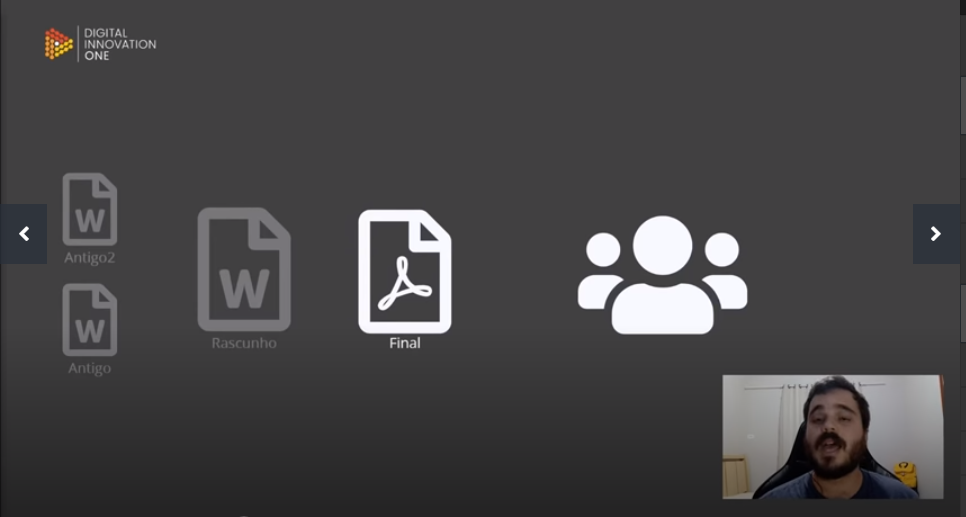
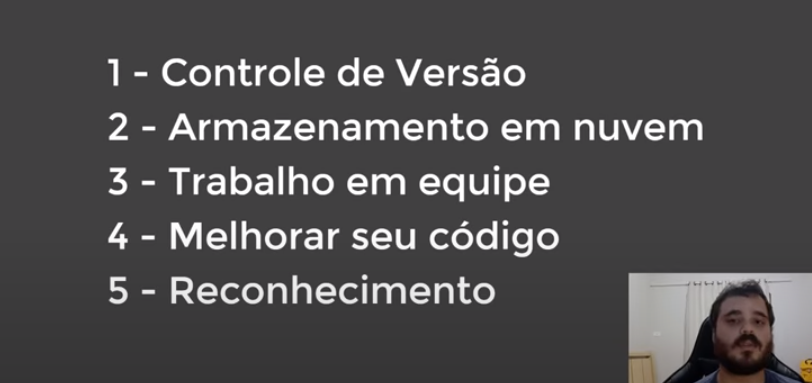
O por que do github

Possivelmente, quando trabalhamos com programação, haverão muitas versões do código.



Git e GitHub são tecnologias complementares, porém, diferentes.



Comando básicos e como instalar o git na máquina

* Comando para mudar de pastas
* Listar as pastas
* Criar pastas/arquivos
* Deletar pastas/arquivos

Windows

* Cd
* Dir
* Mkdir
* Del/rmdir

“Dir”: lista de diretórios contidos na pasta selecionada

Voltar pasta Windows, “cd /” ou cd ..

Acessar pasta “cd (nome da pasta)”

Limpar terminal

Windows cls

Tab auto completa, no widows se digitarmos cd w + tab, ele vai escrever o nome Windows.

Criar uma pasta no Windows, “mkdir nome da pasta”

