



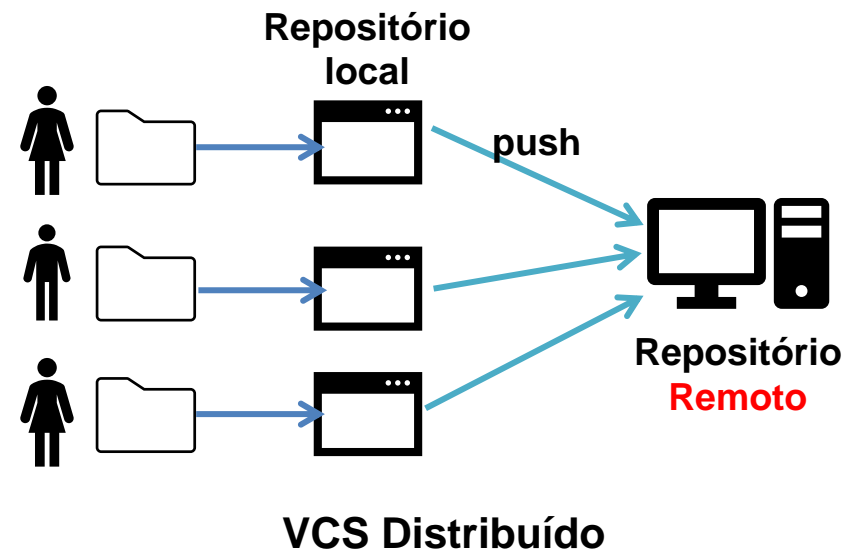
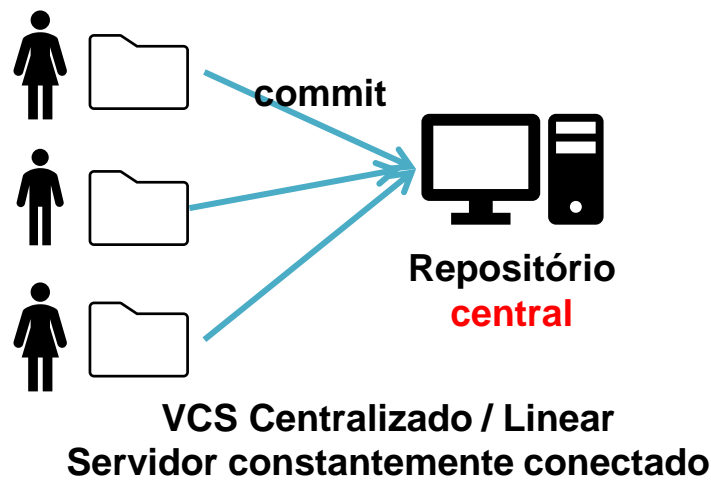
CIÊNCIA DA
COMPUTAÇÃO
GRANDE FLORIANÓPOLIS


Git e GitHub Como utilizar



Profa Daniela Souza Moreira, M. Sc

- Software de controle de versão (VCS) (gerenciar as versões)
- Utilizado para versionamento (não só código)
- Git \neq GitHub

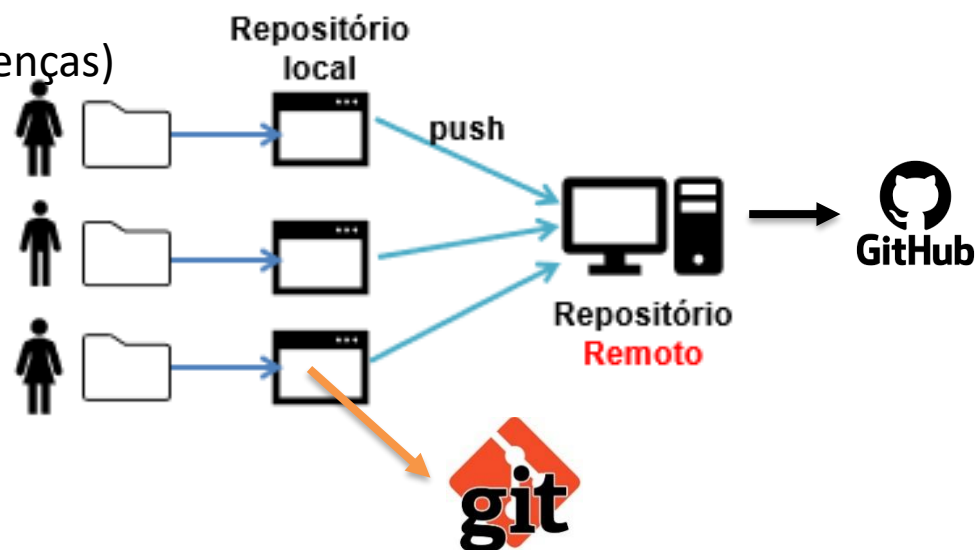


Git

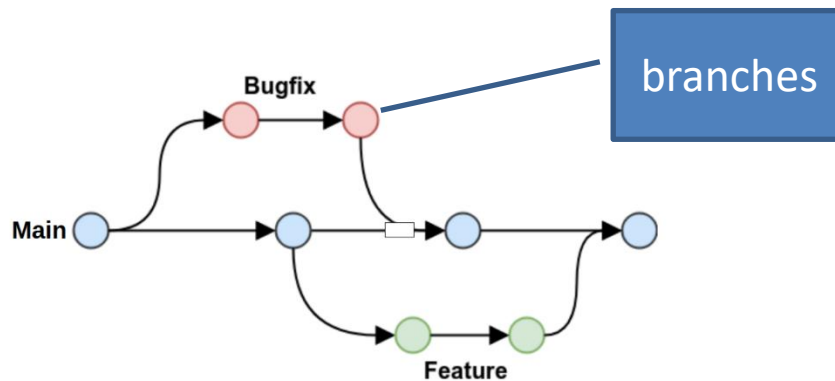
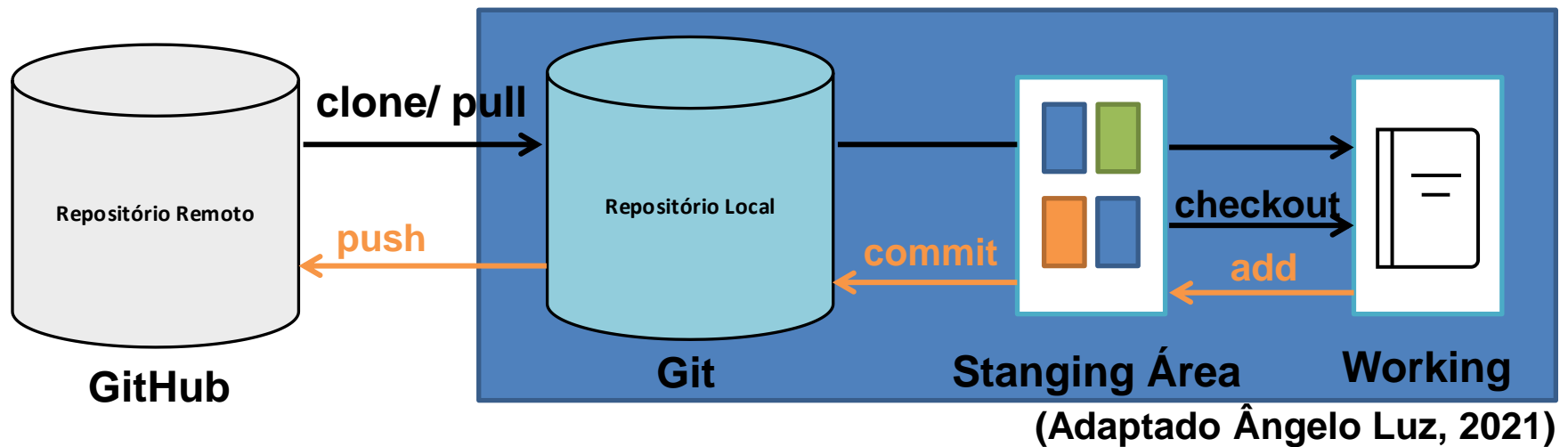
- Vantagens
 - Controle de histórico
 - Permite trabalhar em equipe
 - Ramificação do projeto
 - Segurança
 - Organização

GitHub (Remoto)

- Repositório remoto (Público)
- Plataforma social para desenvolvedores
- Comunidade de desenvolvimento
- Repositórios ilimitados
- Hospedagem de código-fonte
- Características de Rede Social
- Forks de Projetos (respeitando as licenças)

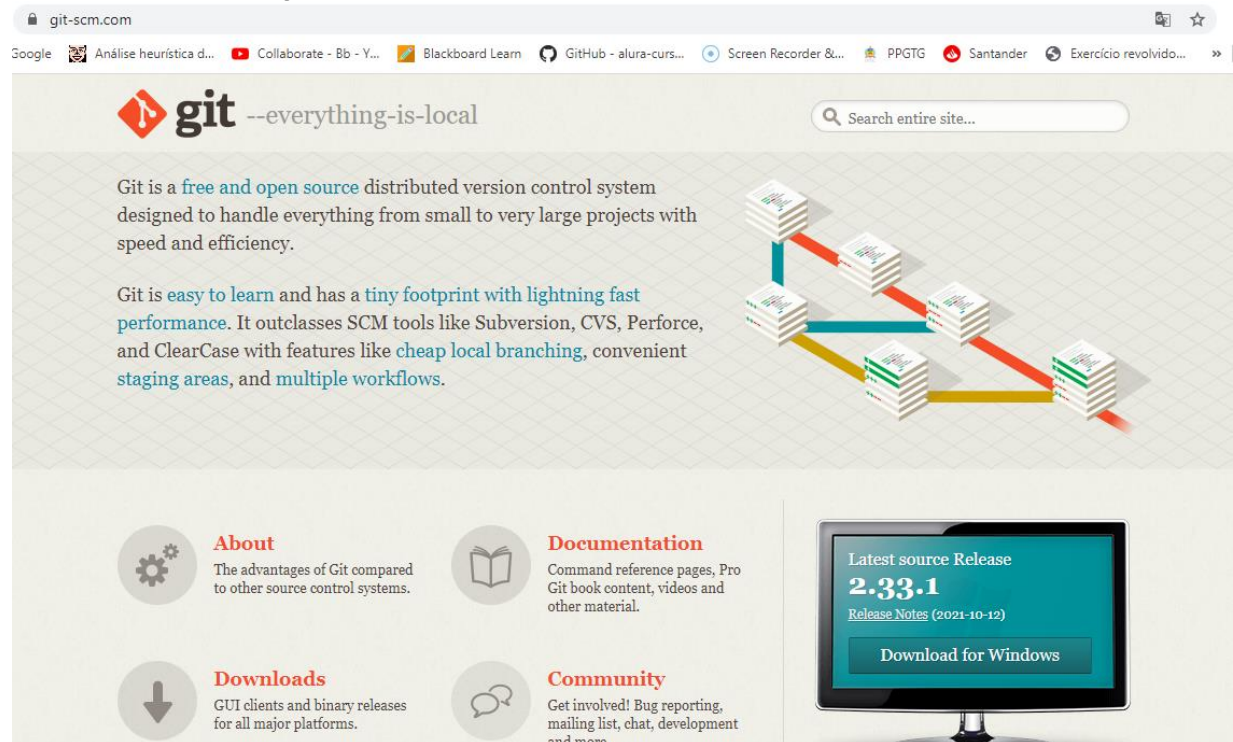


Fluxo do Git



Instalando o Git

- <https://git-scm.com/>

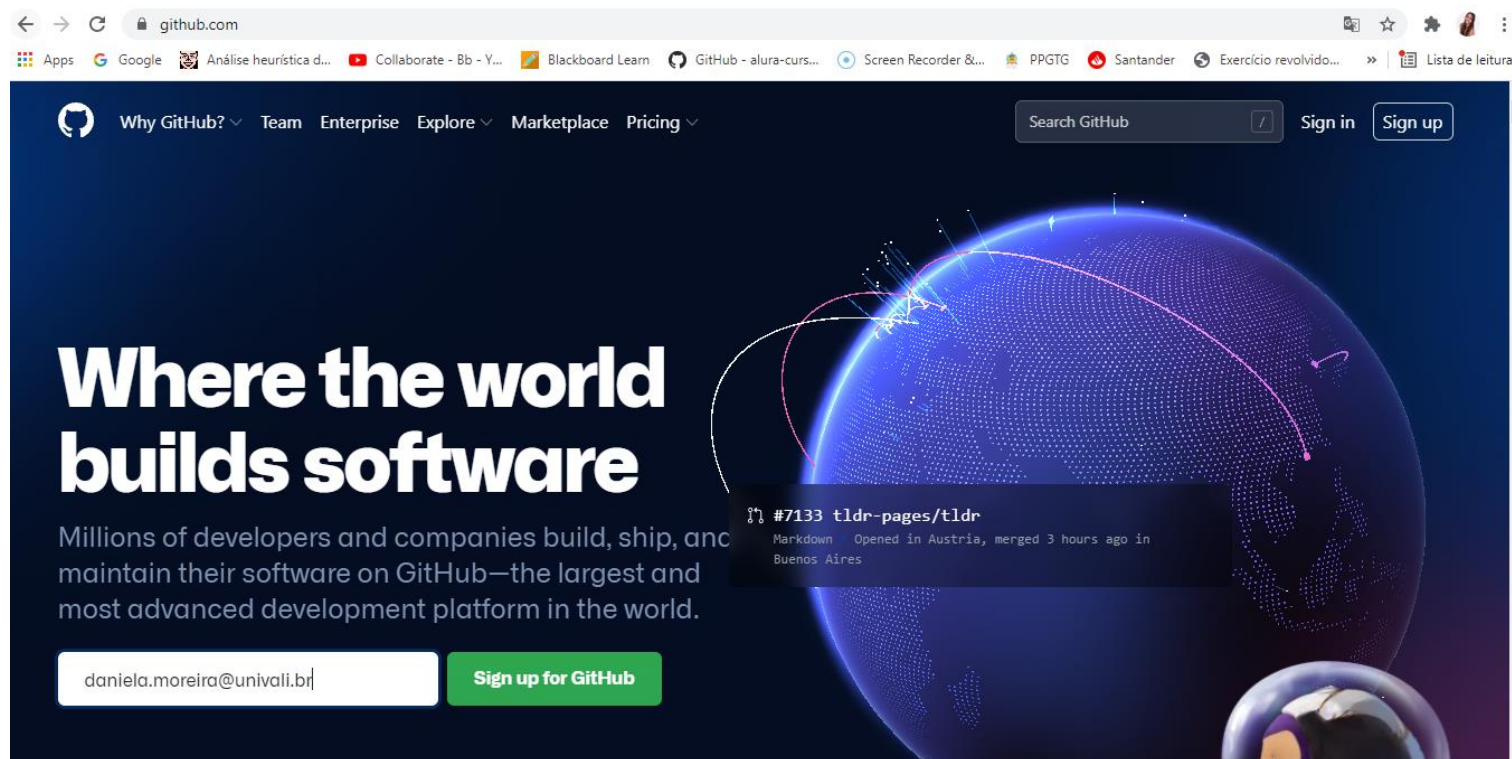


Criando a conta no GitHub (Repositório Remoto)



CIÊNCIA DA
COMPUTAÇÃO
GRANDE FLORIANÓPOLIS
UNIVALI

- <https://github.com/>



Criando a conta no GitHub (Repositório Remoto)



