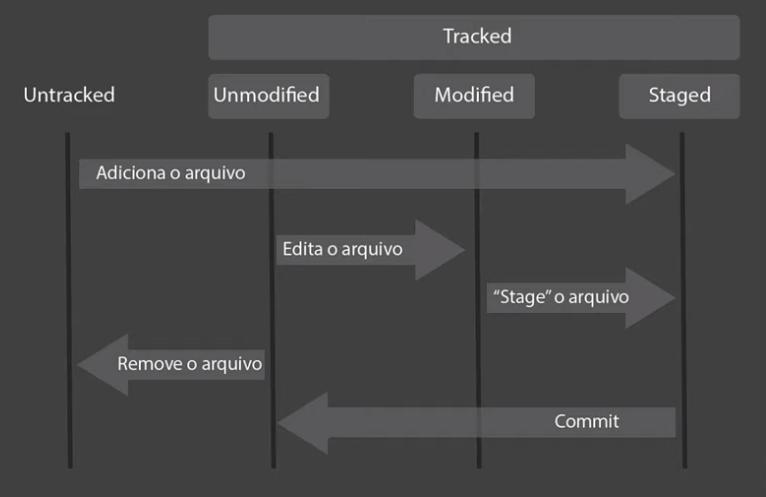
**$ [master (root-commit) 8381bef] comit inicial**

**$ 1 file changed, 34 insertions(+)**

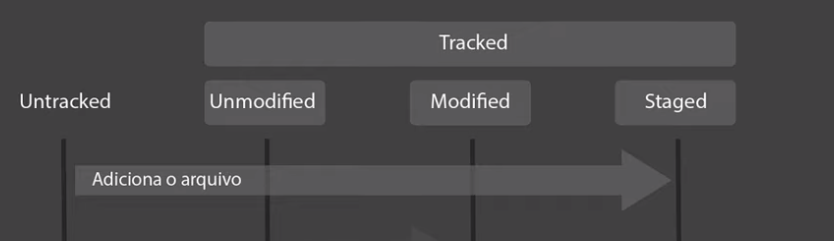
**$ create mode 100644 strogonoff.md.txt**

**Ciclo de vida dos arquivos**

**GIT INIT -**  Ele inicializa o repositório no GIT dentro de uma pasta.



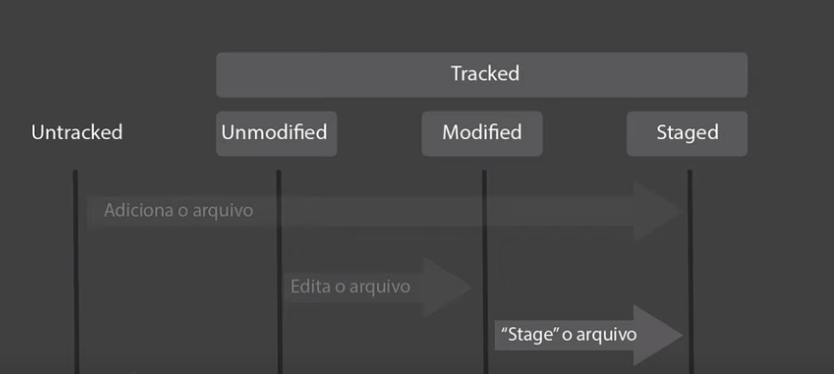
Quando usamos o comando **GIT ADD \***



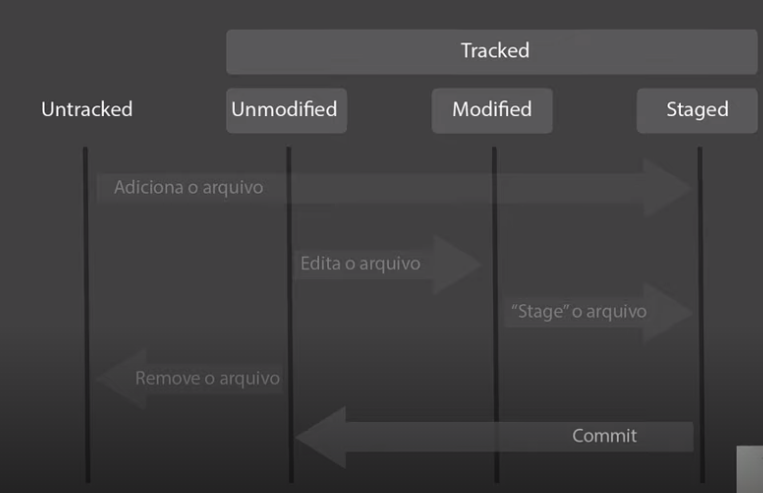
Quando modificamos e vai para o Unmodified.