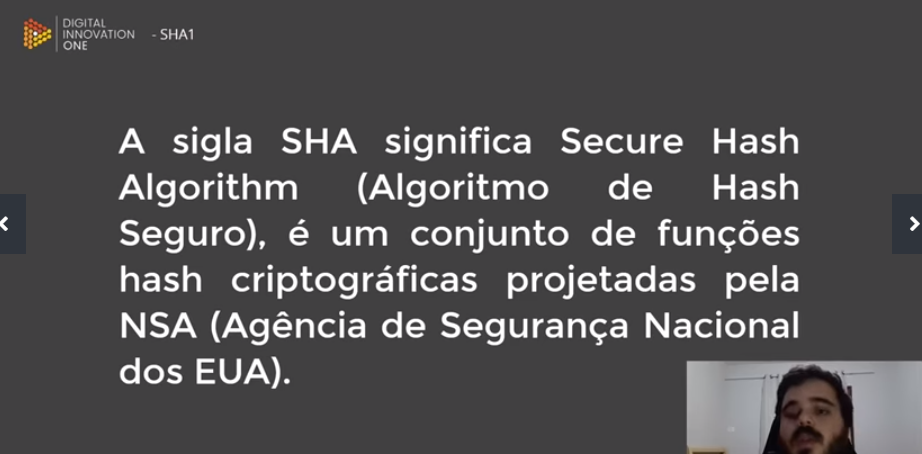
Echo hello, ele escreve echo hello, porém se adicionarmos > , esta seta é um redirecionador de fluxo, então, se fazermos echo hello > hello.txt, ele vai criar um txt, na pasta em questão, escrito hello.

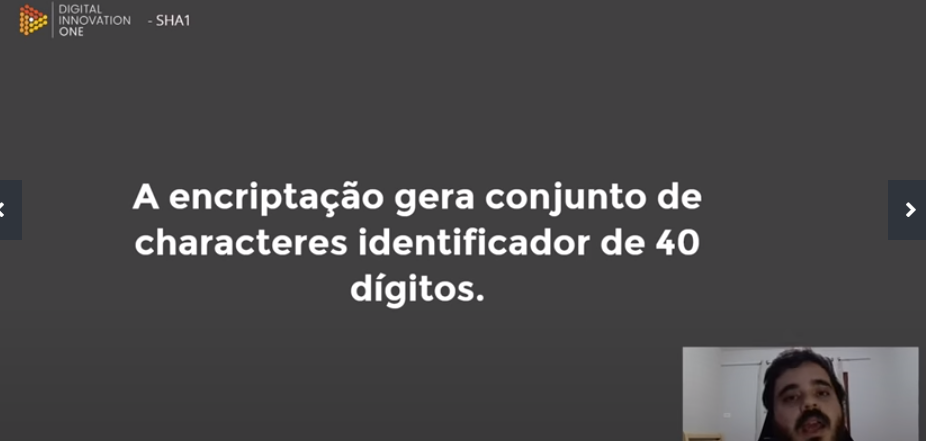
Deletar itens dentro de um diretório

Del workspace

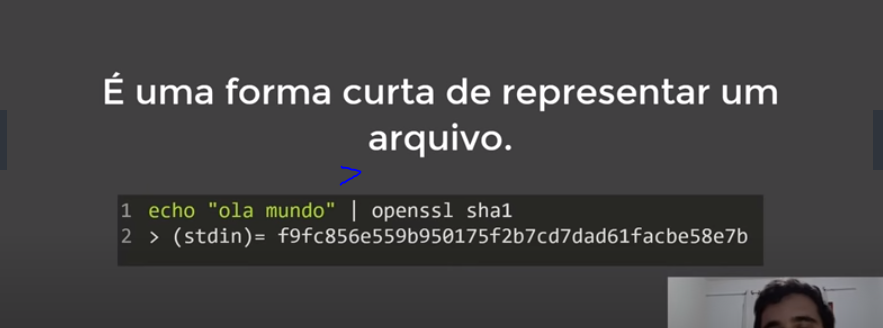
Para remover o diretório, usa-se rmdir workspace /s /q

