Git e GitHub

Git = O git é um sistema de versionamento de arquivos. Com ele você pode fazer alterações simultaneamente com um colega de projeto sem que dê problema. E também tem o histórico das alterações feitas, caso queira voltar atras.

- Commit: é o ato de enviar uma atualização feita no projeto de modo local para o repositório GitHub.

- Branch: é quando você cria uma ramificação dentro do seu projeto de algo que você não tem certeza que irá entrar. Portando você a salva no seu repositório através do commit em uma Branch

- Merge: Quando você decide que a ramificação do projeto está pronta e decide implementá-la no projeto principal, você faz isso através do merge, ou até mesmo fazer o merge com outra Branch

- Remote: É basicamente quando você linka a sua pasta local com o repositório online onde ele estará “hospedado”, e através dos commits pode atualizá-lo.

- Push: é o ato de enviar os arquivos atualizados para o seu repositório

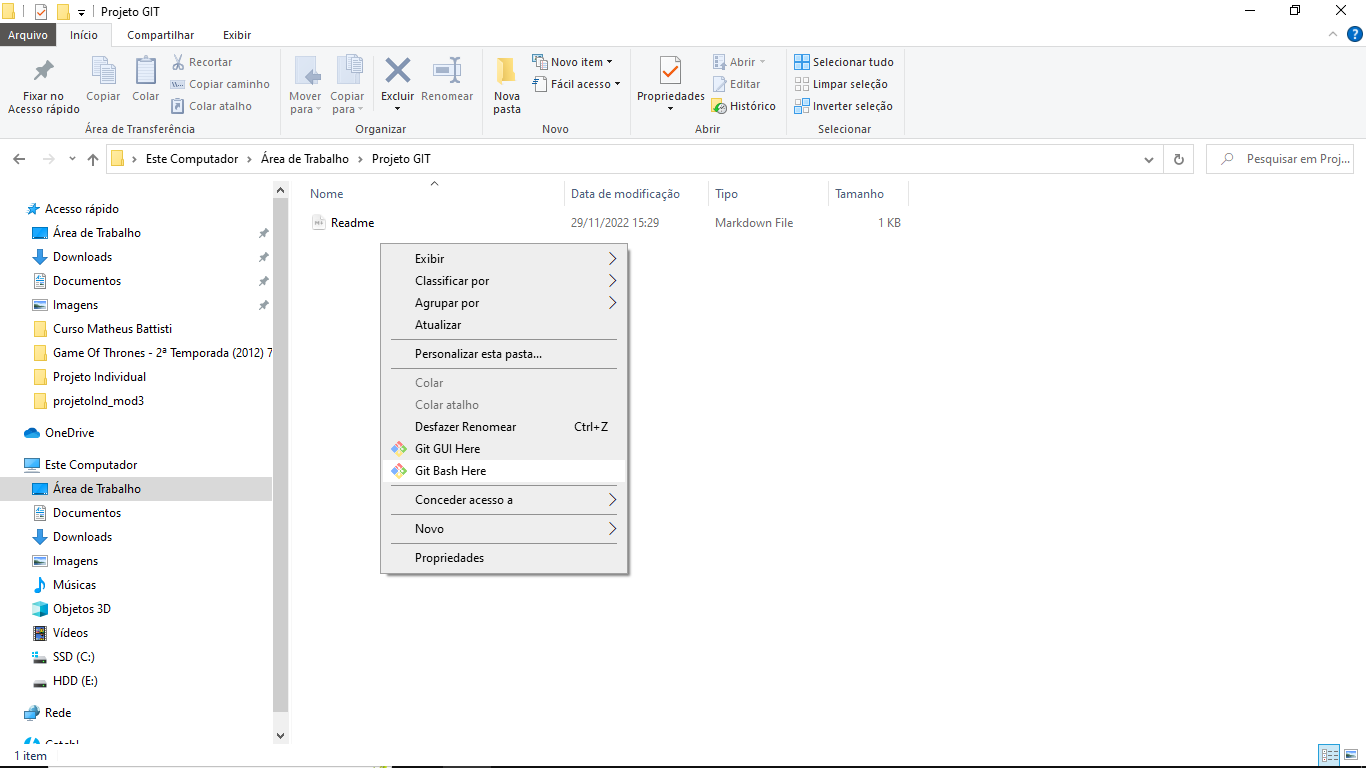
- Pull: é o ato de pegar o que está no repositório do GitHub para a sua máquina