CIÊNCIA DA
COMPUTAÇÃO
GRANDE FLORIANÓPOLIS
UNIVALI

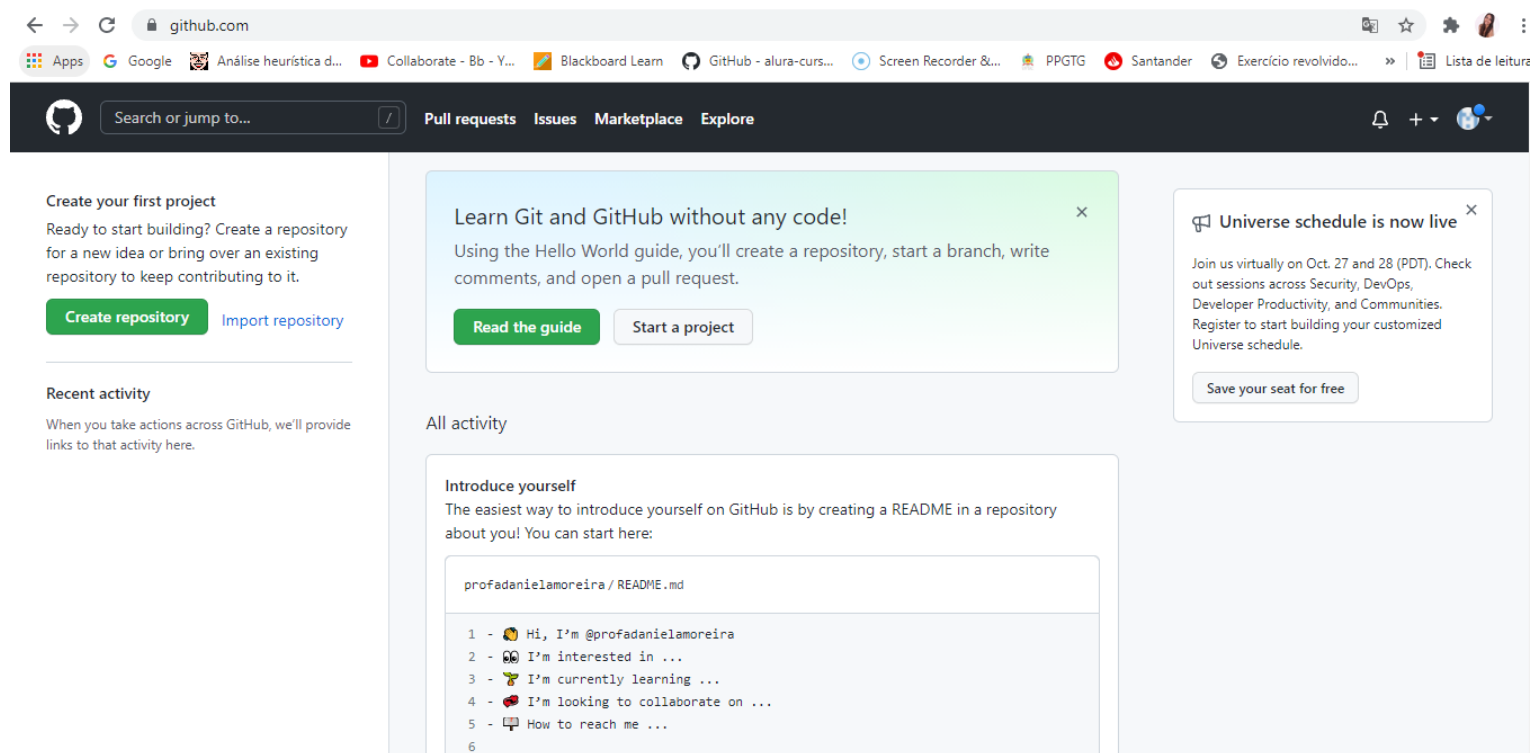
- Definir uma senha e username

The screenshot shows the GitHub signup page in a web browser. The address bar displays the URL: `github.com/signup?user_email=daniela.moreira%40univali.br&source=form-home-signup`. The browser's tab bar shows several open tabs, including 'Apps', 'Google', 'Análise heurística d...', 'Collaborate - Bb - Y...', 'Blackboard Learn', 'GitHub - alura-curs...', 'Screen Recorder &...', 'PPGTG', 'Santander', 'Exercício revolid...', and 'Lista de leitura'. The GitHub page has a dark blue background. In the top right corner, it says 'Already have an account? Sign in →'. The main content area is a light gray box with the following text: 'Welcome to GitHub! Let's begin the adventure'. Below this, there are three sections: 'Enter your email' with a green checkmark and the email 'daniela.moreira@univali.br'; 'Create a password' with a green checkmark and a masked password '.....'; and 'Enter a username' with a red checkmark and the username 'profadanielamoreira'. A green 'Continue' button is located to the right of the username field. At the bottom of the box, it says 'profadanielamoreira is available.'

Criando a conta no GitHub (Repositório Remoto)

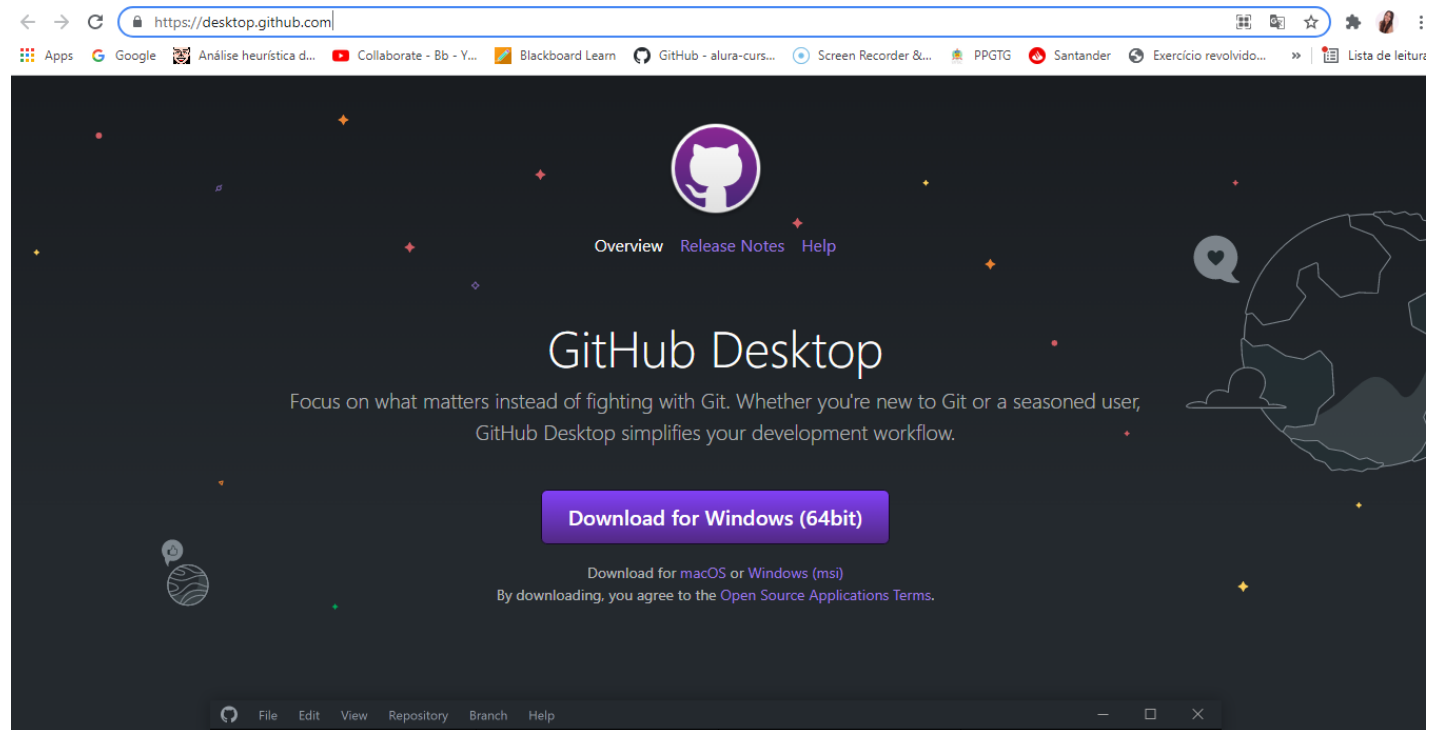


- Tela inicial



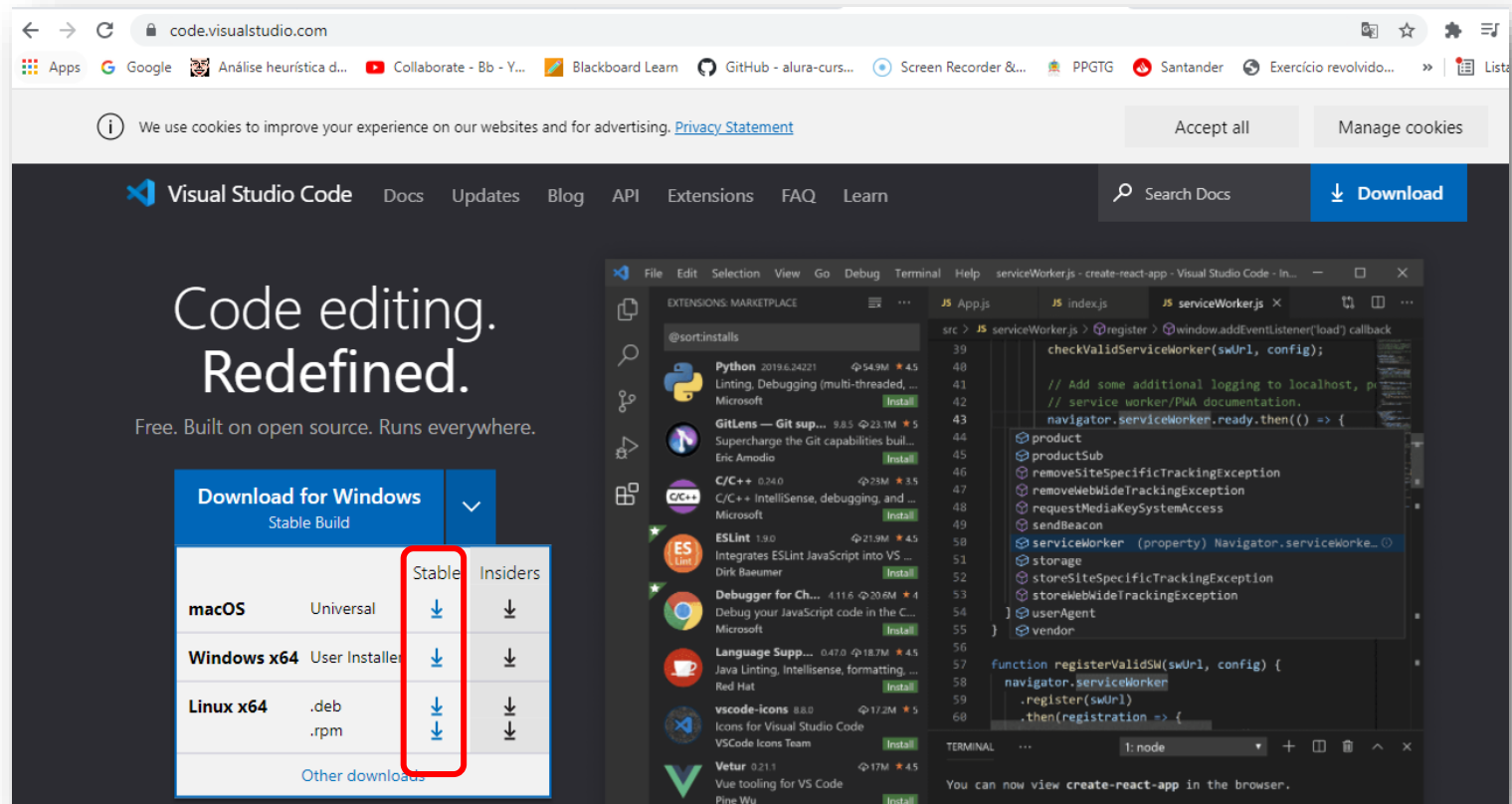
Instalando o GitHub Desktop

- <https://desktop.github.com/>



Instalando o Visual Studio

- <https://code.visualstudio.com/>



The screenshot shows the Visual Studio Code website with the headline "Code editing. Redefined." and a download table. A red box highlights the "Stable" column in the table. To the right, a preview of the Visual Studio Code IDE is shown with the Extensions Marketplace and a code editor.

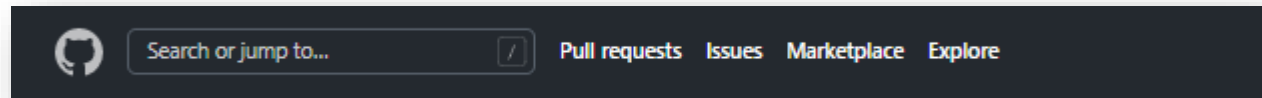
Download for Windows		Stable	Insiders
macOS	Universal	↓	↓
Windows x64	User Installer	↓	↓
Linux x64	.deb .rpm	↓	↓



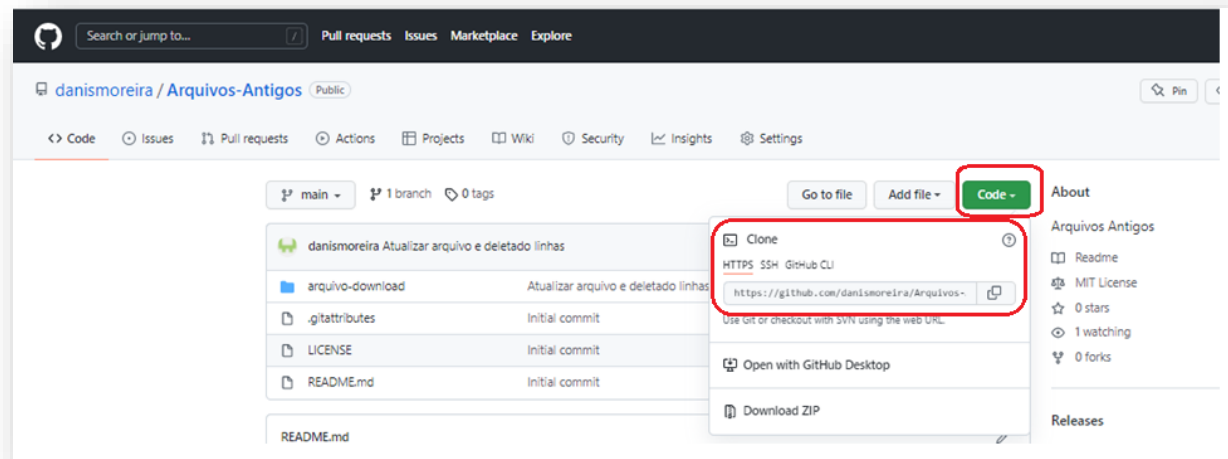
GitHub

Clonar Repositório

- Buscar o repositório desejado por meio do campo de busca do GitHub

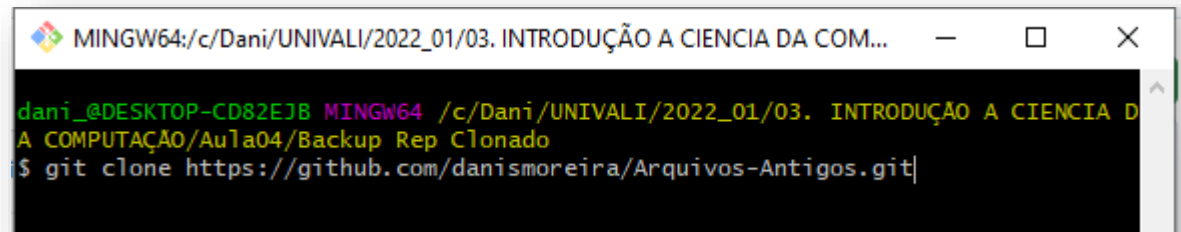


- Copiar o nome do endereço do repositório



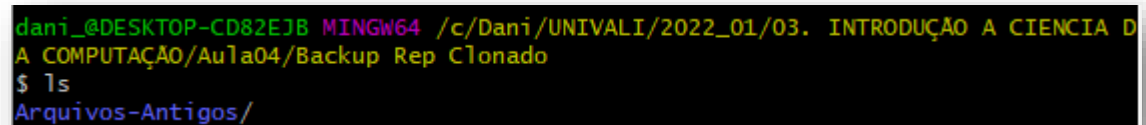
Clonar Repositório

- Criar uma pasta na sua máquina onde os arquivos serão salvos abrir o Git Bash:
 - git clone https://github.com/danismoreirasilva/algoritmos_lista1



```
MINGW64:/c/Dani/UNIVALI/2022_01/03. INTRODUÇÃO A CIENCIA DA COM...
dani_@DESKTOP-CD82EJB MINGW64 /c/Dani/UNIVALI/2022_01/03. INTRODUÇÃO A CIENCIA D
A COMPUTAÇÃO/Aula04/Backup Rep Clonado
$ git clone https://github.com/danismoreira/Arquivos-Antigos.git
```

- Ao dar o comando “ls” você observará que foi criada uma pasta com os dados do repositório remoto.



```
dani_@DESKTOP-CD82EJB MINGW64 /c/Dani/UNIVALI/2022_01/03. INTRODUÇÃO A CIENCIA D
A COMPUTAÇÃO/Aula04/Backup Rep Clonado
$ ls
Arquivos-Antigos/
```