Topicos fundamentais para entender o funcionamento do git





Esse conjunto de caracters funciona como um identificador

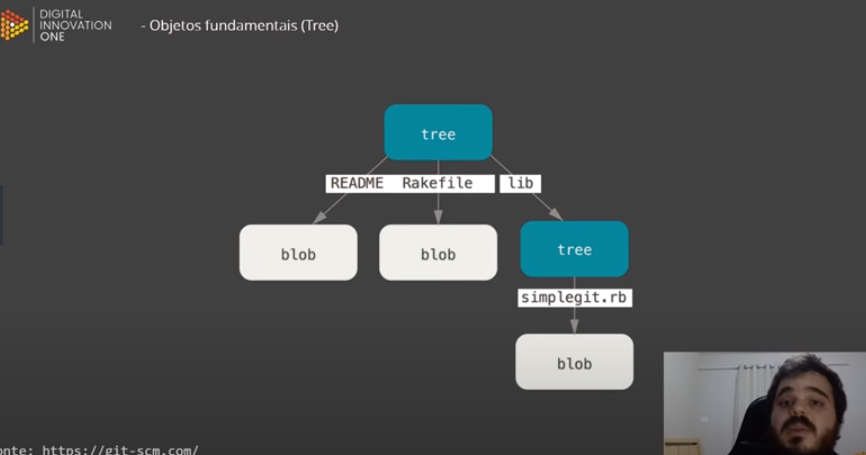


Objetos internos do git

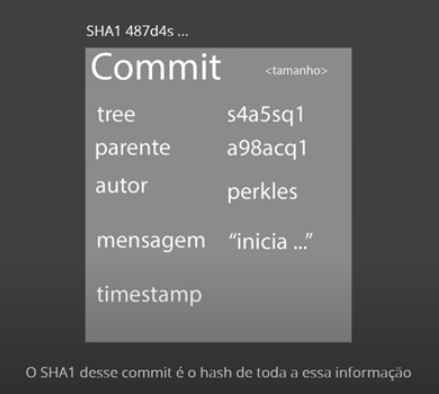
* Blobs
* Trees
* Commits

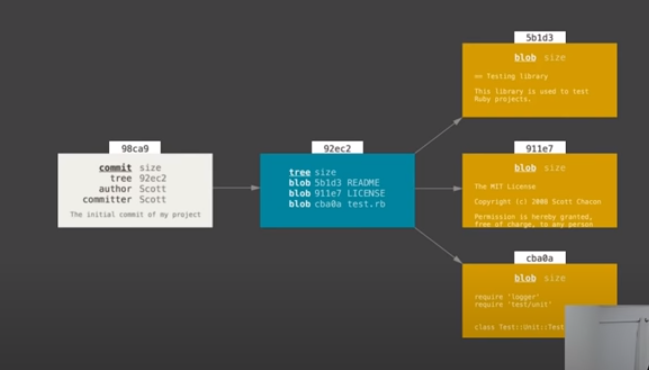
Blob contém metadados, ele vai ter um tipo, tamanho \0 e o conteúdo.

O segundo objeto é a Tree, tree armazenam blobs e commits, ela tbm contém meta dados

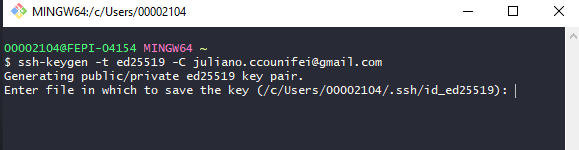


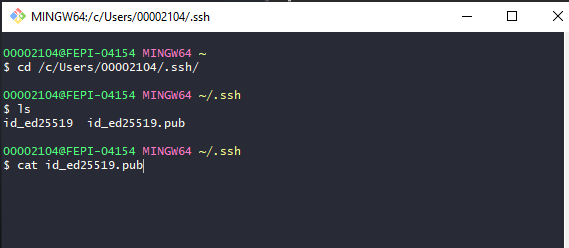
Commit, é um objeto que vai juntar tudo, dando sentido para a alteração que está sendo realizada.