GitHub = É a plataforma onde você hospeda os seus repositórios para alteração no Git, mas além disso é uma rede social de desenvolvedores. Onde você consegue encontrar outros projetos

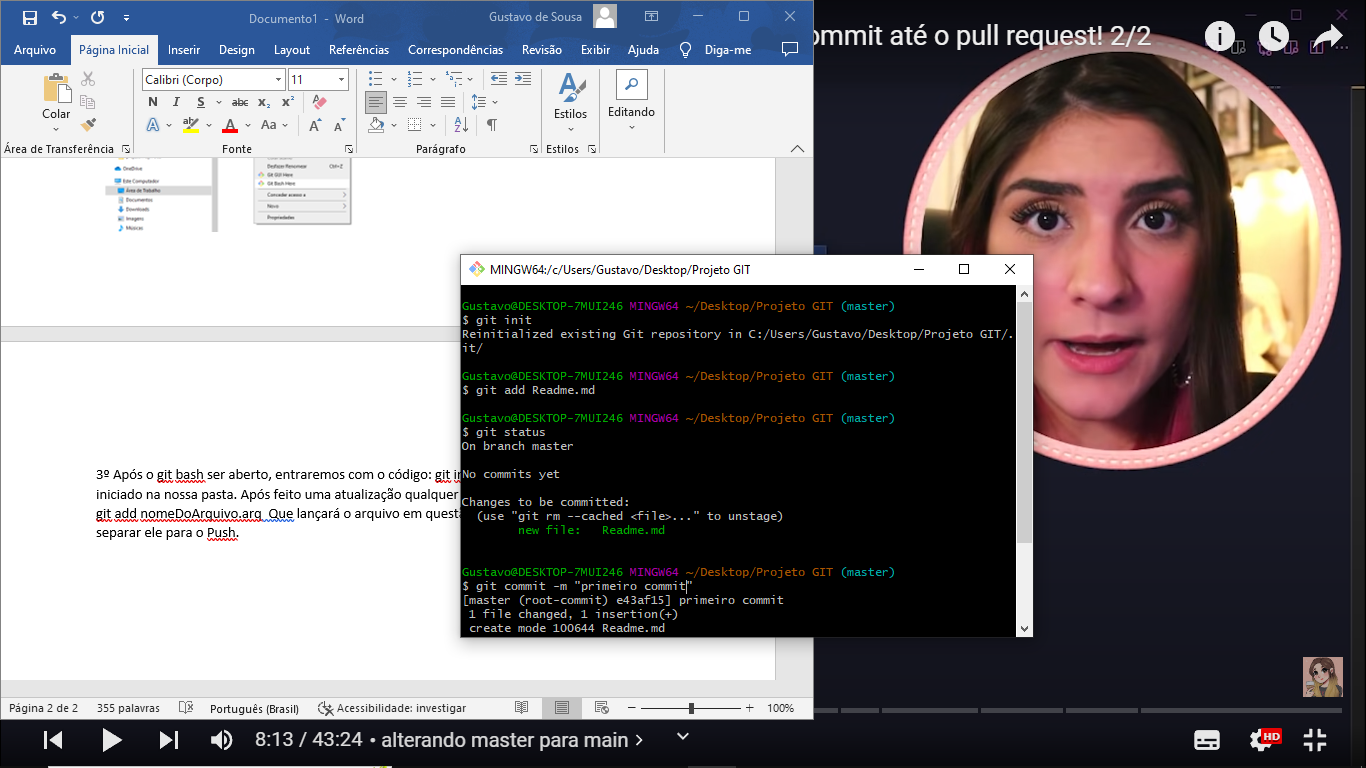
Vídeo 2 – Na prática com Git

1º Criamos a pasta do nosso projeto e dentro dela inserimos um arquivo chamado Readme.md  
 - Readme.md é um arquivo presente em muitos projetos e são as instruções e informações que as pessoas que acessam seu projeto precisam saber (como se fosse um manual de instruções)