Clonar Repositório

- Os dados atualizados remotamente no repositório de origem não são atualizados de modo automático.
- Deve-se acessar a pasta onde os arquivos foram salvos e dar o comando git pull.
 - Cd nome_da_pasta_criada
 - (cd arquivos-antigos)
 - Git pull
- Clear //comando para limpar a tela do git bash.

```
dani_@DESKTOP-CD82E3B MINGW64 /c/Dani/UNIVALI/2022_01/03. INTRODUÇÃO A CIENCIA D
A COMPUTAÇÃO/Aula04/Backup Rep Clonado
$ cd arquivos-antigos

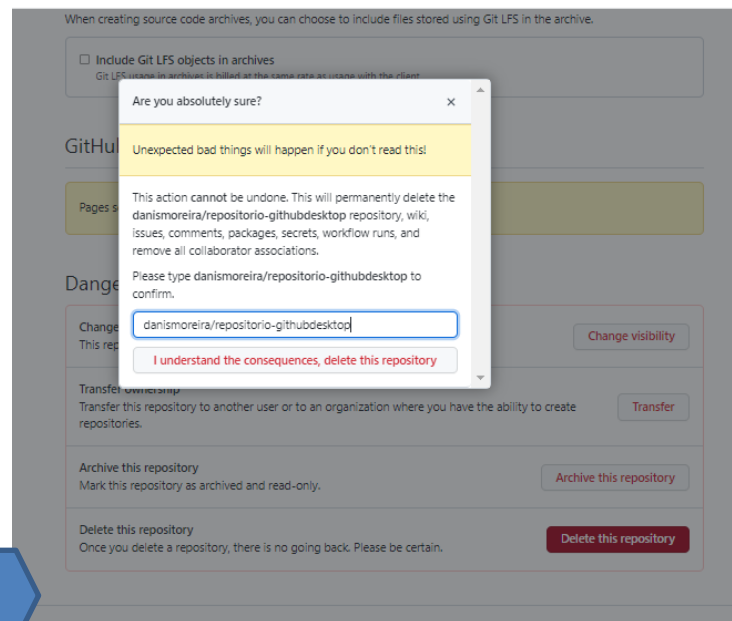
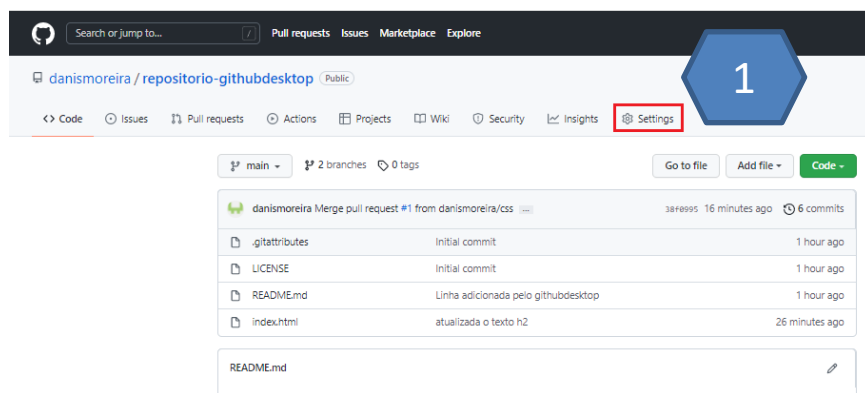
dani_@DESKTOP-CD82E3B MINGW64 /c/Dani/UNIVALI/2022_01/03. INTRODUÇÃO A CIENCIA D
A COMPUTAÇÃO/Aula04/Backup Rep Clonado/arquivos-antigos (main)
$ ls
LICENSE  README.md  arquivo-download/

dani_@DESKTOP-CD82E3B MINGW64 /c/Dani/UNIVALI/2022_01/03. INTRODUÇÃO A CIENCIA D
A COMPUTAÇÃO/Aula04/Backup Rep Clonado/arquivos-antigos (main)
$ git pull
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 814 bytes | 6.00 KiB/s, done.
From https://github.com/danismoreira/Arquivos-Antigos
   7663c62..12c222e  main       -> origin/main
Updating 7663c62..12c222e
Fast-forward
 arquivo-download/novo_repositorio.txt | 1 +
 1 file changed, 1 insertion(+)

dani_@DESKTOP-CD82E3B MINGW64 /c/Dani/UNIVALI/2022_01/03. INTRODUÇÃO A CIENCIA D
A COMPUTAÇÃO/Aula04/Backup Rep Clonado/arquivos-antigos (main)
$ |
```

Deletar Repositório

1. No Github clicar em settings do repositório que será excluído.



2. Baixar a barra de rolagem até o Danger Zone, clicar em “Delete this repository”. Depois inserir o nome do mesmo na área indicada e confirmar a ação.

Vamos praticar

- `git init`
- `git add nome_do_arquivo`
- `git commit -m "first commit"`
- `git remote add origin nome_repositorio_remoto`
- `git branch -m master main` (renomear caso a branch principal esteja como master)
- `git push -u origin main`
- Ps: para add pasta
 - `git add nome_pasta`

Git log

- `git log //log completo`
 - para sair do git bash apos dar o git log, pressionar "q" (quit)
- `git log --oneline //log resumido;`
- `git checkout dca3080//retorna para essa branche`
- `git checkout main //retorna para branch principal`

- Ao criar um repositório o git cria uma branch principal (main)
- Ao desenvolver uma nova feature deve ser criada uma nova branch e somente o que for validado estará na branch principal (main)
 - Git checkout navega entre as branches
 - `git checkout -b feature/teste` (-b cria a feature, caso ela não exista, e já entra nessa branch)

Branch

- `git checkout -b nome_feature`
 - Criou uma branch e copiou todo o conteúdo da main
- `Ls`
 - Lista todos os arquivos da nova branch
- `git branch -d nome_feature`
 - Remove a branch criada
- Criar um novo arquivo no diretório dessa nova branch (xxx.txt)
- `git status` //vai informar que tem um novo arquivo
- `Git add .` //adiciona o novo arquivo
- `Git commit -m "inserindo novo arquivo na branch nova"`
- `Git checkout main` //retorna para branch principal – ao olhar no Windows explorer no diretório, o arquivo novo “sumiu”
- `Git branch` //lista todas as branches
- `git checkout nome_feature` //retorna para branch criada e o arquivo está novamente no diretório.

Merge

- `Git checkout main` //retorna para branch principal
- `Git branch` //lista todas as branches
- `Git merge nome_feature` //junta os dados da branch criada para a branch principal
- `Ls` //lista todos os arquivos

Git show

- Git log --oneline //lista de log dos commits
- git show 44c0b0a //mostra o detalhamento deste código do commit

```
dani_@DESKTOP-CD82EJB MINGW64 /c/Dani/DEEPEN/02. PROJETO LABOR ANALYTICS/Material do Cliente Prolog/ISAC/Processo Separação_Checkout/0. SETUP CD (main)
$ git log --oneline
44c0b0a (HEAD -> main, feature/receita_doce) inserindo novo arquivo na branch nova
6e711a7 add arquivo de receitas
dca3080 atualizada a aba regra de negócio em 1703
229bc96 atualizado o setup equação realizada
051832f atualizado o arquivo com a data da mudança das regras de deposito e etiquetas
603b39c atualizado o cronograma na semana e incluso os farois
810ed40 atualizado o arquivo setup em 07-01-22 com as regras de exclusao
d02df36 incluído o arquivo criaca de pasta no git
cbb223 pasta setup atualizada em 05-01-2022

dani_@DESKTOP-CD82EJB MINGW64 /c/Dani/DEEPEN/02. PROJETO LABOR ANALYTICS/Material do Cliente Prolog/ISAC/Processo Separação_Checkout/0. SETUP CD (main)
$ git show 44c0b0a
commit 44c0b0a7d3c5067a1f06e3c2503bbc39a90e1129 (HEAD -> main, feature/receita_doce)
Author: Daniela S. Moreira da Silva <dani.smoreira@gmail.com>
Date: Thu Mar 17 20:29:41 2022 -0300

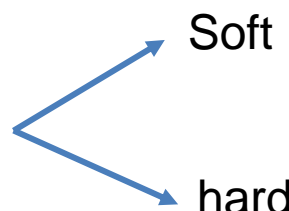
    inserindo novo arquivo na branch nova

diff --git a/arquivo_novo.txt b/arquivo_novo.txt
new file mode 100644
index 0000000..c50bdc6
--- /dev/null
+++ b/arquivo_novo.txt
@@ -0,0 +1,3 @@
+arquivo gerado na branch nova ksdfhkjszdfhkjshdfkjh
+k11zjsdfhkjshdFkjshf
+1ksdjfskjhfKjs
\ No newline at end of file
```

Git checkout

- `Git checkout nome_arquivo //`descarta as alterações deste arquivo
- `Git status`
- `Git checkout . /`descarta todas alterações

Git reset

- Reset 
 - Soft
 - hard
- `Reset head~1 --soft` (o 1 indica a quantidade de commits que iremos voltar)
- Faz alteração no arquivo
- `Git status`
- `Git log //reset` editado terá sumido
- `Git commit -m "ajuste arquivo"`
- `Git log //`aparecerá o ultimo commit
- `Git reset head~1 --hard //`exclui tudo do ultimo commit

Git rm

- Acessar o local do arquivo desejado
- `Git rm nome_Arquivo`
- `Git status`
- `Git commit -m "confirma remocao do arquivo"`
- `Git status`
- `ls`

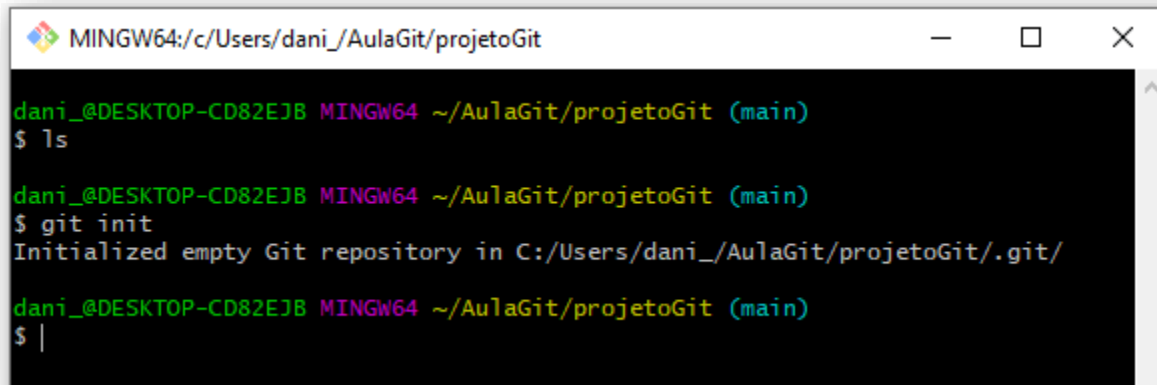
- Criar uma pasta na área de trabalho (o nome da pasta não deve conter espaço)
- Abrir o Git Bash
- Dar o comandos:
 - `pwd` (ver o diretório atual. Ex: `c:\users\2061210`)
 - `ls` (lista todos os arquivos do diretório)
 - `cd desktop` (vai entrar na pasta desktop)
 - `cd nome_pasta` (vai entrar na pasta criada)
 - Pode iniciar um repositório ou clonar outro repositório

Atividade Prática

- Criar uma pasta (por linha comando)
- Abrir o Git Bash
 - Para consultar o diretório atual
 - `pwd` ↵
 - Para criar uma pasta (dentro do diretório atual)
 - `mkdir nome_pasta` ↵
 - Para criar uma subpasta (dentro da pasta criada)
 - `mkdir -p ~/nome_pasta/nome_subpasta` ↵
 - Para entrar na pasta criada
 - `cd ~/nome_pasta/nome_subpasta` ↵

Atividade Prática

- Criar uma pasta na área de trabalho (via linha de comando ou interface gráfica do Win)
- Após criar a pasta, clicar com o botão direito a abrir o Git Bash
- Dentro desta pasta dar o comando:
 - `git --version` /*Consulta versão git instalada
 - `git init` /*cria o repositório vazio



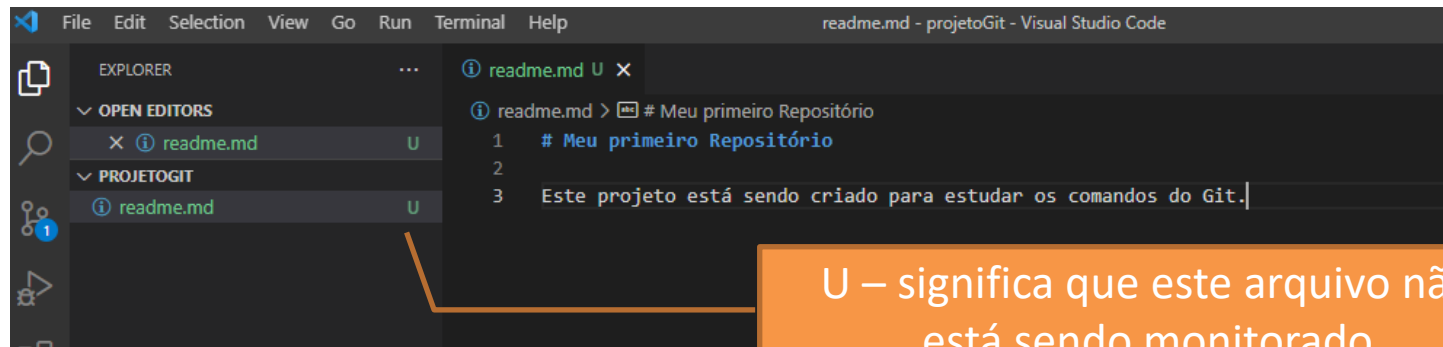
```
MINGW64:/c:/Users/dani_/AulaGit/projetoGit
dani_@DESKTOP-CD82EJB MINGW64 ~/AulaGit/projetoGit (main)
$ ls

dani_@DESKTOP-CD82EJB MINGW64 ~/AulaGit/projetoGit (main)
$ git init
Initialized empty Git repository in C:/Users/dani_/AulaGit/projetoGit/.git/

dani_@DESKTOP-CD82EJB MINGW64 ~/AulaGit/projetoGit (main)
$ |
```


Atividade Prática

- Após ter criado o repositório (git init), vamos criar um arquivo que será monitorado:
- Abrir o Visual Studio Code, abrir a pasta onde o repositório foi criado e criar um novo arquivo denominado readme.md (md é da extensão Markdown, linguagem de marcação)



- O arquivo nesta condição está na área de Working e para ser enviado para o Staging é necessário adicionar o arquivo e depois dar um commit no arquivo para iniciar o monitoramento do mesmo.

Atividade Prática



- Retornar ao Git Bash e adicionar o arquivo criado:
 - git status  /*identifica que tem um arquivo na condição Untracked.

```
dani_@DESKTOP-CD82EJB MINGW64 ~/AulaGit/projetoGit (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  readme.md

nothing added to commit but untracked files present (use "git add" to track)
```

- git add readme.md 
- git status 

```
dani_@DESKTOP-CD82EJB MINGW64 ~/AulaGit/projetoGit (main)
$ git add readme.md

dani_@DESKTOP-CD82EJB MINGW64 ~/AulaGit/projetoGit (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   readme.md
```

Adicionado o arquivo "readme.md" e consultado o status no git em seguida

Atividade Prática

- Dar o comando commit – m “descrição da mensagem”
 - git commit –m “Criado o arquivo readme.md”

```
dani_@DESKTOP-CD82E3B MINGW64 ~/AulaGit/projetoGit (main)
$ git commit -m "Criado o arquivo readme.md"
[main (root-commit) 37b0b6e] Criado o arquivo readme.md
1 file changed, 3 insertions(+)
create mode 100644 readme.md

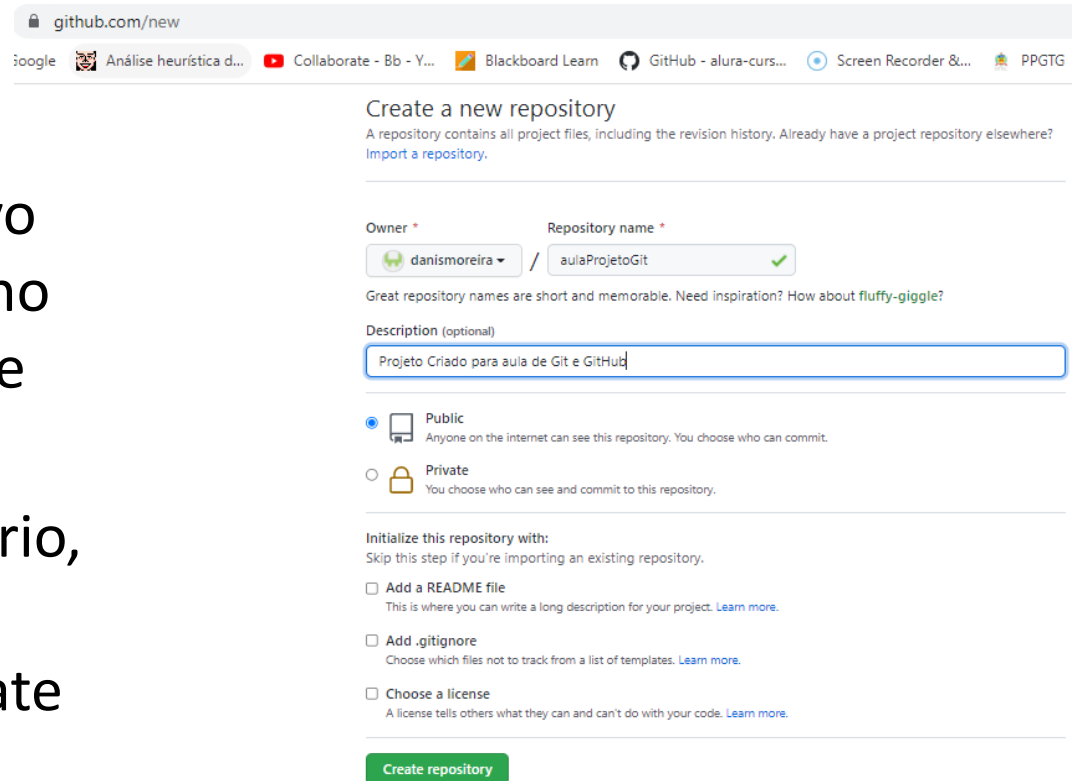
dani_@DESKTOP-CD82E3B MINGW64 ~/AulaGit/projetoGit (main)
$ git status
On branch main
nothing to commit, working tree clean
```

Dar o comando de commit com a mensagem (-m) informando o que foi feito neste comando

- O próximo passo será repositório remoto no GitHub
- Acessar o GitHub com o seu usuário

Atividade Prática

- Criar repositório remoto, neste momento, não serão selecionadas as opções do Readme (porque é o arquivo que está sendo elaborado no Vscode), Gitignore e Choose License.
- Definir o nome do repositório, incluir uma descrição (opcional) e clicar em “Create Repository”).



github.com/new

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * Repository name *

danismoreira / aulaProjetoGit ✓

Great repository names are short and memorable. Need inspiration? How about fluffy-giggle?

