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

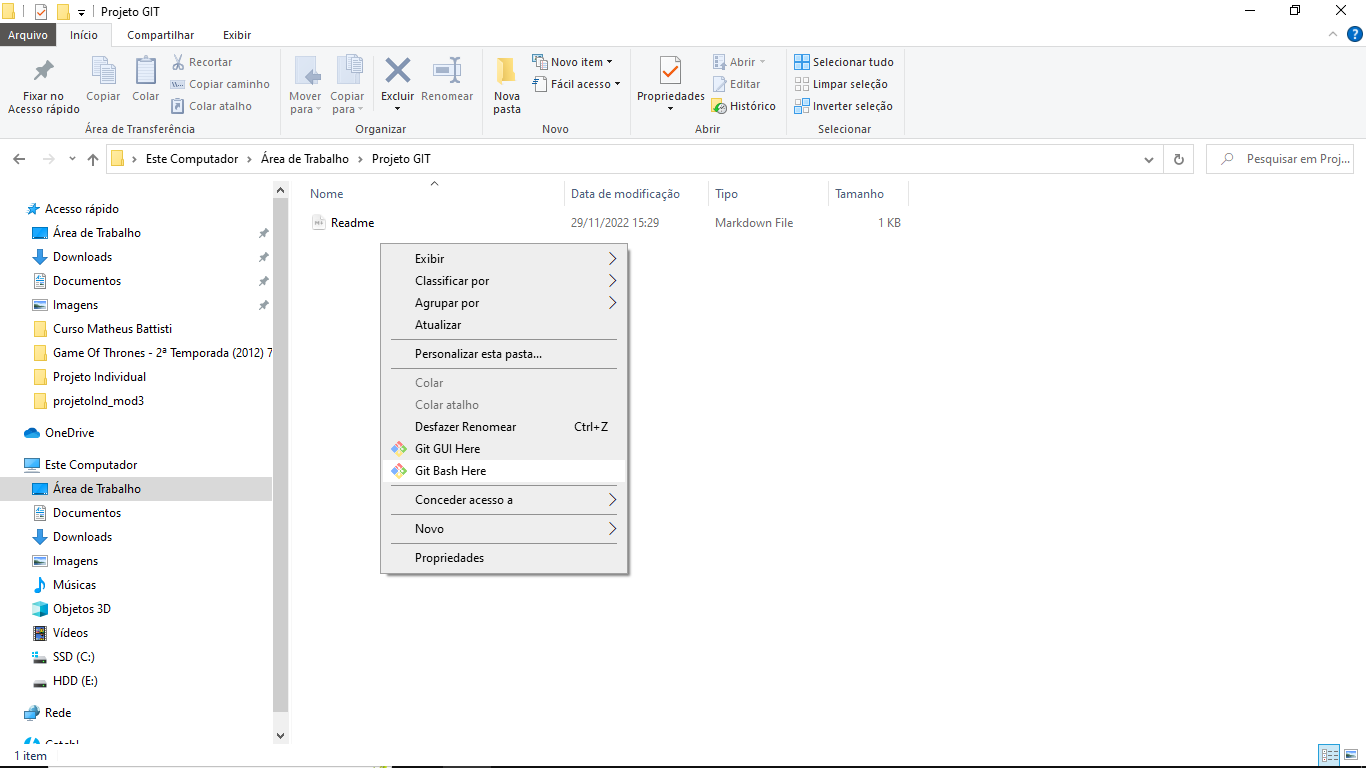
- Pull: é o ato de pegar o que está no repositório do GitHub para a sua máquina