Ai utilizamos novamente o GIT ADD \* para encaminhar ele para o Staged, onde ele está se preparando para darmos um GIT COMMIT



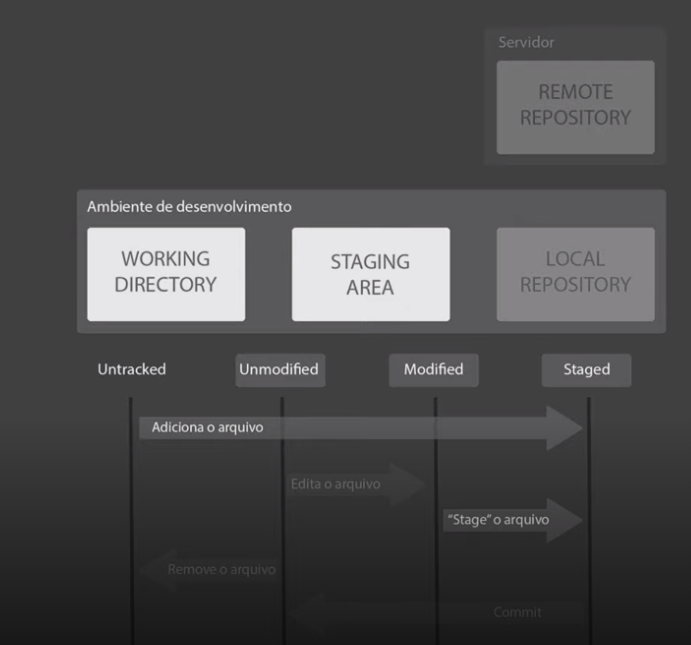
Após isso ele encaminha novamente para o Unmodified



**Repositórios**



**Quando utilizamos o GIT ADD**



**Comando para verificar o status do repositório.**

**EX: git status**

**EX: “mv nomedoarquivo.md ou.txt ./nomedodireotiriodestino”** comando utilizado para mover arquivos de uma pasta para outras.

**Após mover é necessário que usemos o GIT ADD nomedoarquivo.md ou .txt nomedapasta/**

**EX: git add strogonoff.md.txt receitas/**

Após isso se usar o comando status veremos as alterações feitas, após isso basta efetuar o **commit -m “ Descrição da mensagem que iremos passar ao dar commit“**

EX: **git commit -m "criar pasta receitas, move arqivo para receitas"**

**git status**

**On branch master**

**Changes to be committed:**

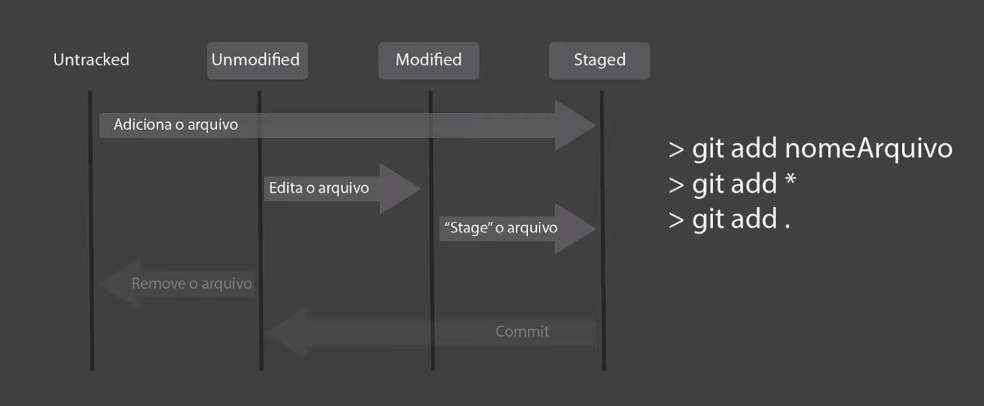
**(use "git restore --staged <file>..." to unstage)**

**renamed: strogonoff.md.txt -> receitas/strogonoff.md.txt**

Criando arquivo: **echo > README**

Agora irei incluir o arquivo com o comando **git add \*.**

O comando **git add \*** : adiciona todos os diretórios que estiverem com o status **Untracked.**