Description (optional)

Projeto Criado para aula de Git e GitHub

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

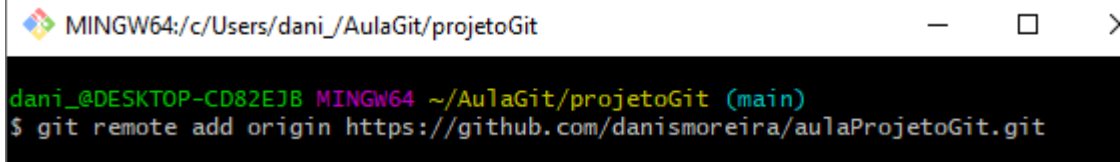
☐ Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Atividade Prática

- Para vincular o repositório remoto criado no GitHub com o seu repositório local, deve-se utilizar o comando `git remote origin add nome_do_repositório`. Esta ação será feita apenas 1 vez em seu projeto.

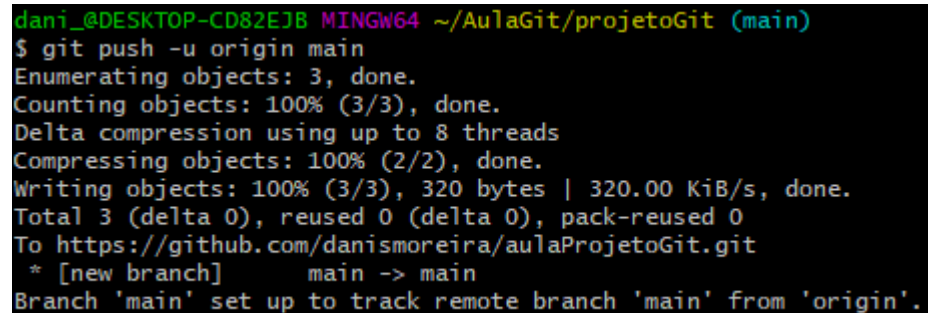
- `git remote add origin https://github.com/danismoreira/aulaProjetoGit.git` ←



```
MINGW64:/c/Users/dani_/AulaGit/projetoGit  
dani_@DESKTOP-CD82E3B MINGW64 ~/AulaGit/projetoGit (main)  
$ git remote add origin https://github.com/danismoreira/aulaProjetoGit.git
```

- Para enviar o arquivo do repositório local (git) para o repositório remoto (GitHub)

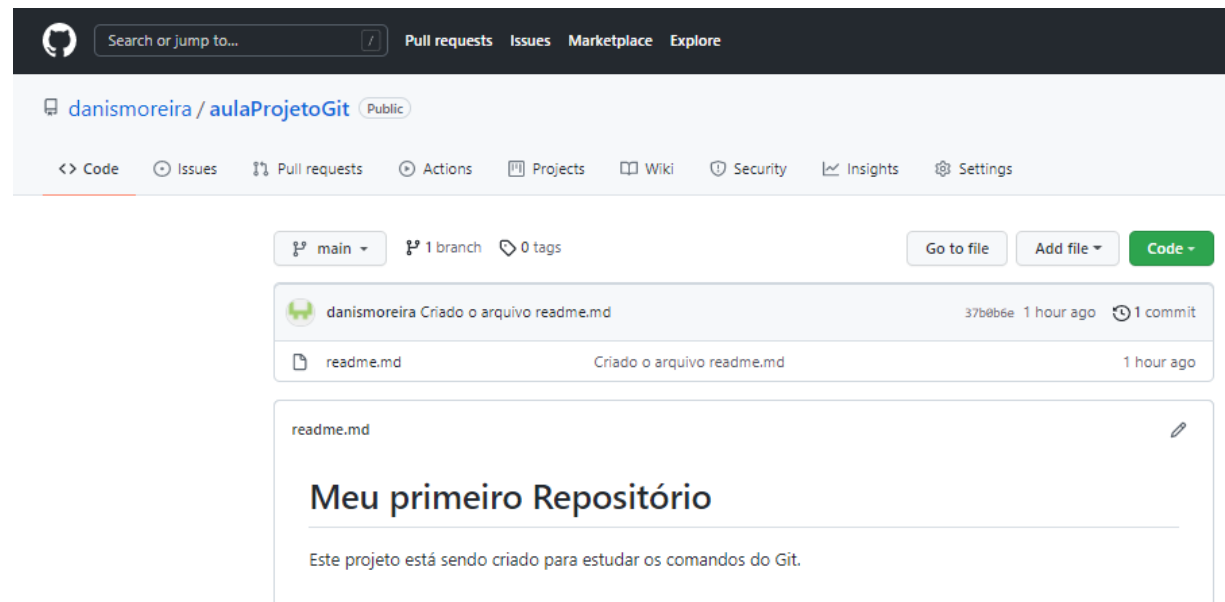
- `git push -u origin main` ←



```
dani_@DESKTOP-CD82E3B MINGW64 ~/AulaGit/projetoGit (main)  
$ git push -u origin main  
Enumerating objects: 3, done.  
Counting objects: 100% (3/3), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (2/2), done.  
Writing objects: 100% (3/3), 320 bytes | 320.00 KiB/s, done.  
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
To https://github.com/danismoreira/aulaProjetoGit.git  
* [new branch]      main -> main  
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

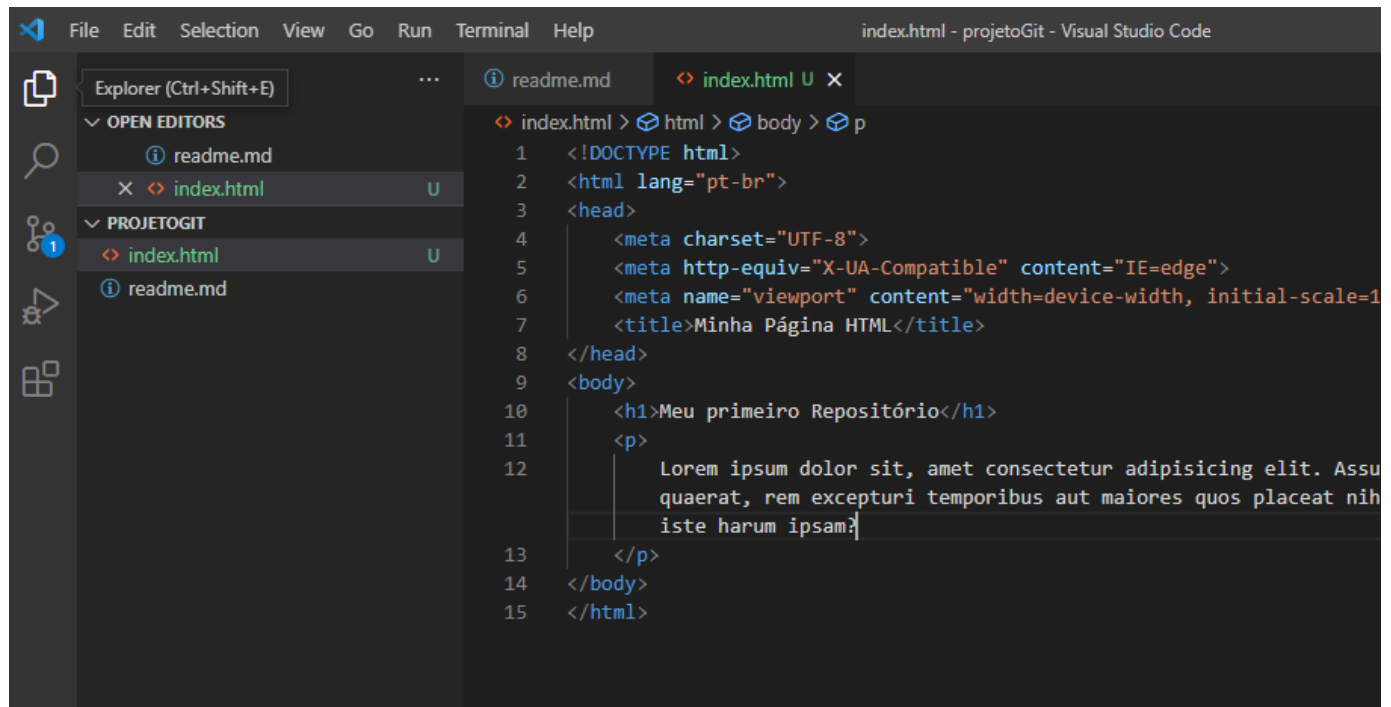
Atividade Prática

- Após enviar os dados do repositório local para o remoto o resultado será similar ao apresentado abaixo.



Atividade Prática

- Versionamento
 - Realizar novas alterações no VSCode e incluir um arquivo index.html



```
index.html - projetoGit - Visual Studio Code

File Edit Selection View Go Run Terminal Help

Explorer (Ctrl+Shift+E)
OPEN EDITORS
  readme.md
  index.html U
PROJETOGIT
  index.html U
  readme.md

index.html > html > body > p
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>Minha Página HTML</title>
8 </head>
9 <body>
10   <h1>Meu primeiro Repositório</h1>
11   <p>
12     Lorem ipsum dolor sit, amet consectetur adipisicing elit. Assu
13     quaerat, rem excepturi temporibus aut maiores quos placeat nih
14     iste harum ipsam?
15   </p>
16 </body>
17 </html>
```

Atividade Prática

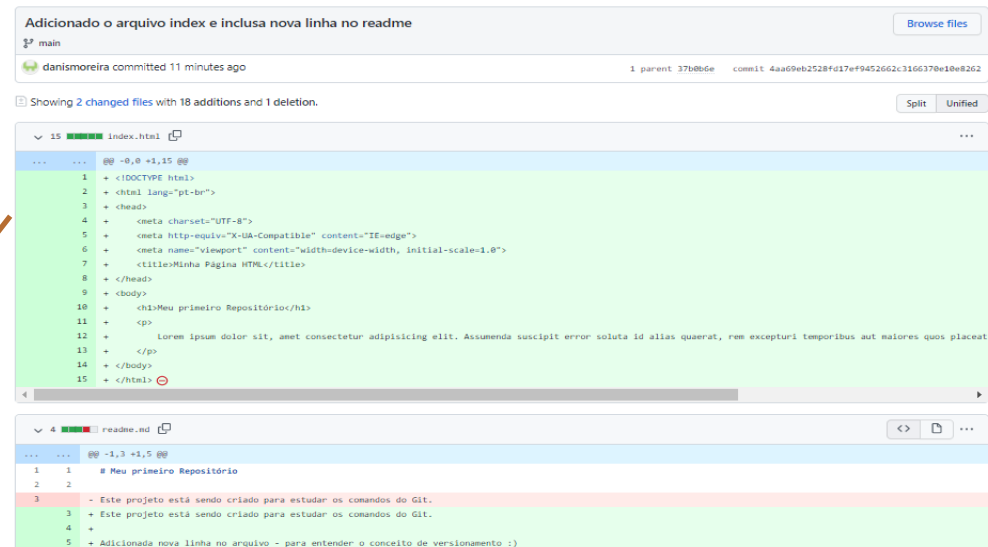
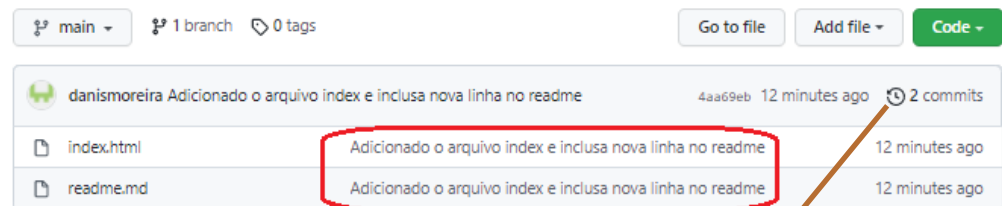
- Após realizar as alterações (incluir index.html e incluir linha no arquivo de readme), dar os comandos adicionar os arquivos na área de *staging* e em seguida o *commit*
 - `git add .` ← /*incluir todas as alterações
 - `git commit -m "git commit -m "Adicionado o arquivo index e inclusa nova linha no readme"` ←
 - `git status` ← /*visualiza o status das alterações (neste caso, já “comitadas”)
- Para enviar ao repositório remoto as alterações realizadas
 - `git push origin main`

Atividade Prática

- Para consultar as mudanças que foram realizadas, pode-se clicar mensagem do commit no repositório remoto

Agora temos 2 commits

O resultado apresentará as mudanças que foram implementadas em verde (adicionado), e em vermelho como era no arquivo original, em relação ao último commit



Atividade Prática

- Branches
 - Criar uma branch no git antes de desenvolver a nova feature:
 - `git checkout -b "novo-paragrafo"` ↩ /* ao dar este comando, estamos saindo a branch principal (main) e estamos indo para branch criada.

```
MINGW64:/c/Users/dani_/AulaGit/projetoGit

dani_@DESKTOP-CD82EJB MINGW64 ~/AulaGit/projetoGit (main)
$ git checkout -b "novo-paragrafo"
Switched to a new branch 'novo-paragrafo'

dani_@DESKTOP-CD82EJB MINGW64 ~/AulaGit/projetoGit (novo-paragrafo)
$ |
```

Agora estamos na nova branch

- `git status` ↩

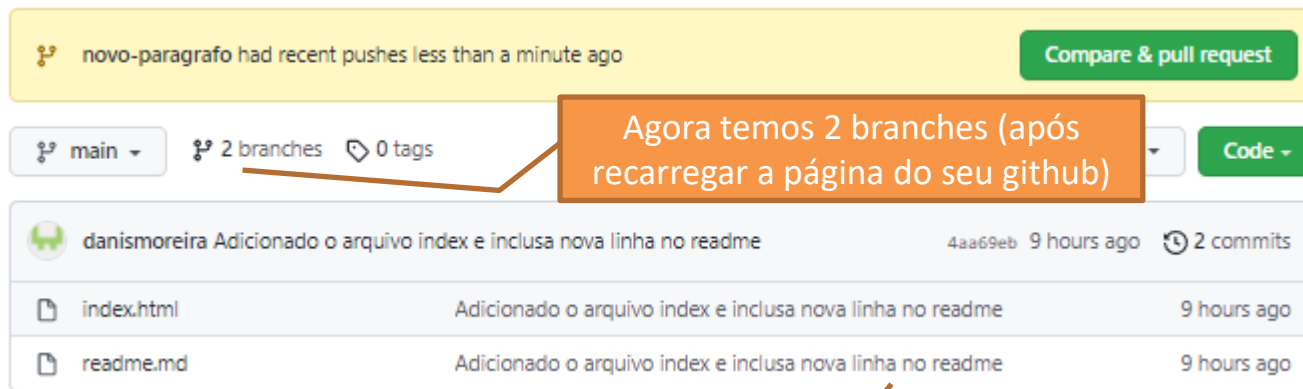
```
dani_@DESKTOP-CD82EJB MINGW64 ~/AulaGit/projetoGit (novo-paragrafo)
$ git status
On branch novo-paragrafo
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   readme.md

no changes added to commit (use "git add" and/or "git commit -a")