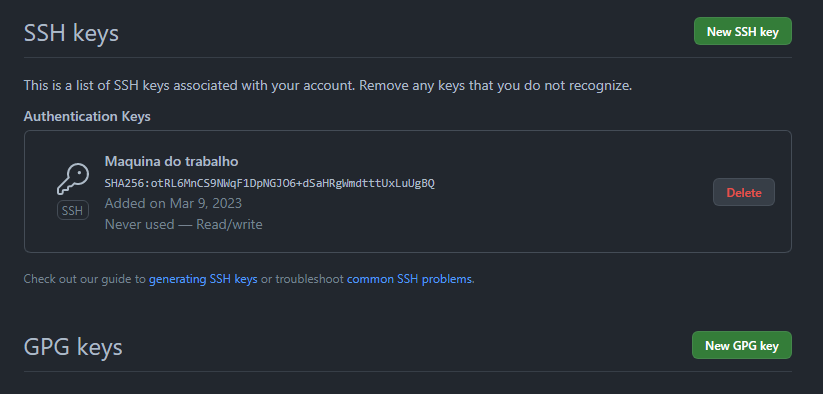
Chaves SSH e Tokens





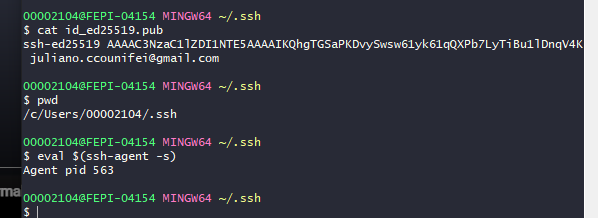
Cat é usado para visualizar o conteúdo da chave

Adicionando a chave no Github, vai ficar assim

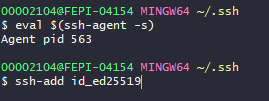


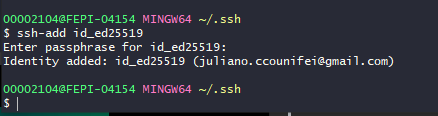
Agora existe mais um configuração a se feita no bash

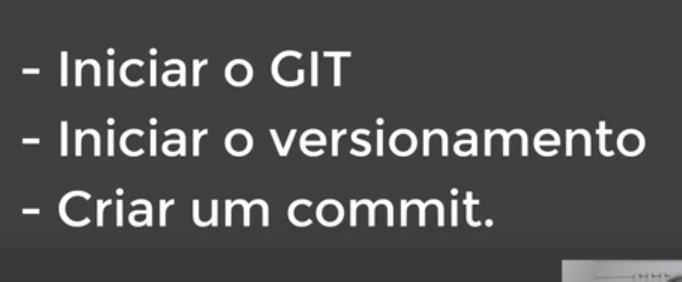
Vai ser preciso iniciar o agent no bash

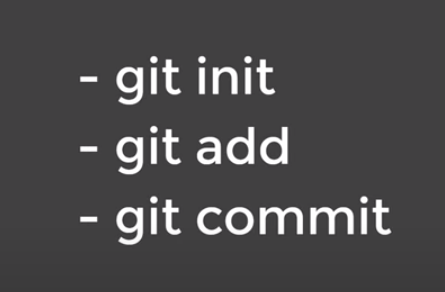


Para passar pro agent, usamos a chave privada.