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

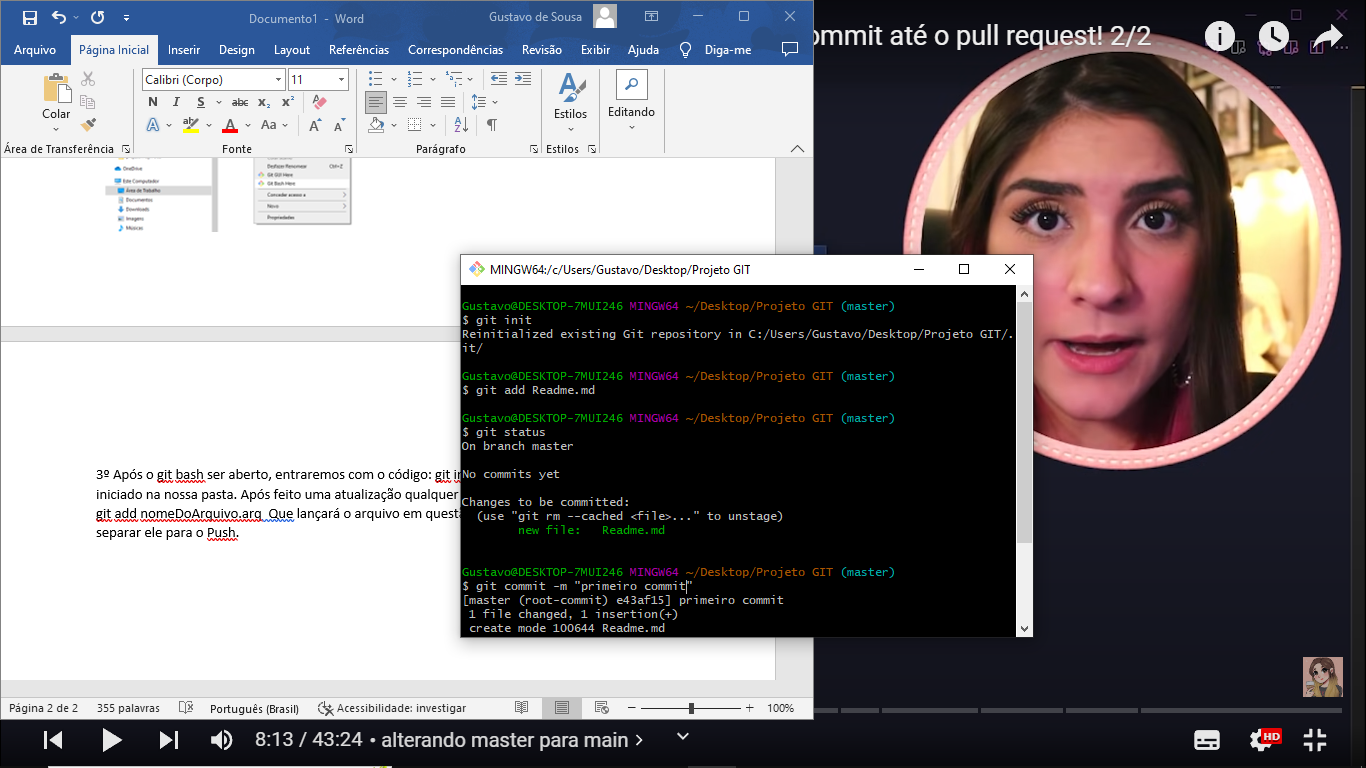
Vídeo 2 – Na prática com Git

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)