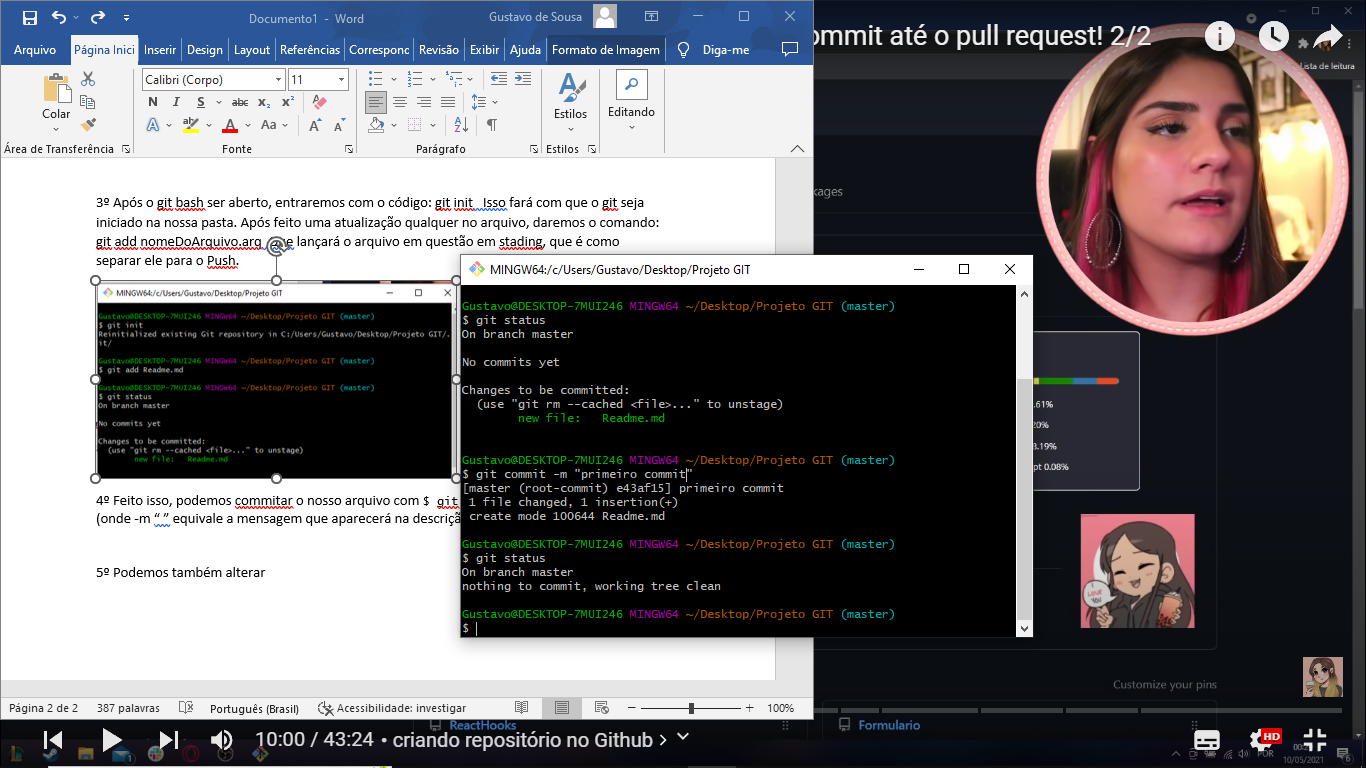
2º Após o arquivo Readme.md ter um conteúdo, acessaremos sua pasta, clicaremos com o direito e escolheremos a opção Git Bash Here (que já abrirá o git bash com a pasta que iremos trabalhar)



