

PrimeiroRepositorio

Teste do Git Hub para desafio da DIO bootcamp da TQI

Olá Colegas! Estou muito feliz por estar entrando neste mundo, espero aprender e ajudar aqui nessa magnifica plataforma!

Gostaria de deixar neste repositório meu aprendizado, o passo a passo para conseguir realizar.

1º Realizei o curso "Introdução ao Git e ao GitHub" com Otávio Reis no bootcamp;

2º Realizei o curso "Criando seu repositório no GitHub para Compartilhar seu progresso" com Venilton Falvo Jr no bootcamp da TQI;

3º Realizei os passos com o video do "Curso de Git e Github COMPLETO 2021 [Iniciantes] + Desafios + Muita Prática" com Jhonatan da DevAprender.

1. - Instalei o Git no computador Windows 64 bits

link: <http://git-scm.com/download/win>

Segui confirmando todas telas para instalação;

2. - Instalei o Typora, aplicativo indicado pelo Otávio Reis para salvar arquivo em tipo md

link: <https://typora.io/>

3. - Criei minha conta no GitHub = wirlamaia

link: <https://github.com/signup?source=login>

4. - ***Importante*** Para conseguir realizar conexões da máquina com o GitHub precisei registrar um token (Senha)

[Seguir este Tutorial. Link](<https://www.alura.com.br/artigos/nova-exigencia-do-git-de-autenticacao-por-token-o-que-e-o-que-devo-fazer>)

5. - Para excluir um repositório do seu GitHub, seleciona-o, clica no botão Settings, ao final da página botão **"Delete this repositor"**, digita o nome de usuario e / (barra) e o nome do repositório, botão **"I Understand"**, Senha de usuário, botão **"Confirm password"**; Pronto, excluído.

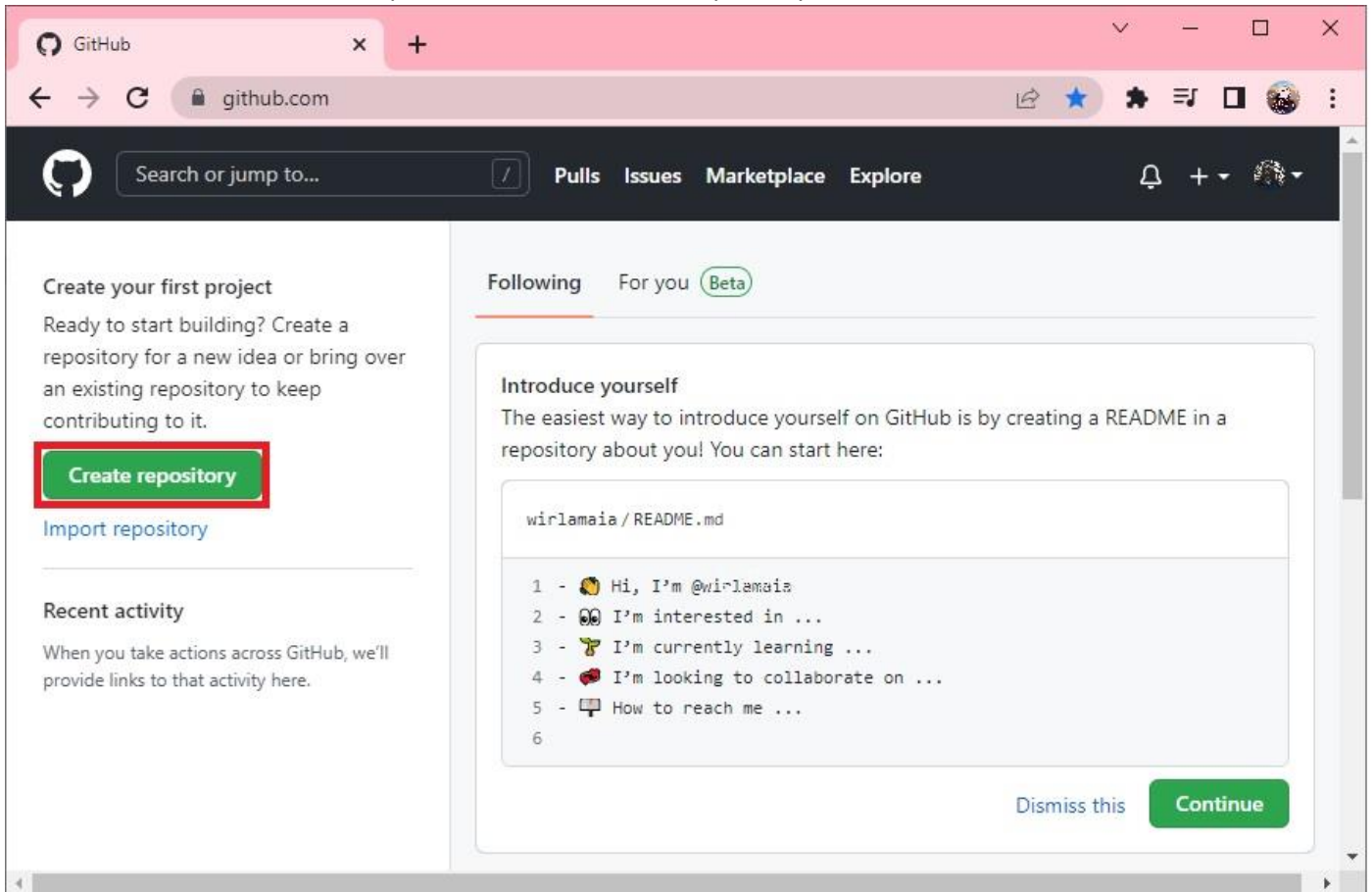
6. Principais comandos utilizados, importantes ao iniciar:

COMANDOS GIT UTILIZADOS

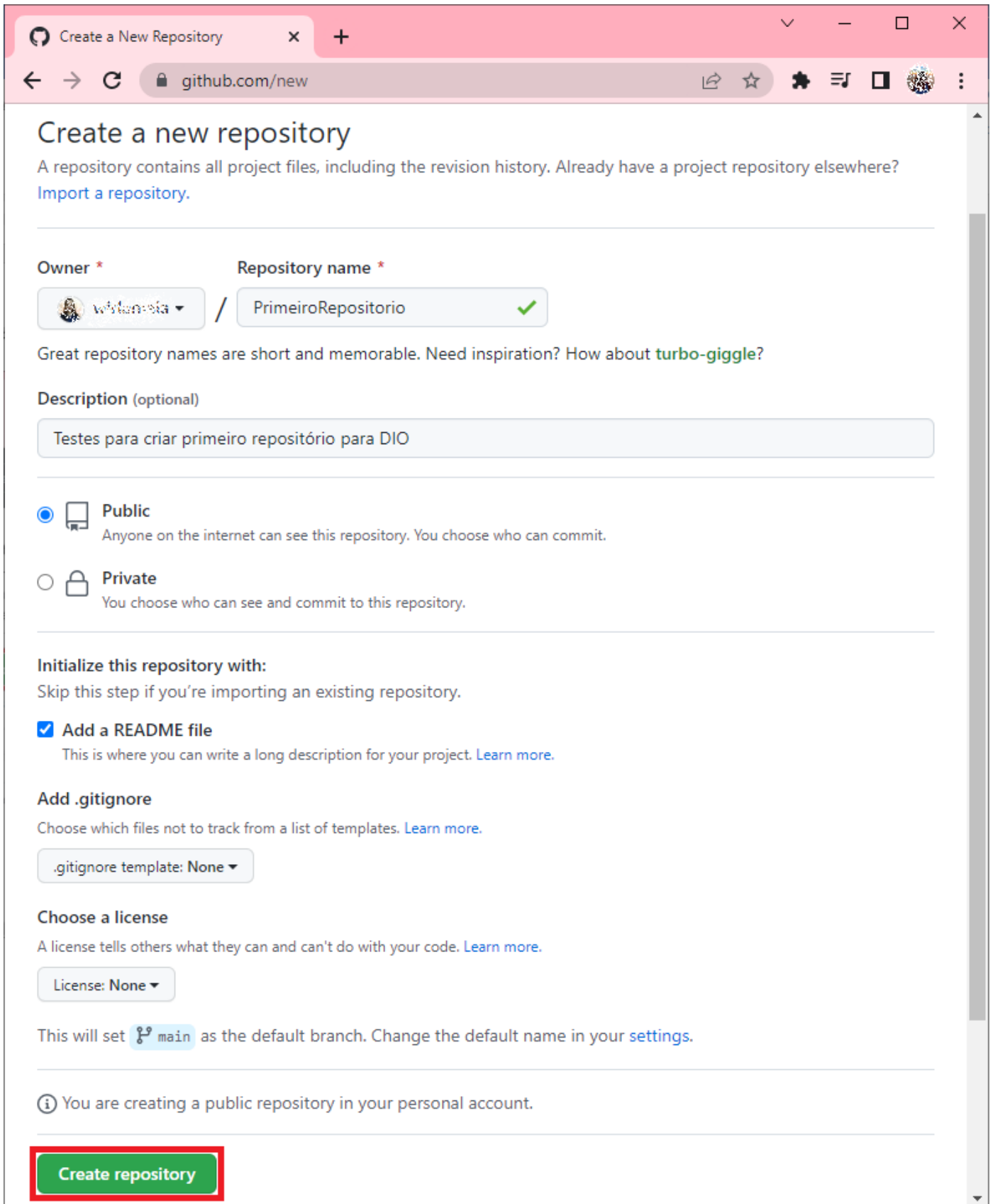
- \$ **git clone** <https://github.com/wirlamaia/PrimeiroRepositorio.git> - Clonar o repositório do GitHub online para o Git local;
- \$ **cd** "Nome do repositório" – Trabalhar no repositório Git local;
- \$ **touch .gitignore** – Criar arquivo ".gitignore" que se informa os arquivos e pastas que não precisam ser enviados da pasta para a versão Git repositório;
- \$ **CTRL+L** – Limpar a tela;
- \$ **ls** – visualizar arquivos da pasta do repositório local;
- \$ **ls -a** – visualizar arquivos da pasta do repositório local, inclusive os ocultos;
- \$ **git status** – Verificar situação do Git repositório local e arquivos conectados (sim ou não);
- \$ **git branch** – Demonstra as branches existentes, sendo a que acompanha o * símbolo é a atual utilizada;
- \$ **git branch NovoNome** – Criar uma nova branch (partição de versão apoio a original);
- \$ **git checkout NomeBranch** – Alternar para branch descrita;
- \$ **git branch -M "main"** – Alternar para branch descrita, tornando a branch como principal;
- \$ **git checkout -b NomeDaNovaBranch master** – Criar nova branch copia da branch master;
- \$ **git add Nome do arquivo** - Adicionar arquivo específico no repositório da versão git local;
- \$ **git add .** - Adicionar todos arquivos e pastas no repositório da versão git local;
- \$ **git add restore .** - Atualiza os arquivos que foram alterados da pasta na versão git local;
- \$ **git commit -m "Commit Inicial Nome dado a versão"** - Importante para identificar as fases do repositório;