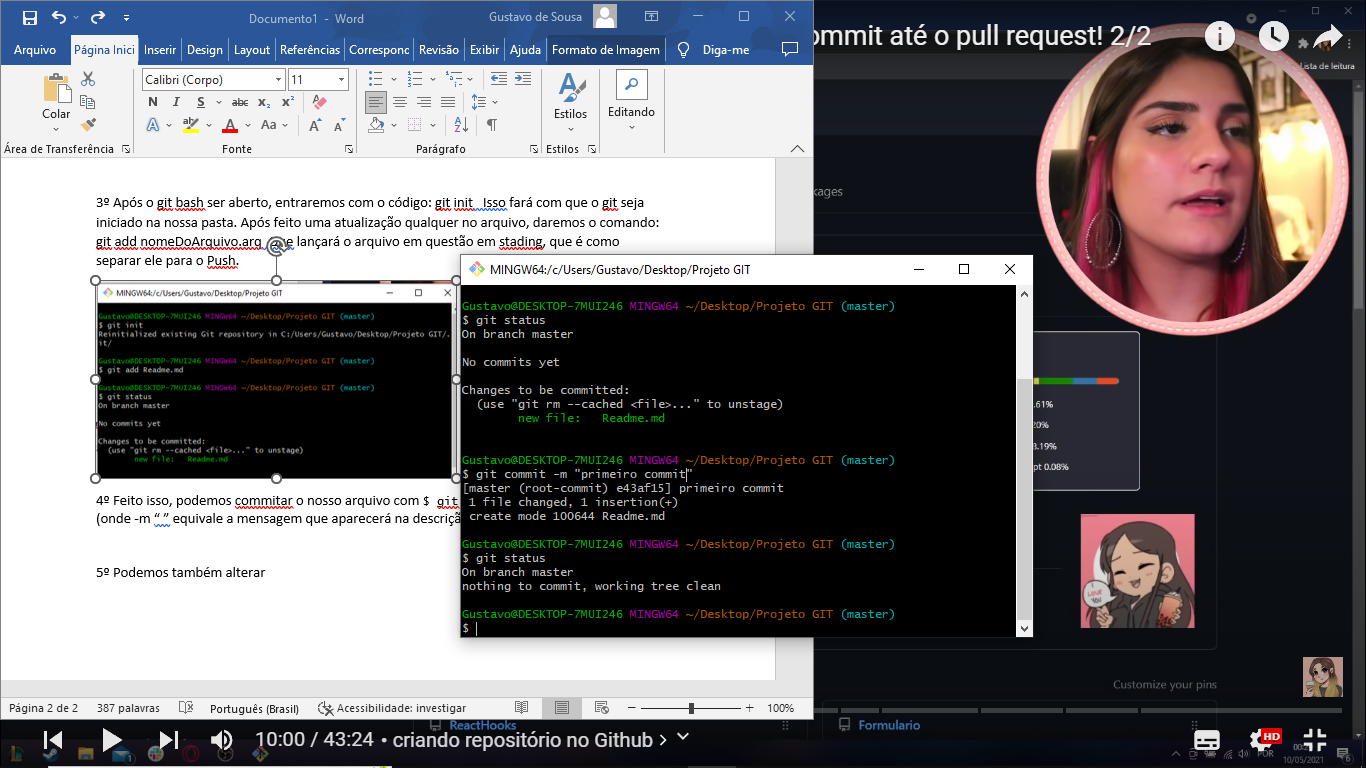
2º Após o arquivo Readme.md ter um conteúdo, acessaremos sua pasta, clicaremos com o direito e escolheremos a opção Git Bash Here (que já abrirá o git bash com a pasta que iremos trabalhar)



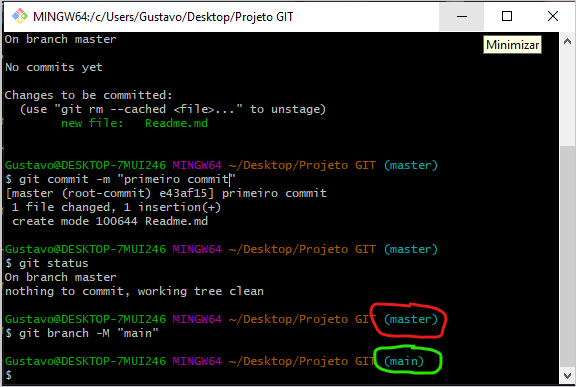
3º Após o git bash ser aberto, entraremos com o código: git init Isso fará com que o git seja iniciado na nossa pasta. Após feito uma atualização qualquer no arquivo, daremos o comando: git add nomeDoArquivo.arq Que lançará o arquivo em questão em stading, que é como separar ele para o Push.