dani_@DESKTOP-CD82EJB MINGW64 ~/AulaGit/projetoGit (novo-paragrafo)
$ |
```

Atividade Prática

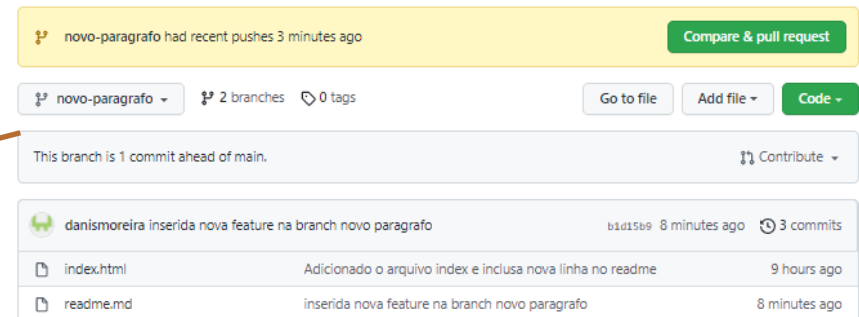
- Branches
 - `git add .`
 - `git commit -m "inserida nova feature na branch novo paragrafo"`
 - `git push origin novo-paragrafo` /*enviando para o repositório remoto a nova branch



Nova feature ainda não está branch principal

Atividade Prática

- Para retornar a branch principal e fazer um merge
 - `git checkout main`
 - `git merge novo-paragrafo` /*dentro da branch principal faz-se um merge com a branch criada
 - `git push origin main` /*enviar as atualizações para o github



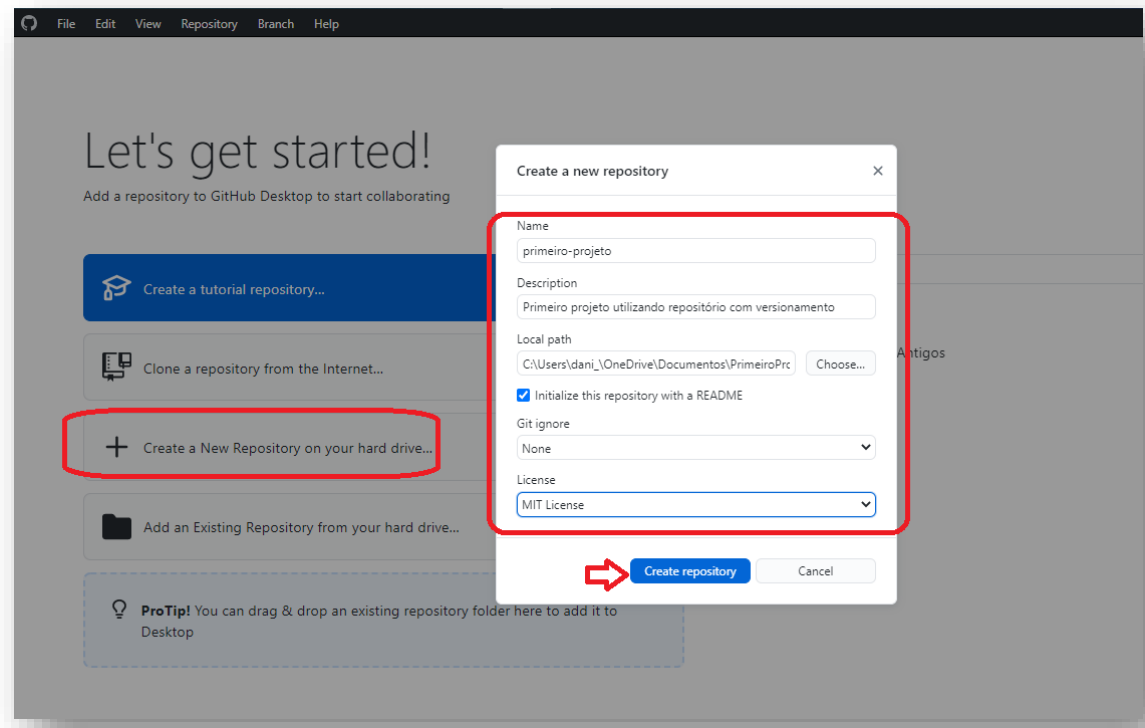
Essa mensagem indica que essa branch está a frente branch principal (antes do merge)



Github Desktop

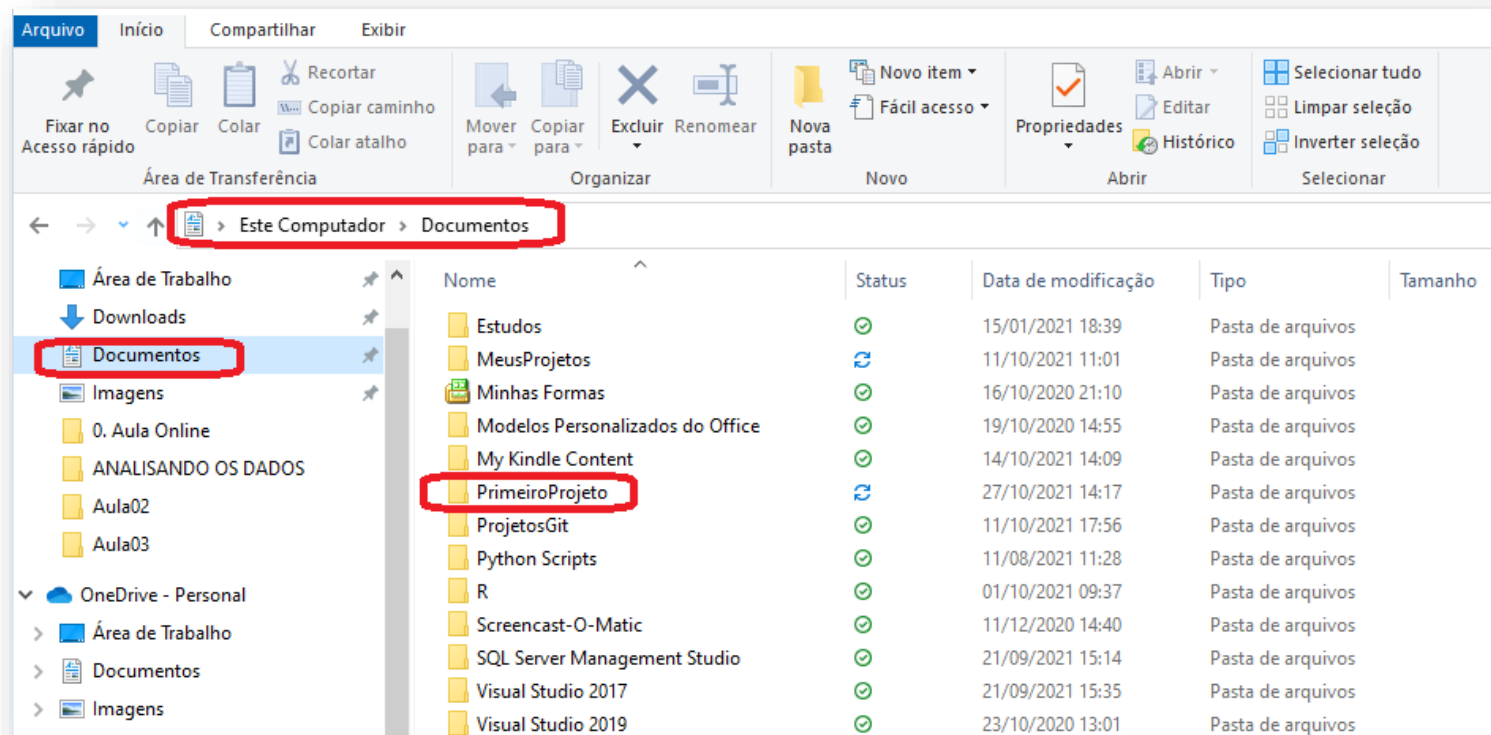
Criando Repositório Local

- Acessar o GitHubDesktop, criar o novo repositório e indicar o caminho da pasta e o nome

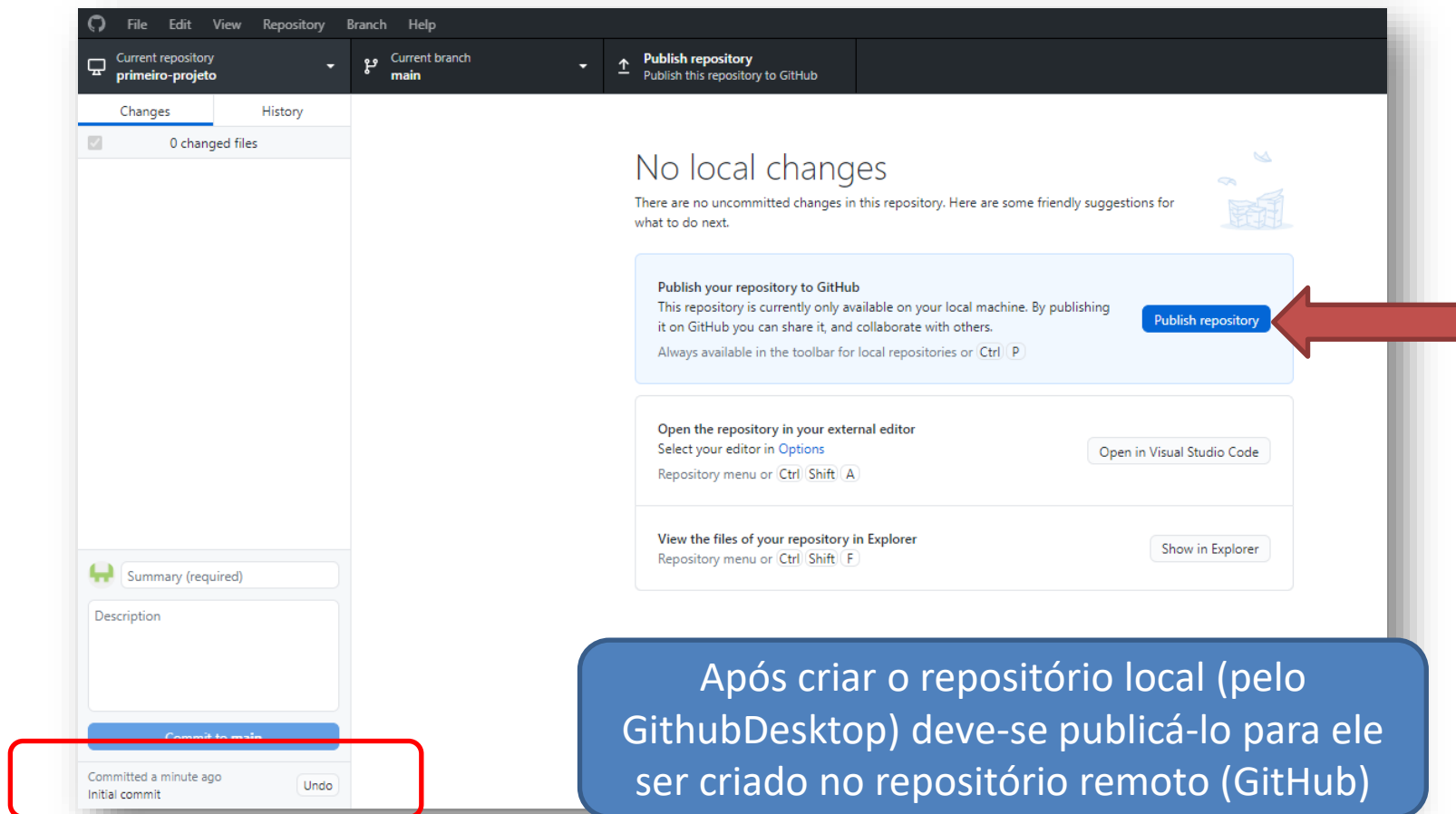


Criando Repositório Local

- Após criar o repositório será criada a pasta para o versionamento.



Criando Repositório Local



File Edit View Repository Branch Help

Current repository
primeiro-projeto

Current branch
main

Publish repository
Publish this repository to GitHub

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Publish your repository to GitHub
This repository is currently only available on your local machine. By publishing it on GitHub you can share it, and collaborate with others.
Always available in the toolbar for local repositories or **Ctrl** | **P**

Publish repository

Open the repository in your external editor
Select your editor in [Options](#)
Repository menu or **Ctrl** | **Shift** | **A**

Open in Visual Studio Code

View the files of your repository in Explorer
Repository menu or **Ctrl** | **Shift** | **F**

Show in Explorer

Summary (required)

Description

Commit to main

Committed a minute ago
Initial commit

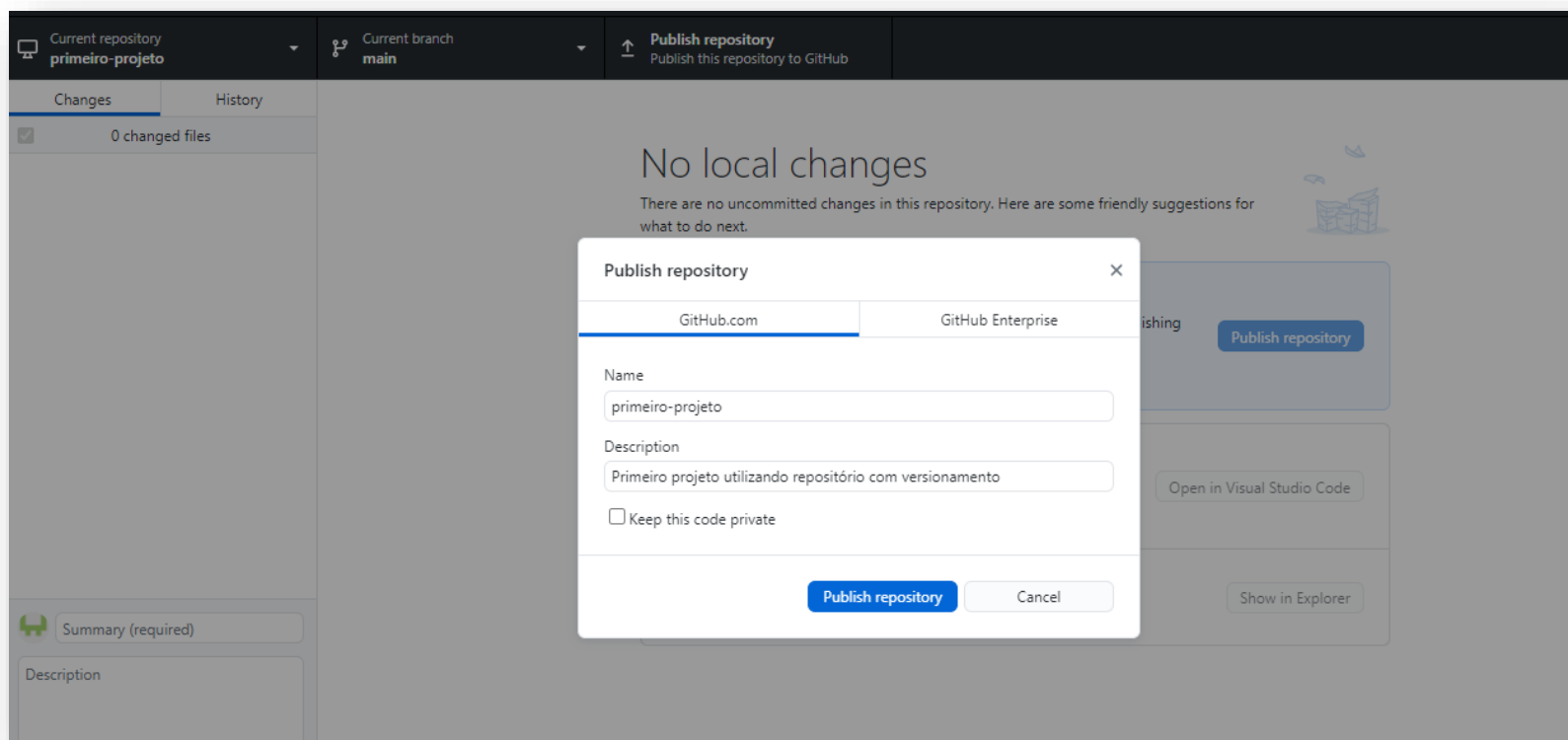
Undo

Após criar o repositório local (pelo GithubDesktop) deve-se publicá-lo para ele ser criado no repositório remoto (GitHub)

Publicando um repositório no GitHub

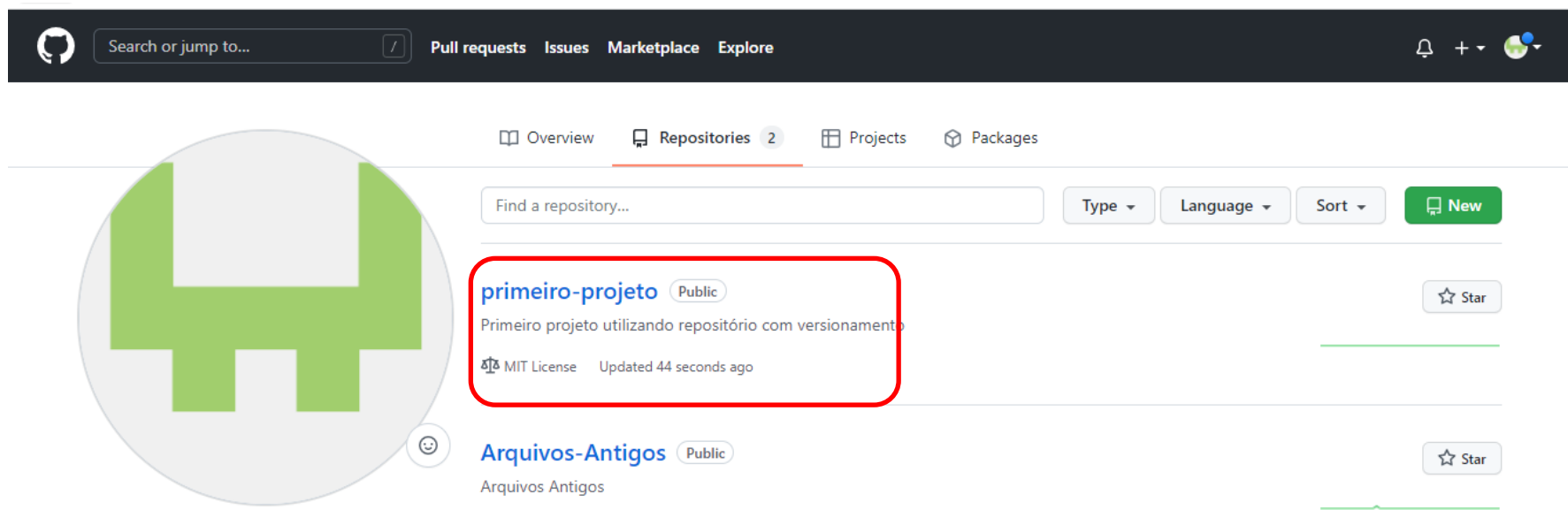


- Desmarcar a opção Keep this code private (para que repositório seja público)

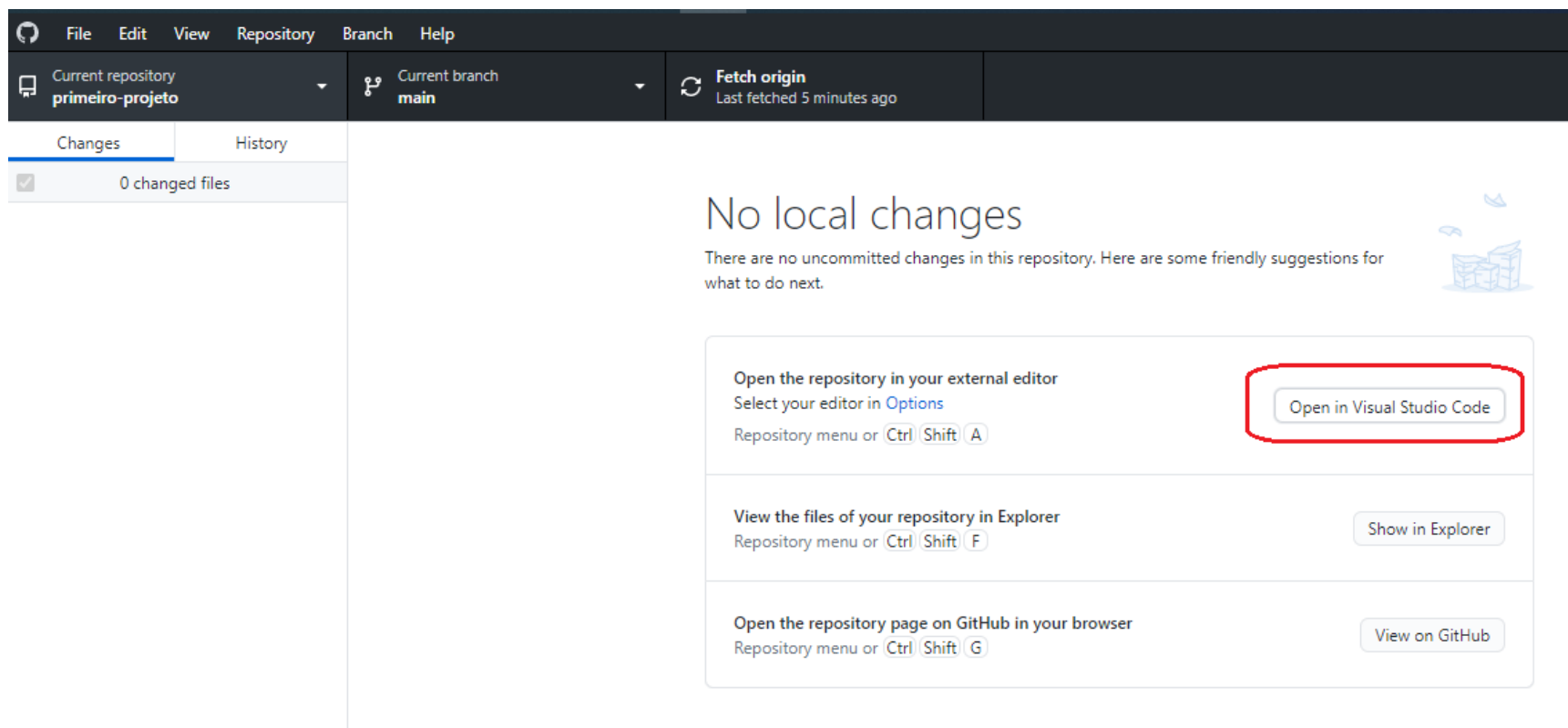


Consultando no GitHub

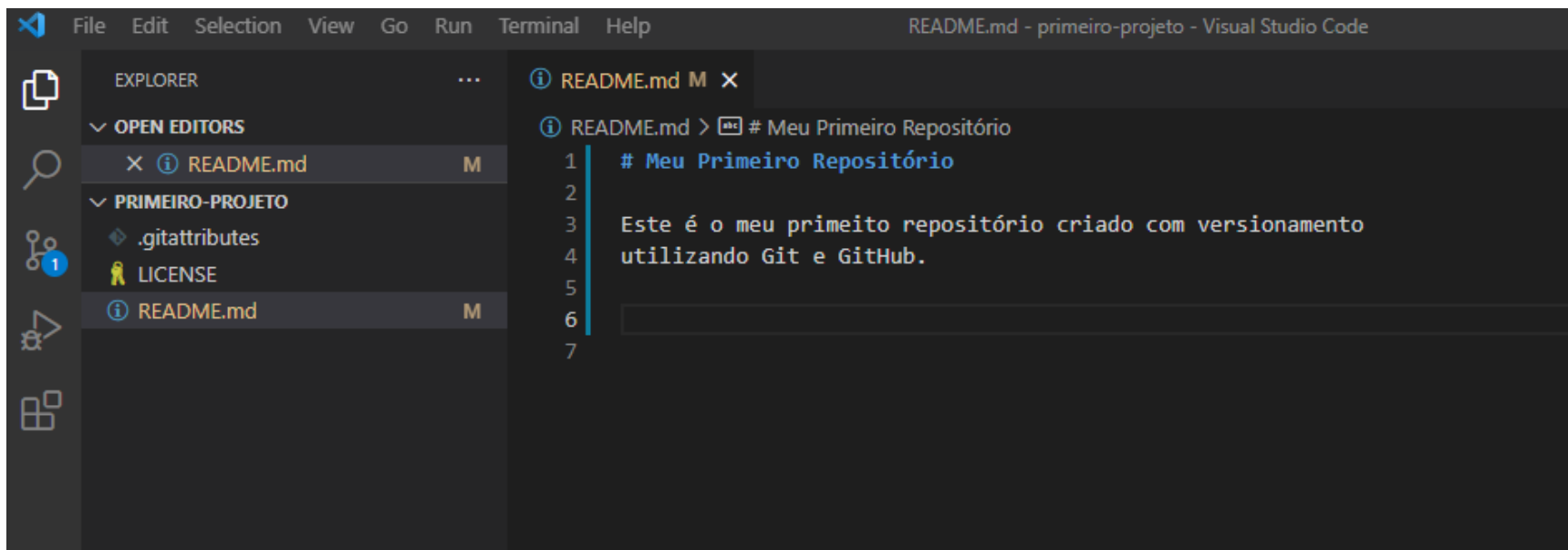
- Verificar seus repositórios



Abrir o Visual Studio Code



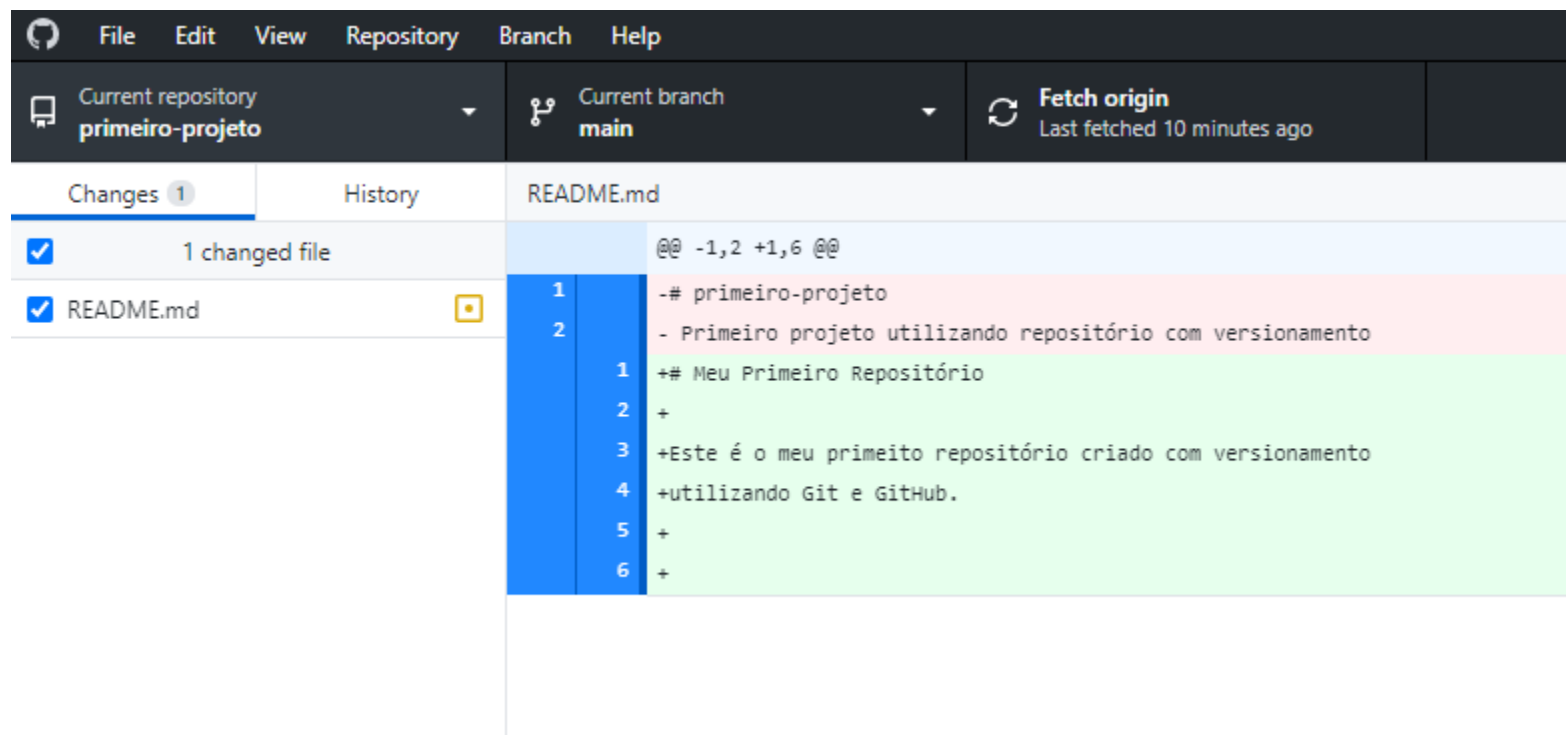
Editar o texto do Readme



The screenshot shows the Visual Studio Code interface with the README.md file open in the editor. The Explorer sidebar on the left shows the project structure with files like .gitattributes, LICENSE, and README.md. The editor window displays the following content:

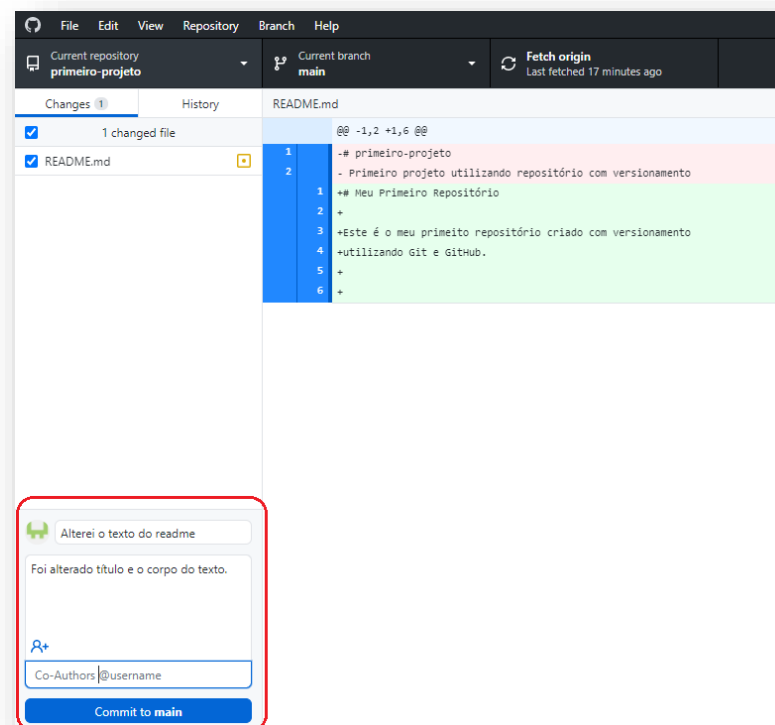
```
1 # Meu Primeiro Repositório
2
3 Este é o meu primeiro repositório criado com versionamento
4 utilizando Git e GitHub.
5
6
7
```


Observar as alterações após a edição



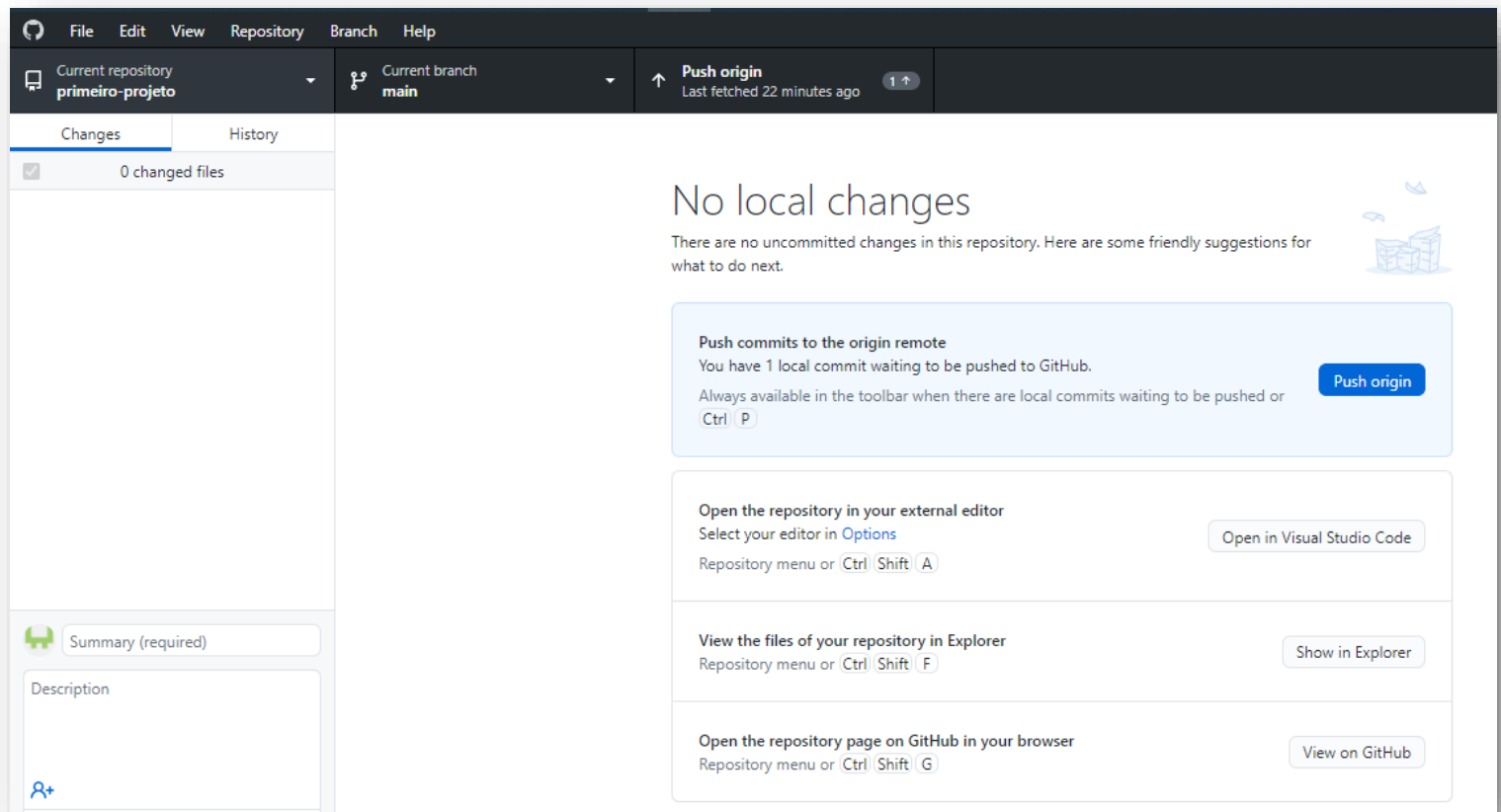
Fazer o Commit para iniciar o versionamento local pelo Git

- Dar um título para o commit e inserir um descrição.



Enviar para o repositório Remoto - Push

- Clicar em Push origin (Enviar/Empurrar para o repositório)



Editando o Readme pelo Github

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file

Add file

Code

1

danismoreira Alterei o texto do readme ... 435697a 5 minutes ago 2 commits

.gitattributes	Initial commit	40 minutes ago
LICENSE	Initial commit	40 minutes ago
README.md	Alterei o texto do readme	5 minutes ago

README.md

primeiro-projeto / README.md in main

Cancel changes

<> Edit file Preview

Spaces 1 Soft wrap