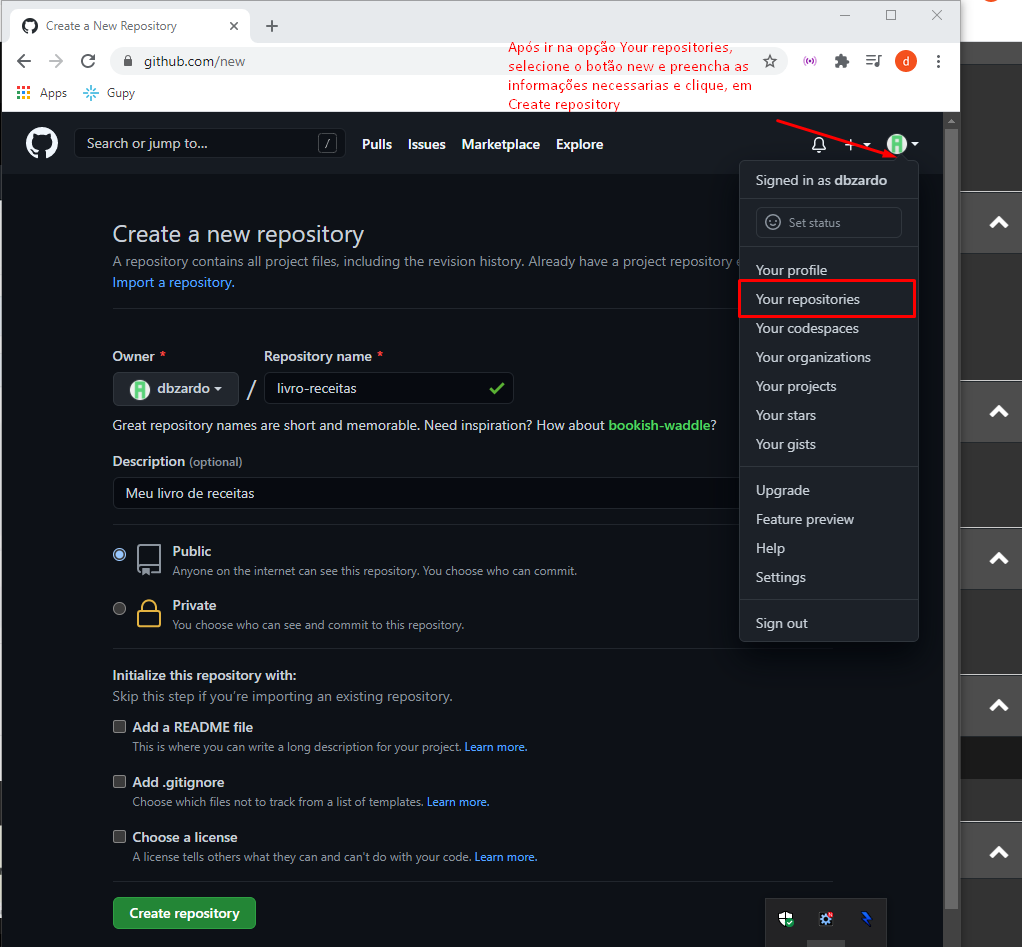


O git bash utilize o comando **git config – list**

Com esse comando você verifica as configurações feitas no Git Bash e lá ajusta o e-mail e usuário ao do GITHUB