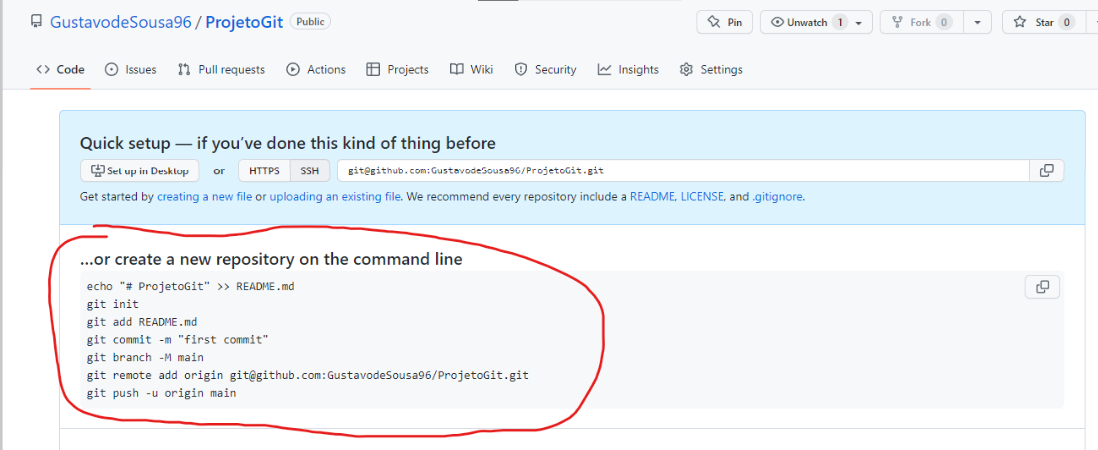


4º Feito isso, podemos commitar o nosso arquivo com $ git commit -m "primeiro commit" (onde -m “ ” equivale a mensagem que aparecerá na descrição do commit) E agora os arquivos estão em um repositório na nossa máquina.



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:

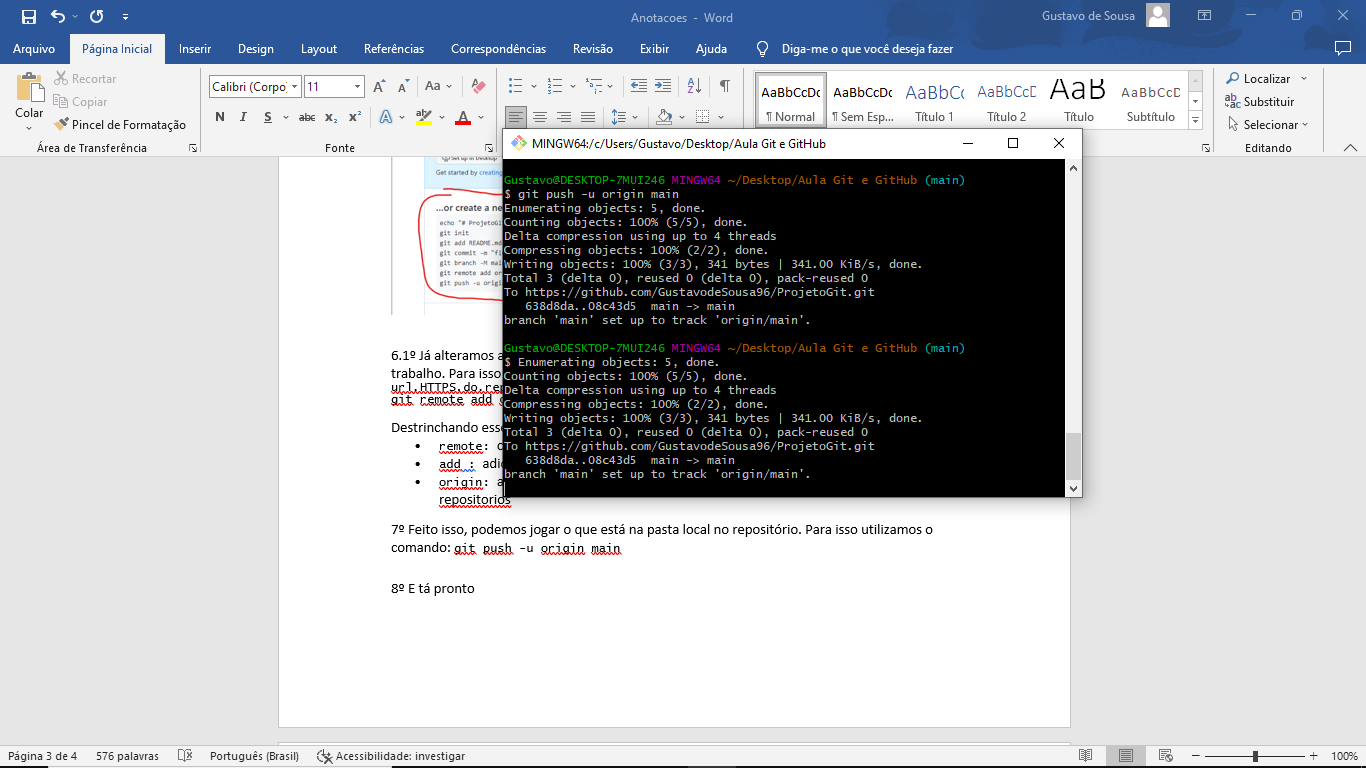
git remote add origin <https://github.com/GustavodeSousa96/ProjetoGit.git> )