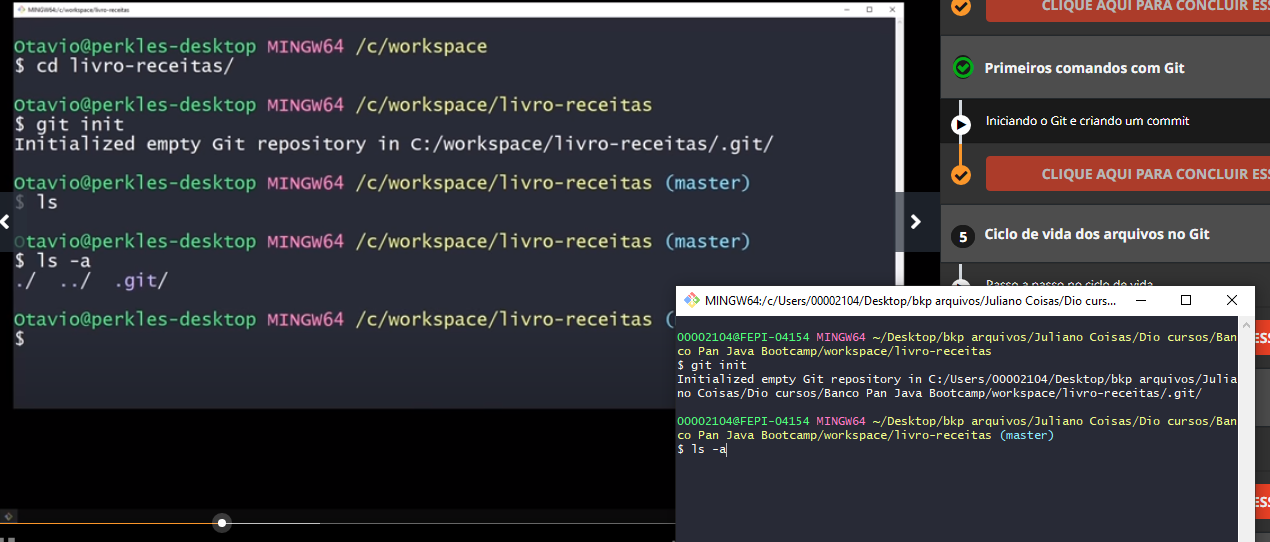




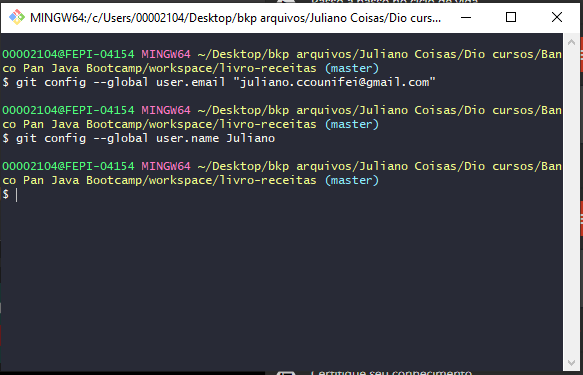




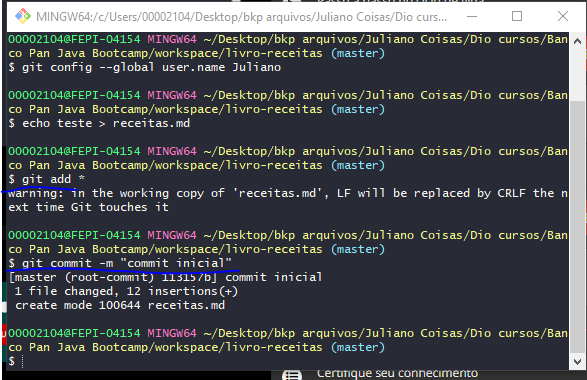
O comando -a mostra arquivos ocultos



Criando um usuário e email global

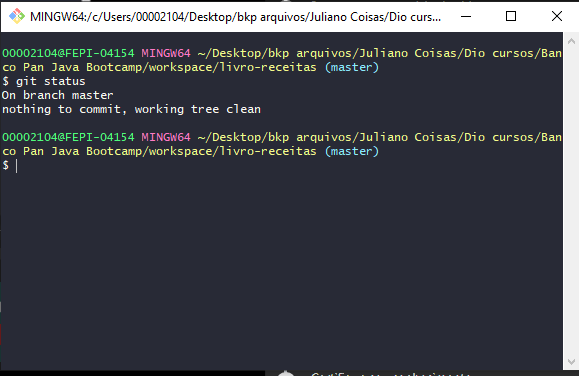




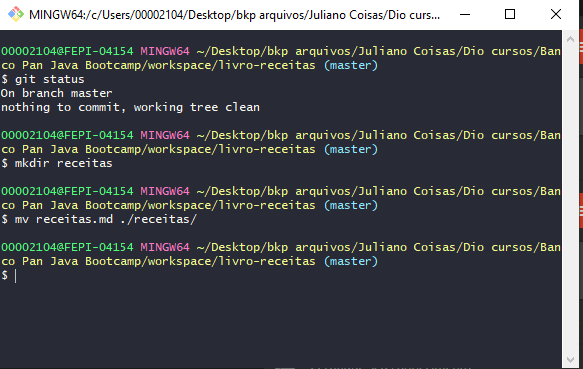


Ciclo de vida dos arquivos no git

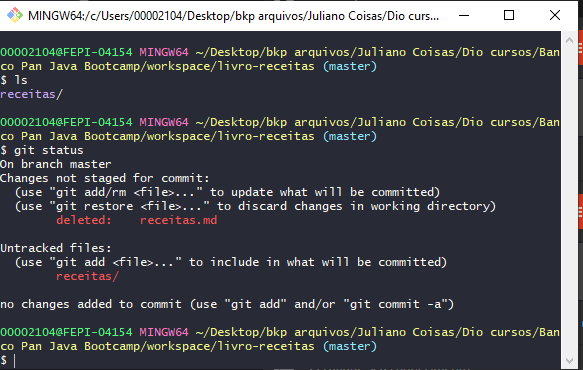
Com o git init, criamos um repositório dentro do git



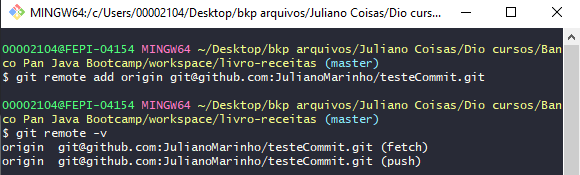
Git status, vê o status dos arquivos na pasta em relação ao git

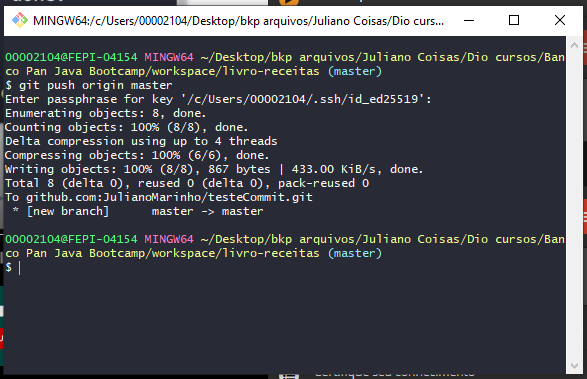