```
1 # Meu Primeiro Repositório
2
3 Este é o meu primeiro repositório criado com versionamento
4 utilizando Git e GitHub.
5
6 Linha adicionada pelo editor do Github (remoto)
7
8
9
```

2



Commit changes

Linha adicionada pelo Github

Add an optional extended description...

3

☒ Commit directly to the main branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes

Cancel

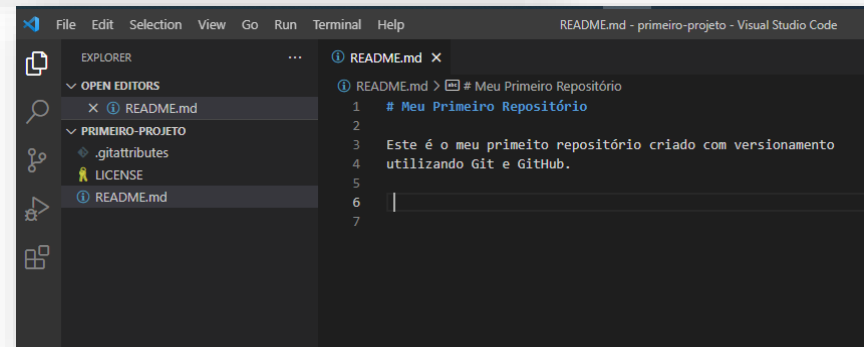
Após mudar no Github a alteração não reflete na máquina local

8 lines (4 sloc) | 167 Bytes

Meu Primeiro Repositório

Este é o meu primeiro repositório criado com versionamento utilizando Git e GitHub.

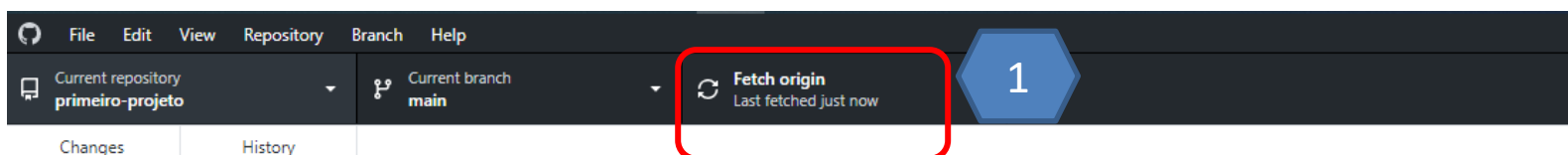
Linha adicionada pelo editor do Github (remoto)