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

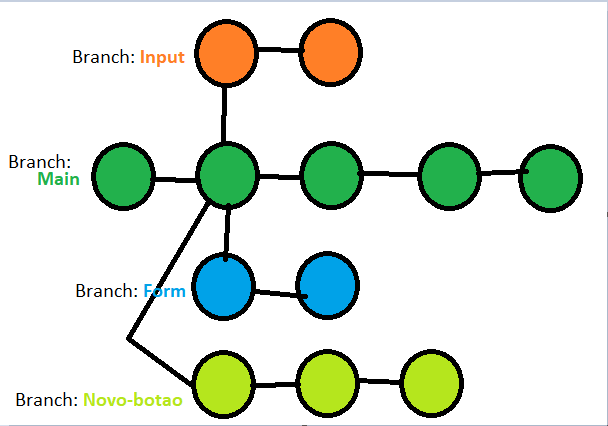
1º Para isso iremos ao git após as atualizações serem feitas e temos duas possibilidades de comando:

* **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”

Entendendo as Branchs

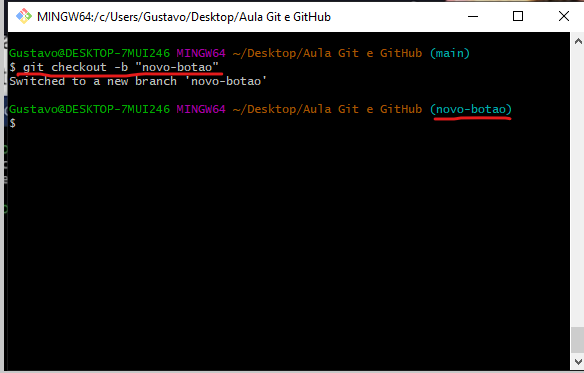
As Branchs são caminhos onde a Main é o caminho principal do seu projeto e os outros são caminhos provisórios, criados para fazer alterações ou adicionar arquivos sem alterar a Main. Se a atualização feita em uma Branch secundária for ser utilizada na main, nós fazemos sua transferência através do merge.



E podemos criar uma nova Branch da seguinte forma:

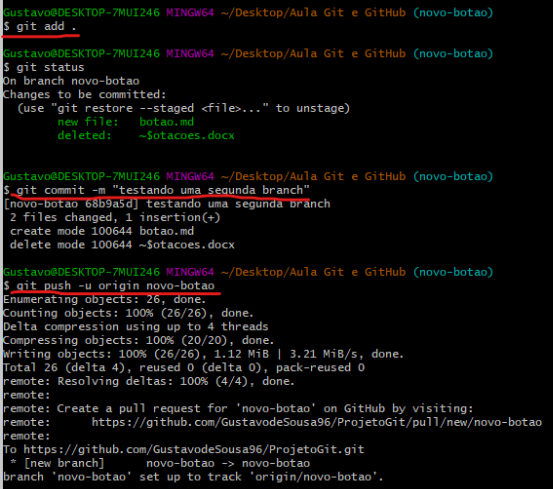
1º Crie a nova Branch antes de fazer as alterações.

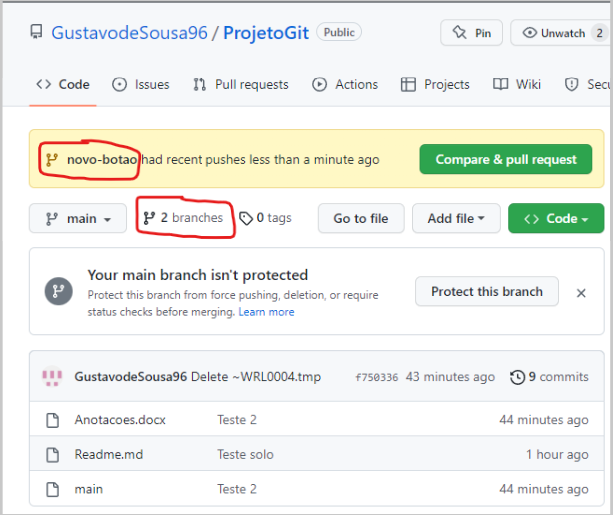
2º No git bash dê o seguinte comando: git checkout -b “nome-da-nova-branch” . Com isso a nova Branch será criada e você já será transferido para ela.