Movendo arquivos no git comando mv



Adicionar a url do github

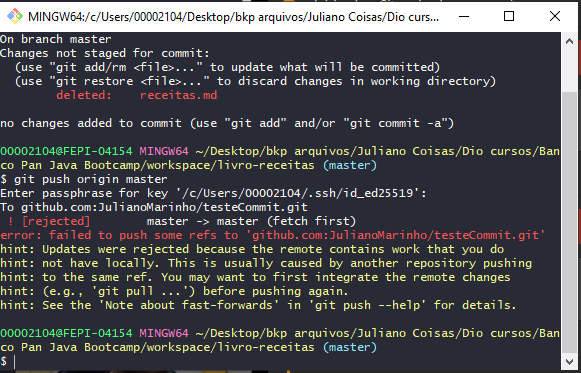




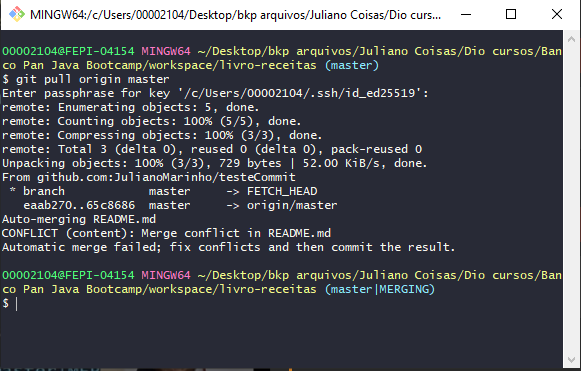


Exemplo hipotético: vc tem o código no git, uma outra pessoa acessa seu repositório e o clona, neste momento ninguém mexeu em código ainda. Em seguida vcs abrem o arquivo para editar, neste ponto, os arquivos vão passar a ser diferentes, uma pessoa pode ter alterado uma mesma linha que vc, e essa pessoa envia o código para o github, neste ponto o código q esta no github é mais atualizado que seu código.