- \$ **git init** – Criar arquivo "git." na pasta no computador, um dos primeiros passos, iniciando o repositório local;
- \$ **git config --global user.email "email@gmail.com"** – Informar ao Git local o email de usuário utilizado;
- \$ **git config --global user.name "NomeUsuario"** – Informar ao Git local o Nome de usuário utilizado;
- \$ **git config --global --unset user.name "NomeUsuario"** – Limpar ao Git local o Nome de usuário utilizado;
- \$ **git config --list** – Demonstrar lista de configurações;
- \$ **git remote add origin https://github.com/NomeUsuario/NomeProjetoRepositorio** - Informar ao Git repositório local para onde será enviado, endereço do GitHub;
- \$ **git push** – Enviar a versão Git local para GitHub online;
- \$ **git push --set-upstream origin main** – Ao rodar o comando git push, o Git informa para dar este comando exemplo relativo, que realizamos para enviar corretamente a versão;
- \$ **git push -u origin main** – Enviar a versão Git local para GitHub online pela primeira vez, utilizando a branch main;
- \$ **git pull** – Trazer a versão repositório do GitHub online para Git local;
- \$ **git merge NomeBranch** – Passa os arquivos do repositório da NomeBranch para o repositório master;
- \$ **git reflog** – Demonstrar as versões commitadas;
- \$ **git reset --hard Numero123Nu** – Volta o repositório para a versão informada, Código da versão demonstrada no comando git reflog, que foi inserida no comando git commit -m "Nome";
- \$ **git remote -v** - Visualizar arquivos que não foram enviados para o GitHub repositório online;