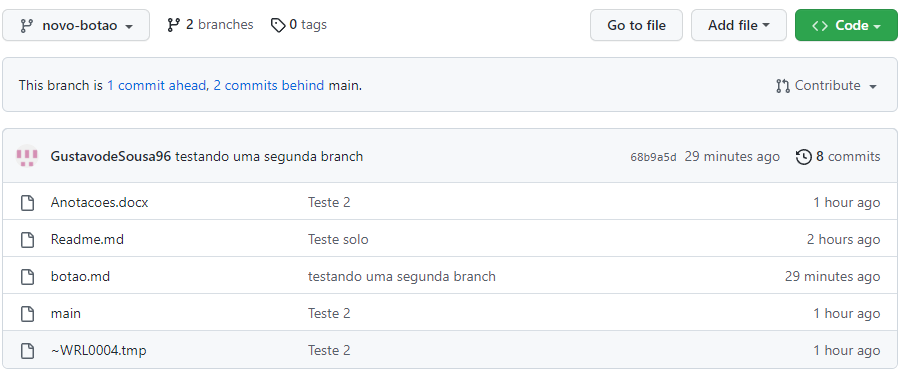
2º E agora para fazer o push dos arquivos é seguindo os mesmos passos.

* git add . (para transferir todos arquivos para o standing;
* git commit -m “ “ (para identificar do que se trata esse commit;
* git push -u origin nomeDaBranch (nesse caso seria: git push -u origin novo-botao



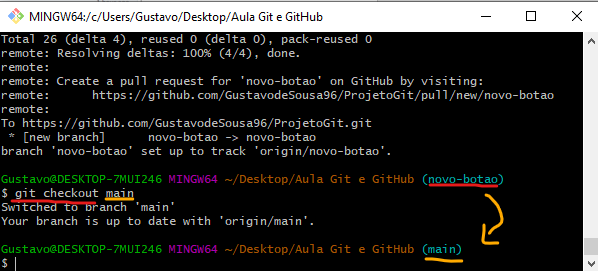


Como podemos ver abaixo, os arquivos da Branch principal(main) são repetidos. E qualquer alteração feita neles aqui, não interfere no arquivo principal, pois está sendo feita numa Branch secundária.



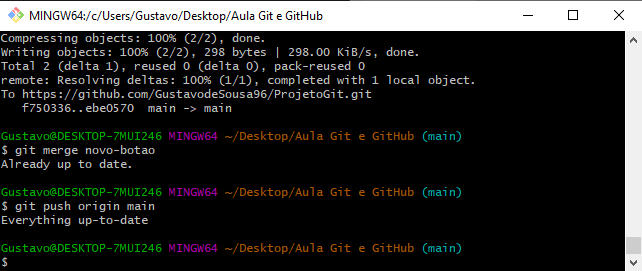
É possível transitar entre as branchs, fazemos isso usando o comando:

git checkout nomeDaBranch



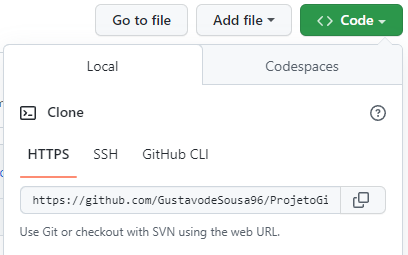
Para unirmos uma Branch secundária a Branch principal(main) usamos o comando **merge.  
Veja como:**

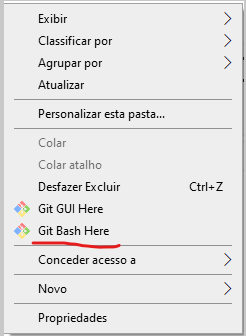
1. Entre na Branch principal(main) caso já não esteja nela.
2. E colocamos o código: git merge nomeDaBranch
3. Por fim daremos um: git push origin main . Para atualizar a nossa main do GitHub