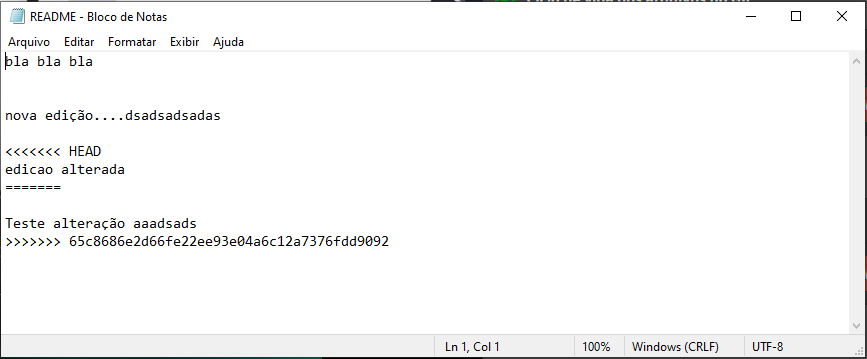
Dai vc envia seu código para o github, o github vai dar a seguinte mensagem:

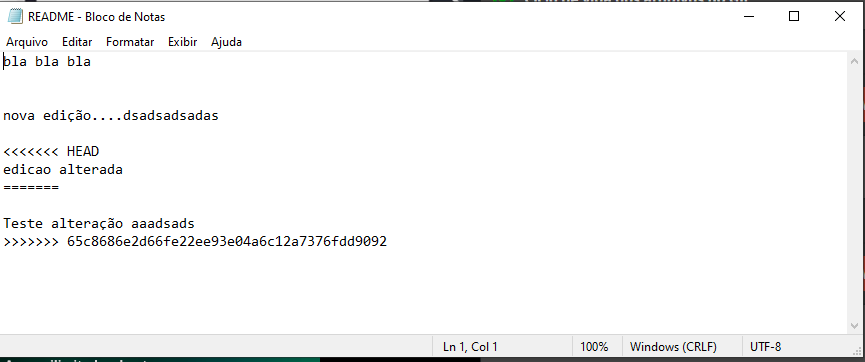


Aqui vc terá que integrar ao seu arquivo o que foi modificado antes de enviar, usando o git pull

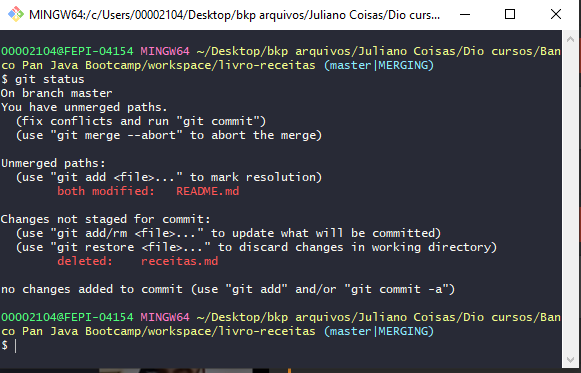


Usando o pull, ele vai falar, ali em baixo, que existe um conflito de merge. Abrindo o arquivo, ele vai estar assim:

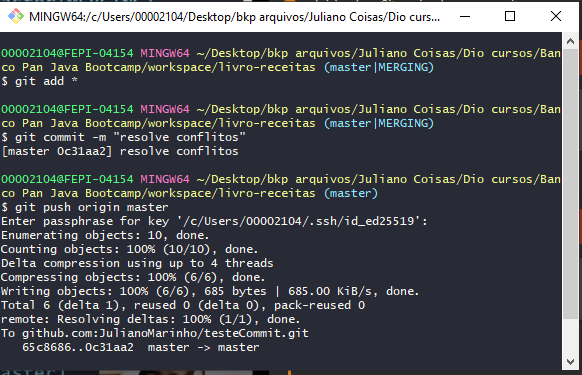




Aqui no head diz a alteração que está la no github



Ainda vai falar que existem 2 arquivos, para resolver isso, temos q usar o commit



Git clone + url do repositório

O que diferencia uma pasta comum de um repositório é a pasta git