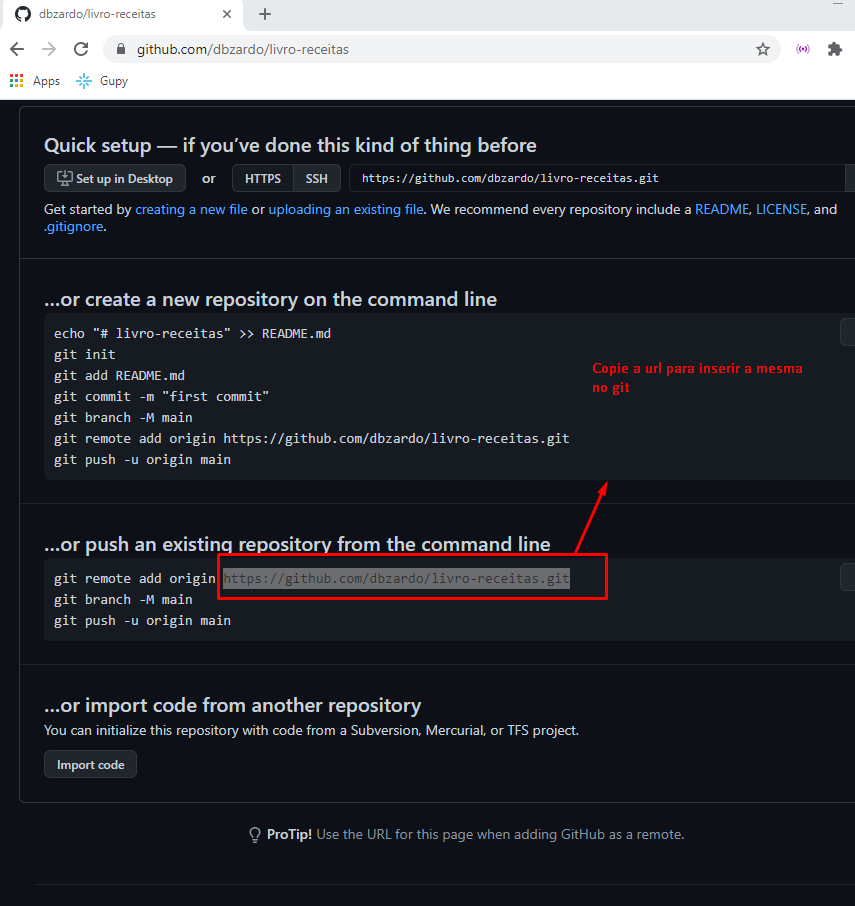
**GIT HUB**

**Criando um novo repositório**

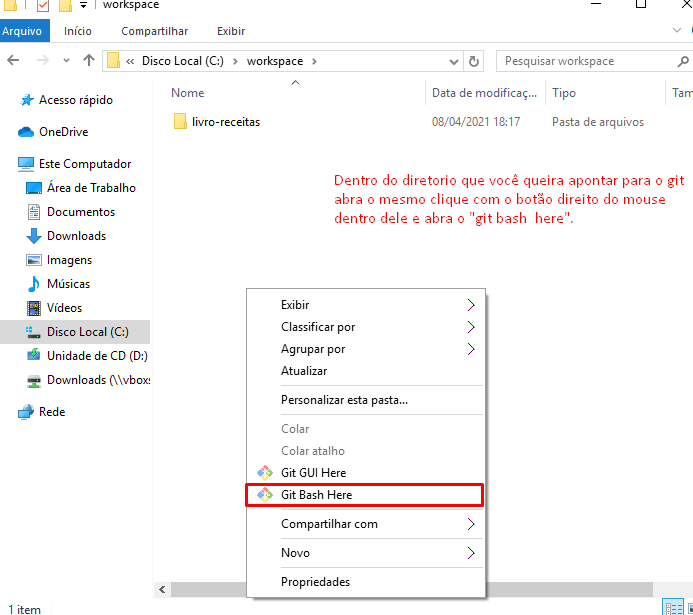


**Apontando o repositório local para o do github**

**1º passo**



**2º passo**



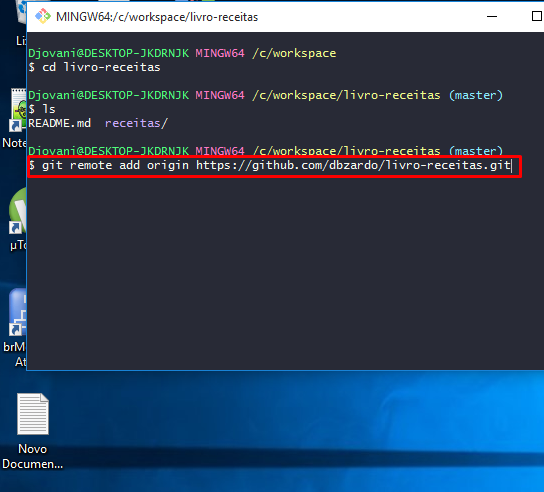
**3º passo:** Com o git bash aberto utilize o comando: **git remote add origin** [**https://github.com/dbzardo/livro-receitas.git**](https://github.com/dbzardo/livro-receitas.git)

OBS: o comando “**git remote add origin”** foi colocado junto com a URL copiada no passo 1

O comando: **git remote -v** , lista os repositórios remotos cadastrados

Allias = apelido

No caso o origin é um apelido dado para facilitar o manuseio para não precisar ficar digitando o link