****

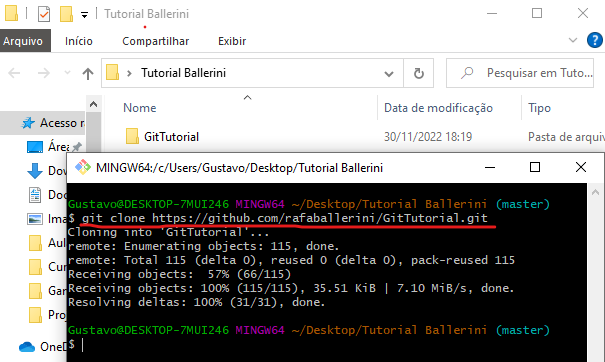
Utilizando a ferramenta Clone

O GitHub nos permite baixar os repositórios públicos de outros usuários e nossos próprios repositórios em outras máquinas, utilizando a ferramenta clone.  
Passo a Passo:

1. Entre no repositório da sua escolha e clique em code;
2. Ele abrirá um shotdown com a opção de clonar;  
   
3. Copiaremos esse link HTTPS
4. Escolhemos uma pasta onde iremos clonar o repositório;
5. Clicaremos com o direito dentro da pasta e depois em GitBash Here



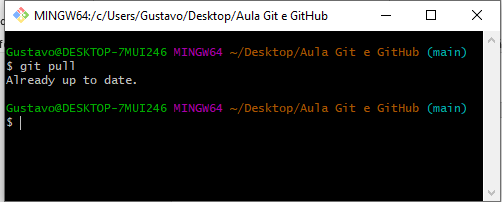
1. Após abrir o git digite o comando: git clone link.git (aqui entra o link que copiamos)



1. E é isso, está clonado.

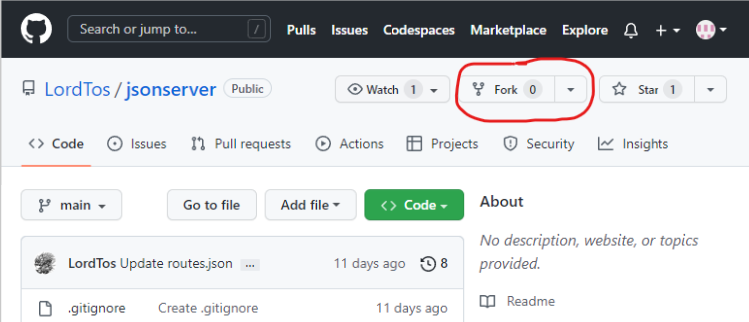
Fazendo o Pull

O Pull é a **atualização** do repositório local com o repositório online. Simplificando, vamos supor que você está em um projeto em grupo e outra pessoa faz uma alteração nele. O Push como vimos faz o “upload” para o GitHub já o Pull faz o “download”. Ele baixa as atualizações no projeto para o seu repositório local. Como podemos fazer isso?

1. Para isso basta abrir a pasta onde está o projeto, clicar com direito e abrir o Git Bash Here;
2. Feito isso coloque o comando: git pull
3. Que baixará qualquer atualização que não esteja no seu repositório local  
   

Fork, para quê serve?