```
File Edit Selection View Go Run Terminal Help
EXPLORER
  OPEN EDITORS
    README.md
  PRIMEIRO-PROJETO
    .gitattributes
    LICENSE
    README.md
README.md x
1 # Meu Primeiro Repositório
2
3 Este é o meu primeiro repositório criado com versionamento
4 utilizando Git e GitHub.
5
6
7
```

Atualizando localmente possíveis alterações remotas

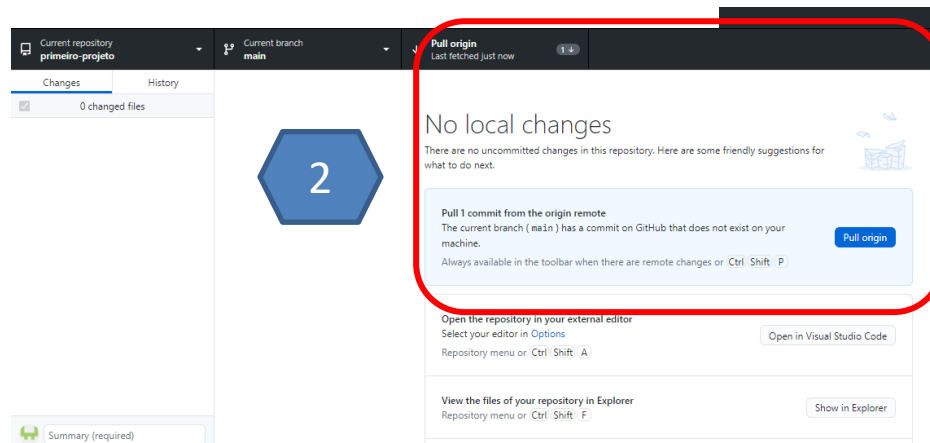
- Para atualizar a máquina local e verificar se houve alguma alteração remota deve-se clicar em “Fetch origin”



No local changes

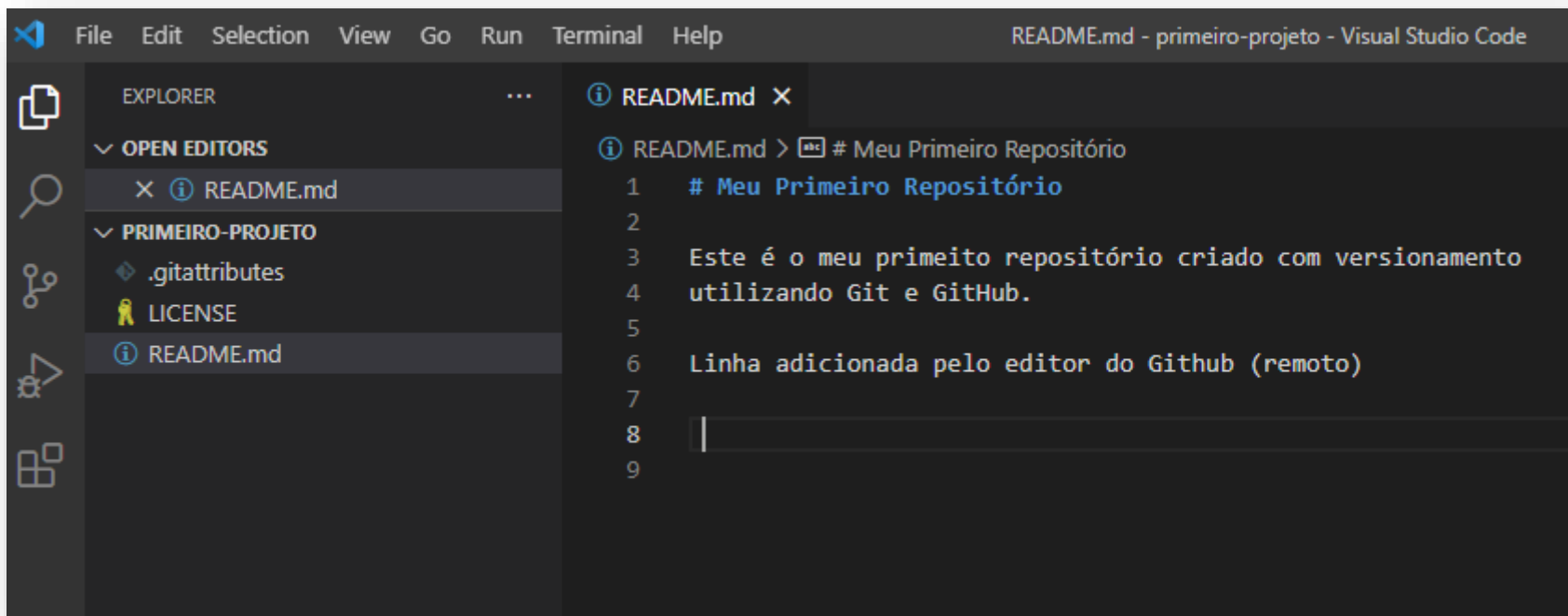
There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Pull – “puxar” as alterações



Verificando o arquivo atualizado no VSCode

- Resultado após o Fetch (arquivo atualizado com a mudança feita no repositório remoto)

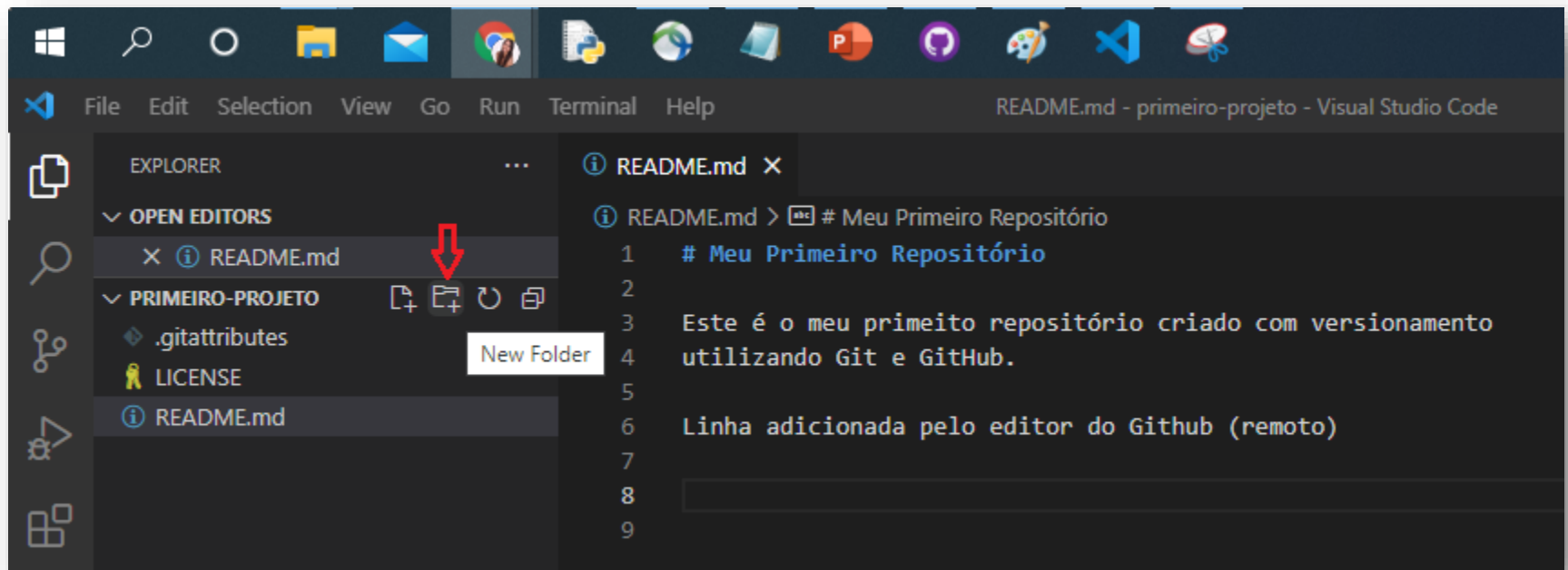


The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer sidebar is expanded, showing the 'PRIMEIRO-PROJETO' folder. Inside this folder, the files '.gitattributes', 'LICENSE', and 'README.md' are listed. The 'README.md' file is selected and its content is displayed in the main editor area. The content of the README.md file is as follows:

```
1 # Meu Primeiro Repositório
2
3 Este é o meu primeiro repositório criado com versionamento
4 utilizando Git e GitHub.
5
6 Linha adicionada pelo editor do Github (remoto)
7
8
9
```

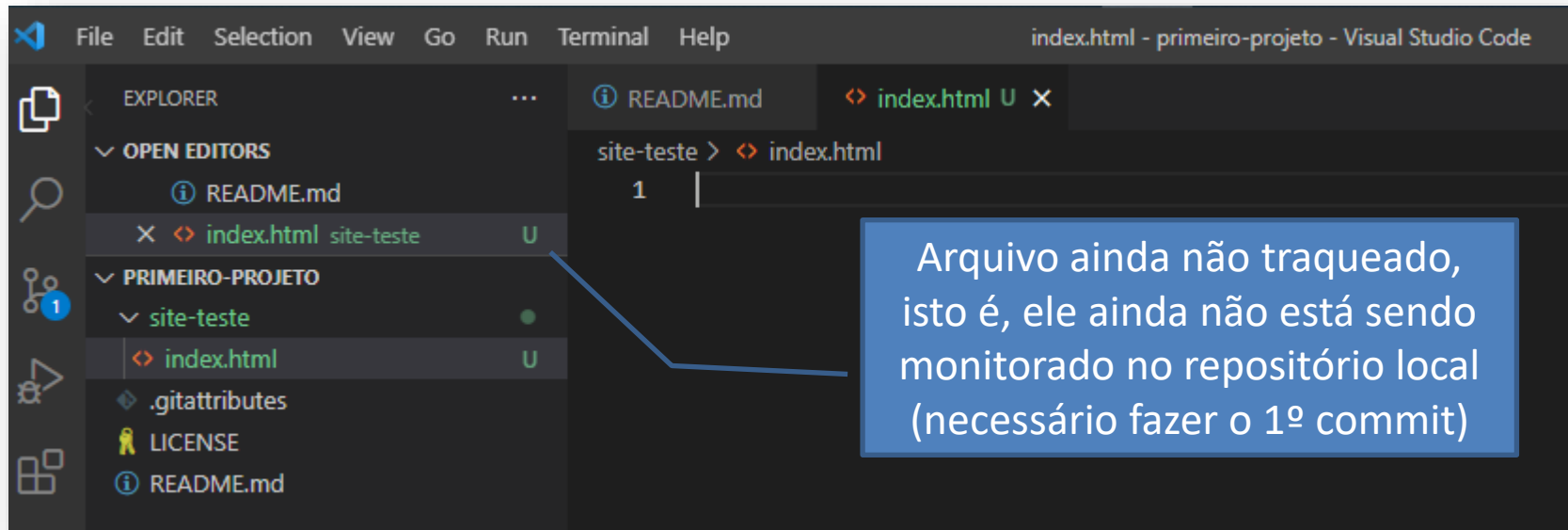
Criar Página HTML de Teste

- No VSCode criar uma pastinha para um site HTML de teste que será criado.



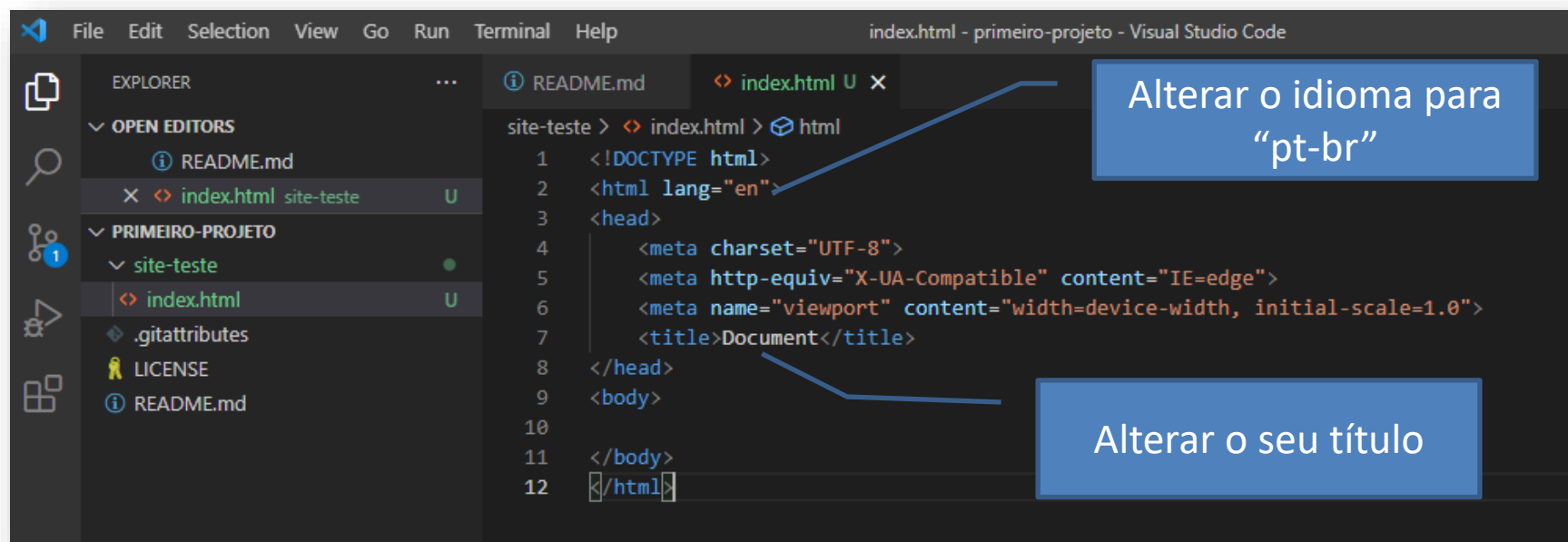
Criar Página HTML de Teste

- Dentro da pasta criada, adicionar um arquivo com o nome “index.html”



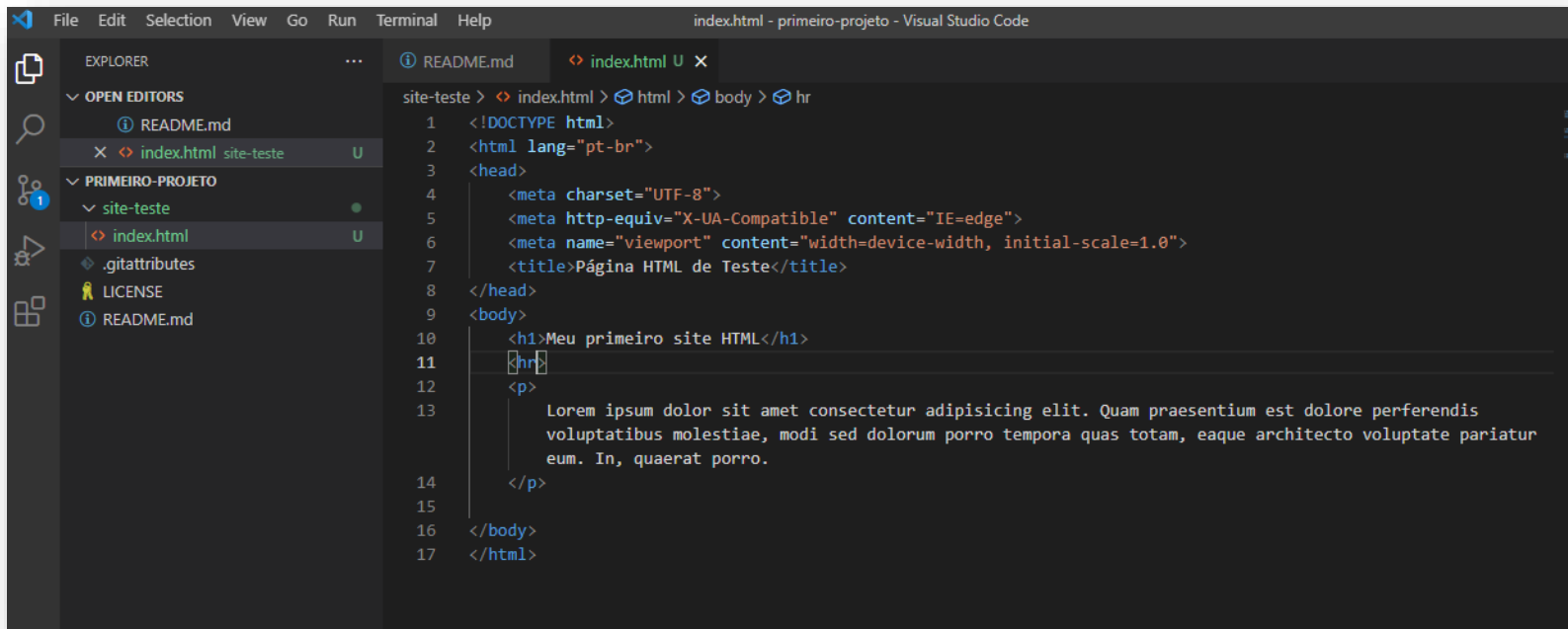
Criar Página HTML de Teste

- Dentro do VSCODE:
- Digite “!” e pressione “Enter”
- O Vscode irá inserir o código básico de um HTML



Criar Página HTML de Teste

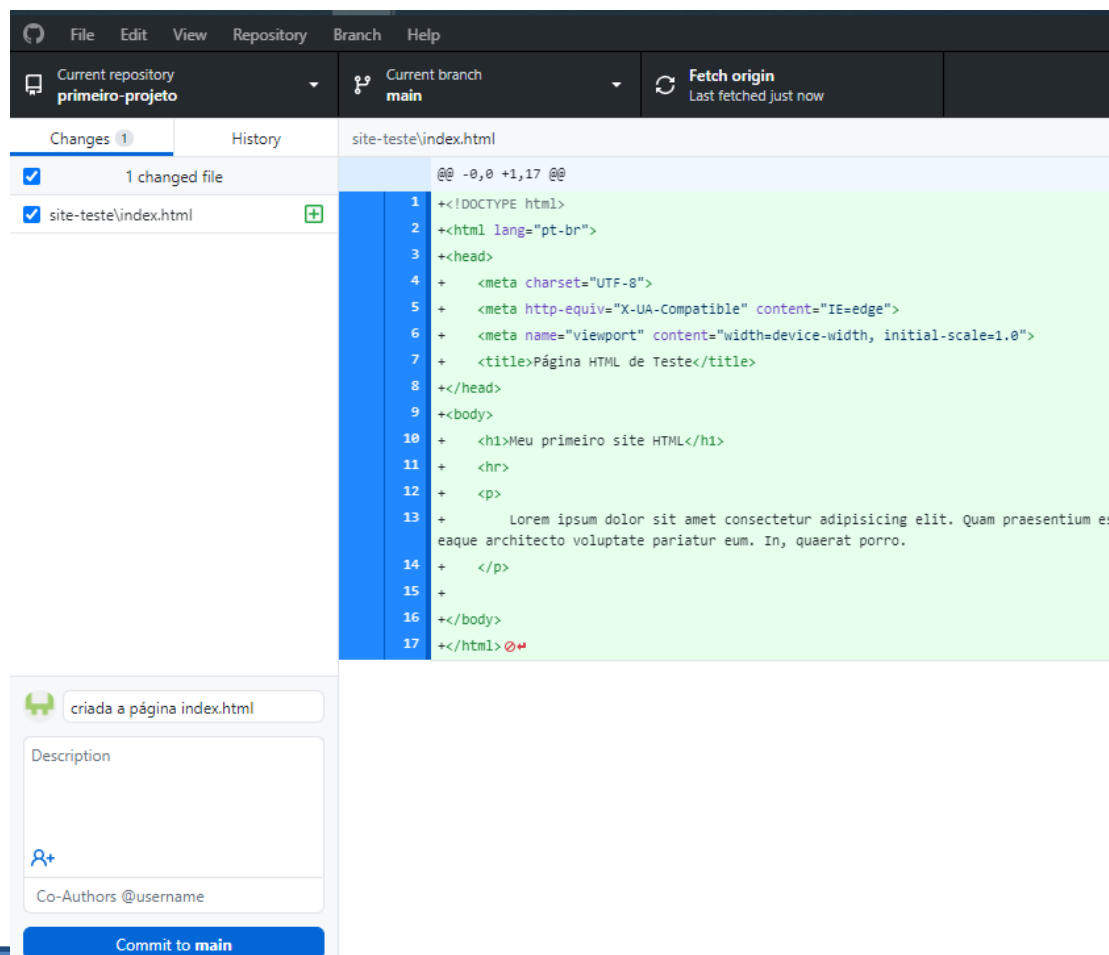
- Dentro das tags “<body></body>” digitar h1 e dar enter
- O Vscod automaticamente criará as tags <h1></h1> para incluir título principal da página.
- Depois digite p e de enter, e o Vscod criará as tags <p></p> para incluir o seu parágrafo
- Dentro destas tags <p></p> digite lorem e de enter



The screenshot shows the Visual Studio Code interface with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'PRIMEIRO-PROJETO' with a subfolder 'site-teste' containing 'index.html'. The code editor shows the content of 'index.html' with the following code:

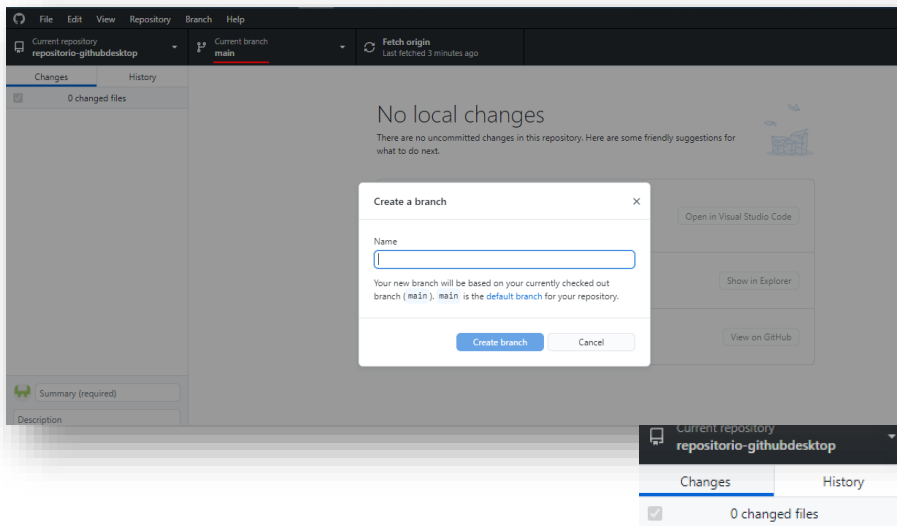
```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Página HTML de Teste</title>
8 </head>
9 <body>
10  <h1>Meu primeiro site HTML</h1>
11  <hr>
12  <p>
13    Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam praesentium est dolore perferendis
14    voluptatibus molestiae, modi sed dolorum porro tempora quas totam, eaque architecto voluptate pariatur
15    eum. In, quaerat porro.
16  </p>
17 </body>
18 </html>
```

Dar commit para enviar ao repositório local

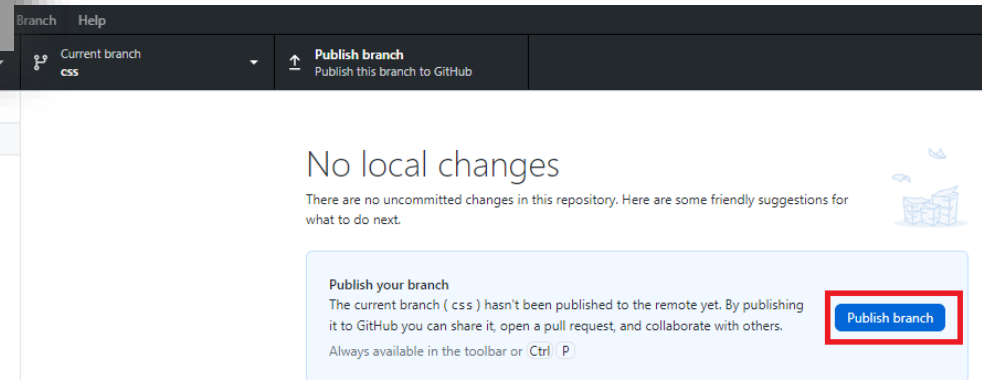


Criar nova Branch

- Ir em *Current branch*
- Informar o nome da nova *branch* e clicar em *Create branch*

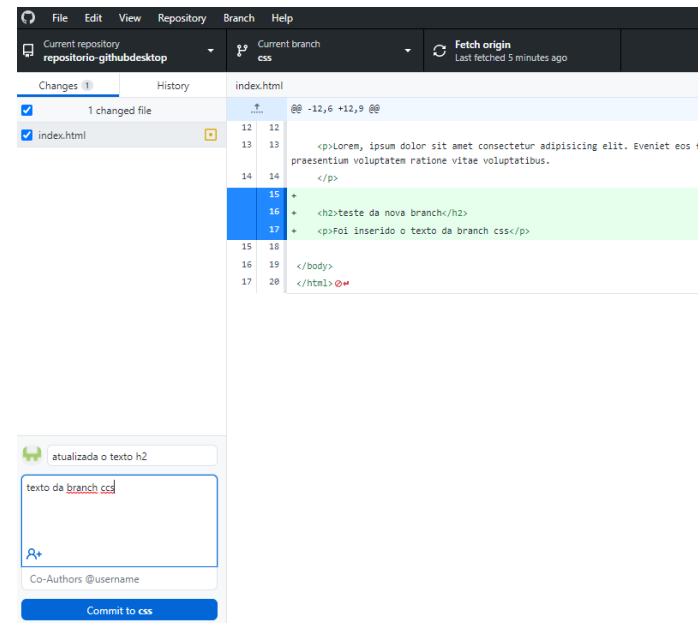
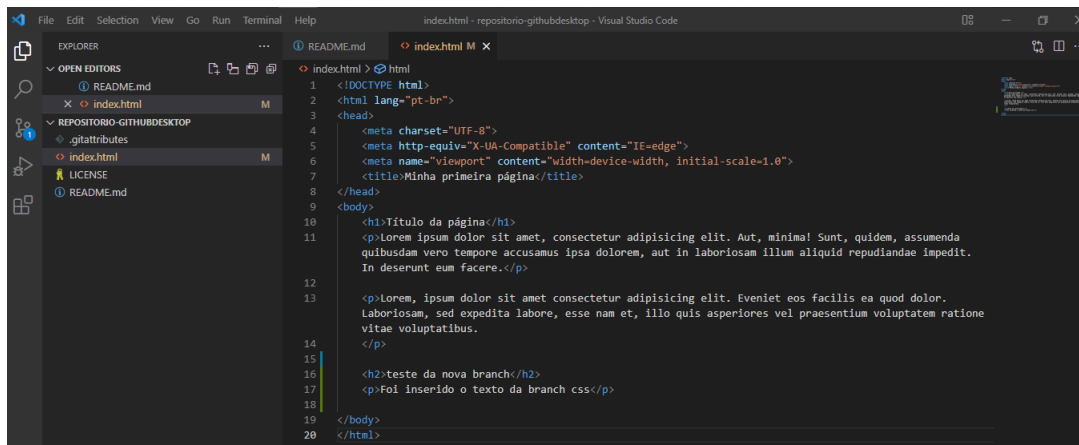


- Depois clicar em *Publish branch*



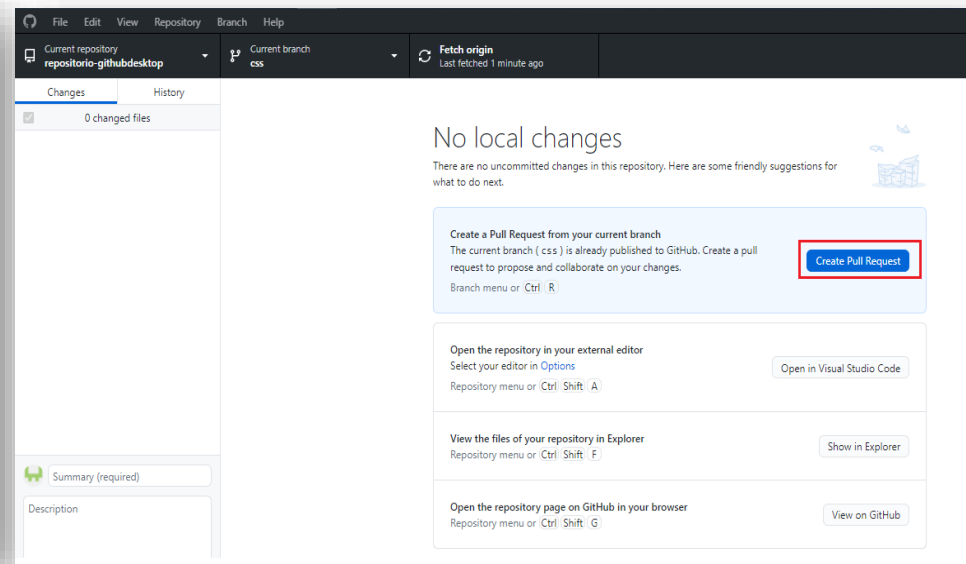
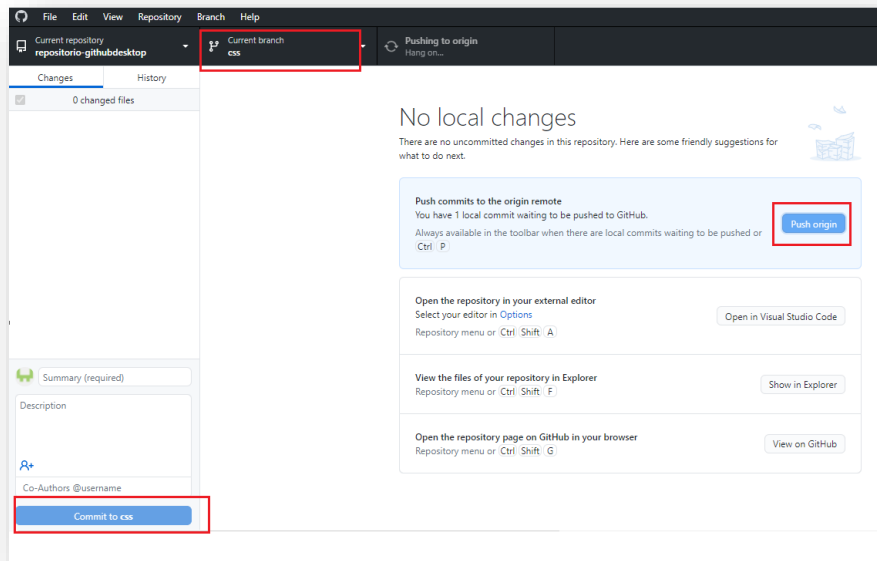
Modificando o arquivo na nova branch

- Realiza as modificações no arquivo
- Faz o commit na branch nova.



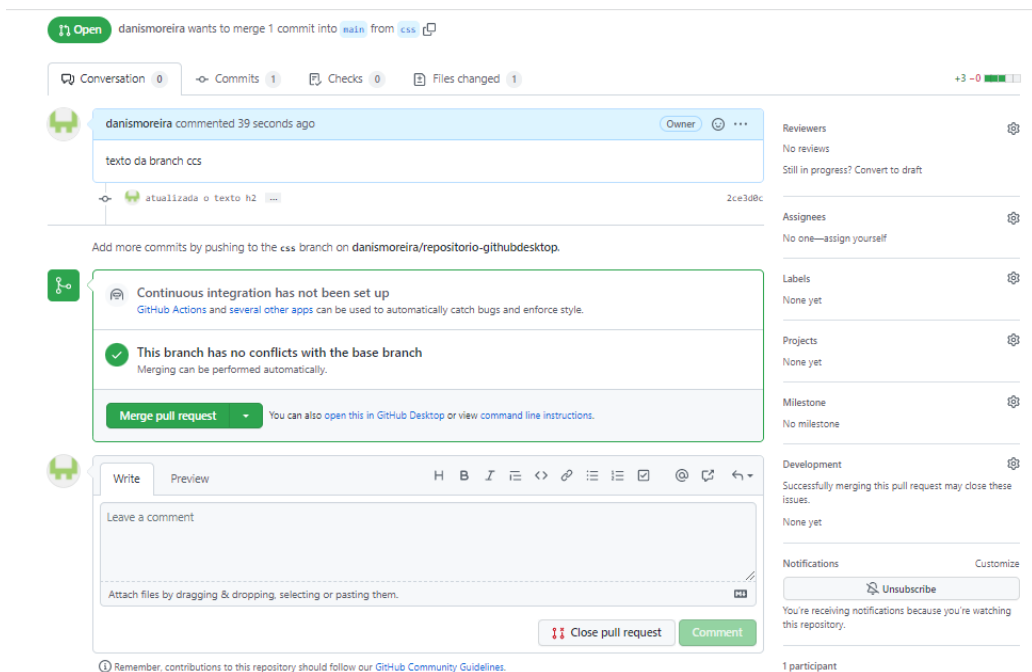
Criar PR (Pull Request)

- Após atualizar a “branch” localmente (por meio do “commit”) é necessário clicar em “Push origin” para enviar os dados ao repositório remoto.



Fazer Merge

- Após analisar a solicitação de mudança aprovar o merge
- Github remoto ficará com os mesmos dados da máquina local.



Dúvidas ???



Referências

- Canal YouTube <https://www.youtube.com/c/AngeloLuz>
- Canal YouTube
<https://www.youtube.com/channel/UCrWvhVmt0Qac3HgsjQK62FQ>
- Canal do Youtube Curso em Vídeo – Git e GitHub. Disponível em <https://www.cursoemvideo.com/>