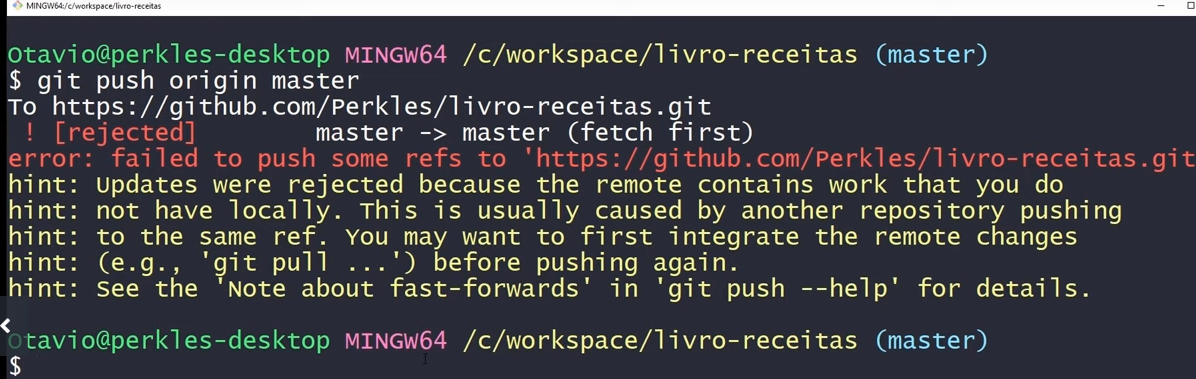


Utilize o comando abaixo, para importar o repositório local para o do github.  
**git push origin master**

**master** é branch.

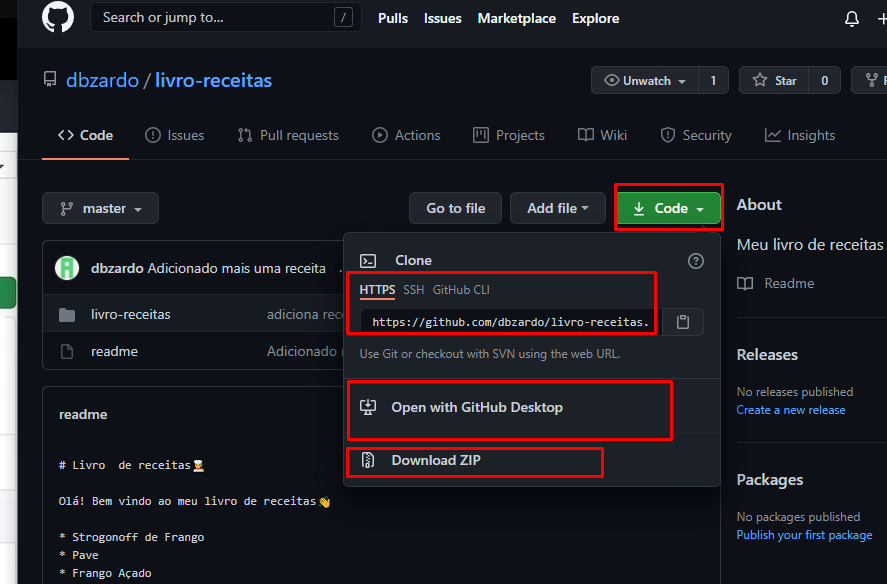
**Conflitos**

OBS: Erro quando, alguns trabalho que tem no remoto e que você não tem localmente.

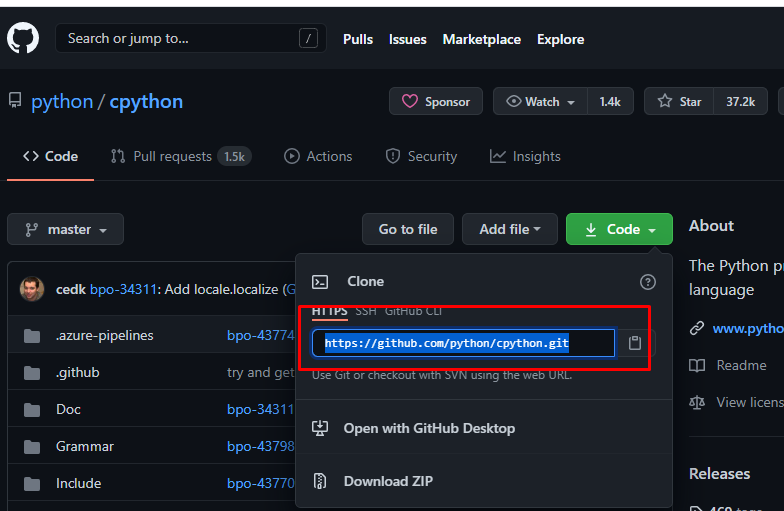


**Git pull -**  Utilizado para trazer (puxar) os objetos remotos para o local (sua maquina).

**Baixando códigos**



Copie o link do repositório e utilize ele com o comando **“git clone”**



Ex: **git clone** [**https://github.com/python/cpython.git**](https://github.com/python/cpython.git)

**OBS: git diff (Utilizado para verificar o que foi alterado do origal para o que foi alterado)**

**OBS1: git marge**