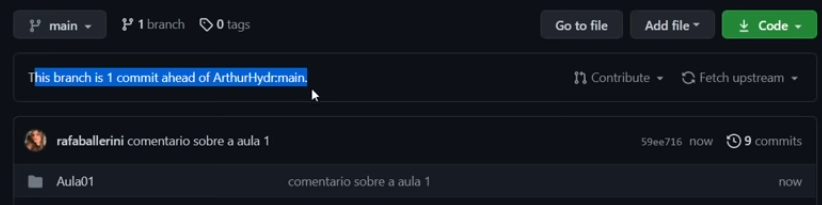
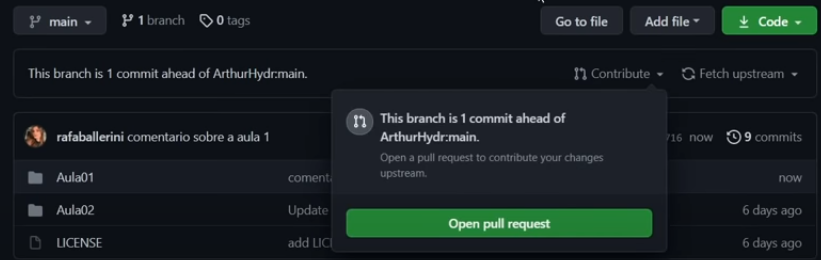
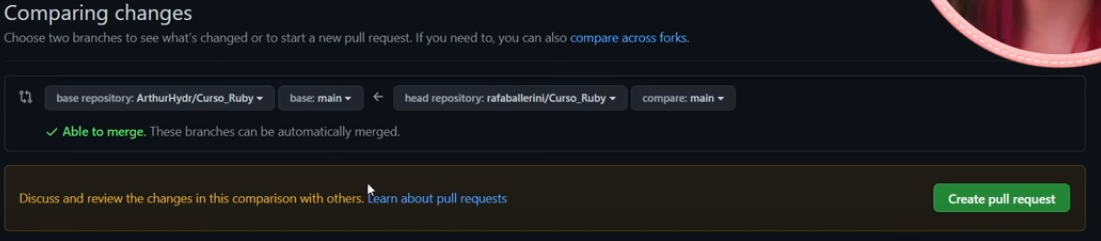
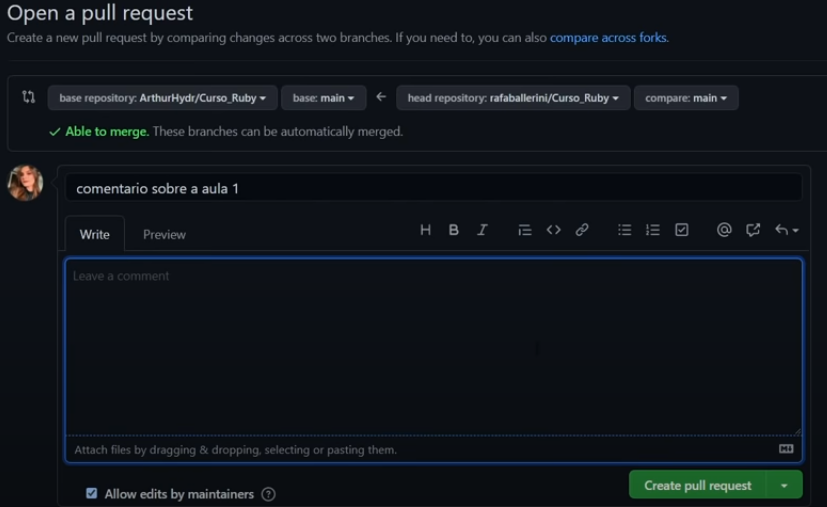
O Fork é como se fosse uma cópia do repositório de outra pessoa nos seus repositórios, assim você pode fazer alterações e melhorias no repositório de outra pessoa sem impactar no repositório original. Como fazer?

1. Para isso basta ir no repositório que pretende copiar e clicar em fork  
     
   \*Lembrando que os forks ficam registrados tanto no seu repositório copiado, quanto no repositório original\*

Pull Request o que é?

O Pull Request é uma solicitação que o usuário A envia ao usuário B do GitHub após realizar alguma alteração em uma Fork do usuário B. Essa solicitação é um pedido para implementar no projeto principal alguma alteração/modificação/implementação que o usuário A fez em uma Fork do usuário B.   
Se o usuário B aceitar, o seu projeto irá ser modificado com as alterações que o usuário A fez.

Como fazer um Pull Request:

1. Após ser feito algum tipo de alteração em uma Fork que você pegou, aparecerá essa mensagem: “Essa Branch tem 1 commit a mais que o projeto principal”  
   
2. Então você clica em “Contribute” e em “Abrir um pull request”  
   
3. Depois você confirma a criação do pull request  
   
4. E na próxima página você precisa detalhar o que fez, como essa alteração pode melhorar o projeto original e até mesmo como a pessoa pode testar o que você fez.  
   
5. Após detalhar tudo, clique em Create pull request. Aí é só aguardar se a solicitação vai ser aceita ou não.