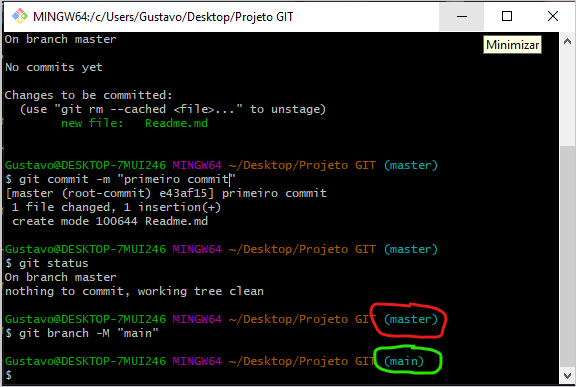
3º Após o git bash ser aberto, entraremos com o código: git init Isso fará com que o git seja iniciado na nossa pasta. Após feito uma atualização qualquer no arquivo, daremos o comando: git add nomeDoArquivo.arq Que lançará o arquivo em questão em stading, que é como separar ele para o Push.

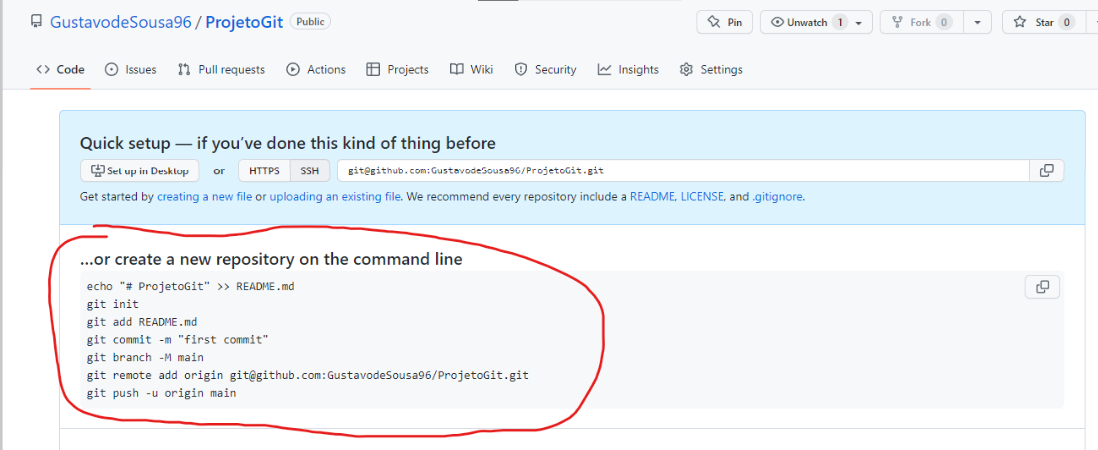


4º Feito isso, podemos commitar o nosso arquivo com $ git commit -m "primeiro commit" (onde -m “ ” equivale a mensagem que aparecerá na descrição do commit) E agora os arquivos estão em um repositório na nossa máquina.



5º Podemos também alterar a raiz de onde o arquivo ficará mudando de “master” para “main”. Após uma atualização do GitHub o nome que passou a ser utilizado foi o main.



6º Criando um repositório no GitHub: Vá em repositórios, depois nisso em New e crie. Após o repositório ser criado o GitHub nos mostras alguns comando que usaremos no git para linkarmos o repositório da nosso arquivo na nossa máquina para o repositório online. Basta seguirmos o passo a passo e conseguiremos linkar.  


6.1º Já alteramos a Branch para Main. Agora iremos linkar o repositório com a pasta de trabalho. Para isso daremos o comando: git remote add origin url.HTTPS.do.repositório (Nesse caso ficou:

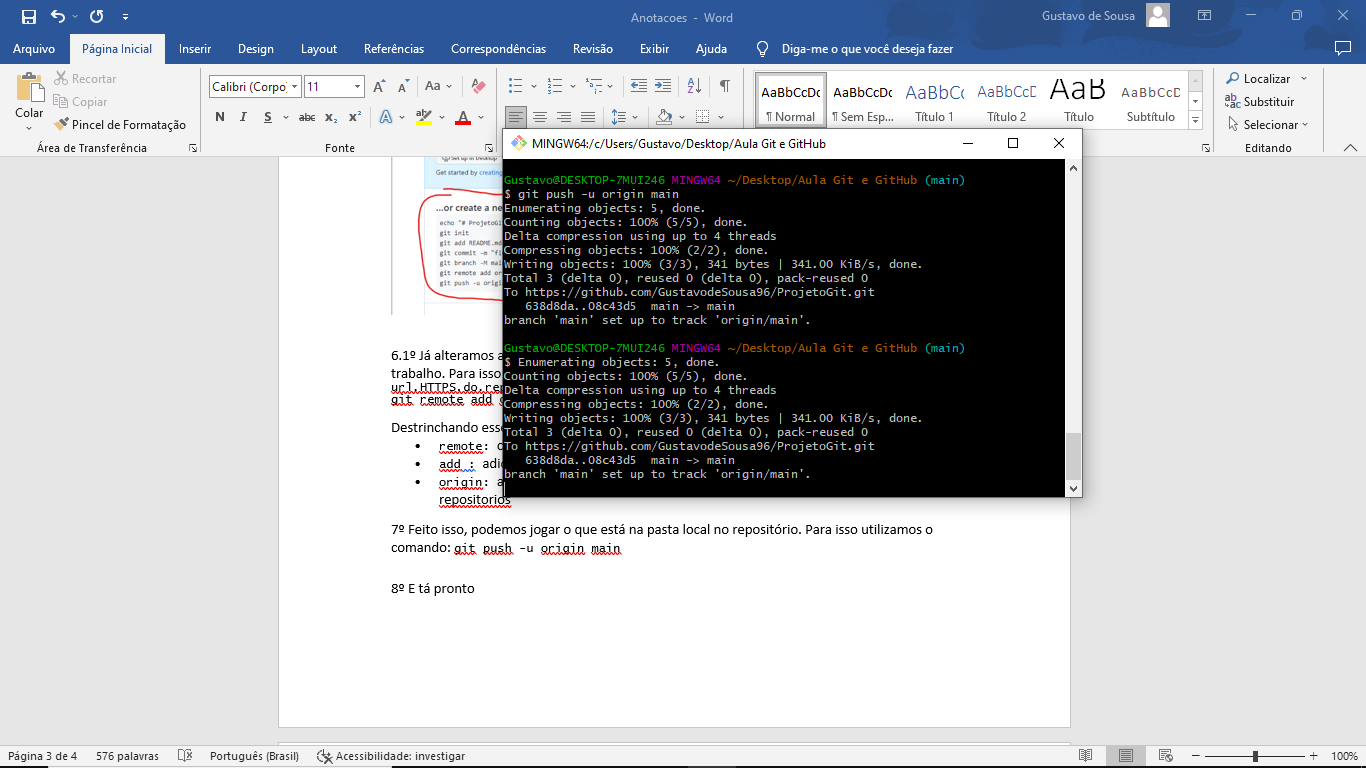
git remote add origin <https://github.com/GustavodeSousa96/ProjetoGit.git> )

Destrinchando esse comando:

* remote: comando Deixa a pasta com o acesso remoto ao repositorio
* add : adiciona o arquivo
* origin: apelido dado ao repositório online. Esse é o pode ser usado em todos os repositórios

Esse comando só é dado uma vez, pois depois de linkado com o repositório ele fica atrelado a ele.

7º Feito isso, podemos jogar o que está na pasta local no repositório. Para isso utilizamos o comando: git push -u origin main



8º E tá pronto, é só atualizar a página do seu repositório que poderá visualizar que o arquivo já subiu para ele.

Alterando e Adicionando arquivos por Git

1º Para isso iremos ao git após as atualizações serem feitas e temos duas possibilidades de comando:

* **git add** **<nome do arquivo>** (sem sinais de menor e maior)= Para mandar apenas um arquivo que foi alterado ou adicionado para a área de standing;
* **git add** **.** = Esse comando manda todos os arquivos que sofreram alteração ou foram adicionados a área de standing.

2º Segundo passo é dar o commit com o mesmo comando que já vimos: git commit -m “texto que quiser”