7. Prática de realizar um Repositório no GitHub:

I. Criar um novo repositório – Botão ‘Create repository’



- II. Digitar nome do repositório, no caso deixei público e a criação do arquivo README.md (Leia-me), Botão 'Create repository'




Create a New 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 *


 wilton-silva / PrimeiroRepositorio ✓

Great repository names are short and memorable. Need inspiration? How about [turbo-giggle?](#)

Description (optional)

Testes para criar primeiro repositório para DIO

☒  **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.](#)

.gitignore template: None ▾

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

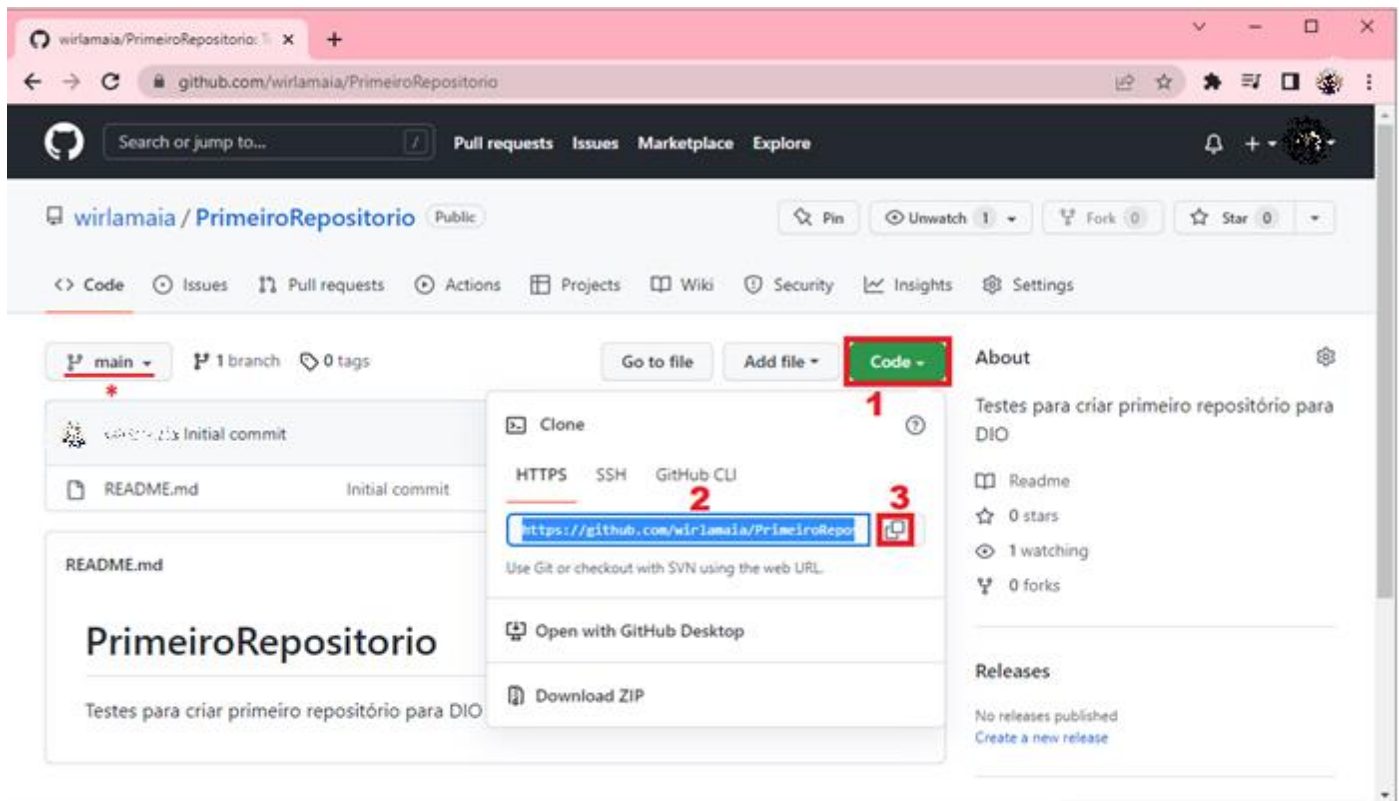
License: None ▾

This will set  **main** as the default branch. Change the default name in your [settings](#).

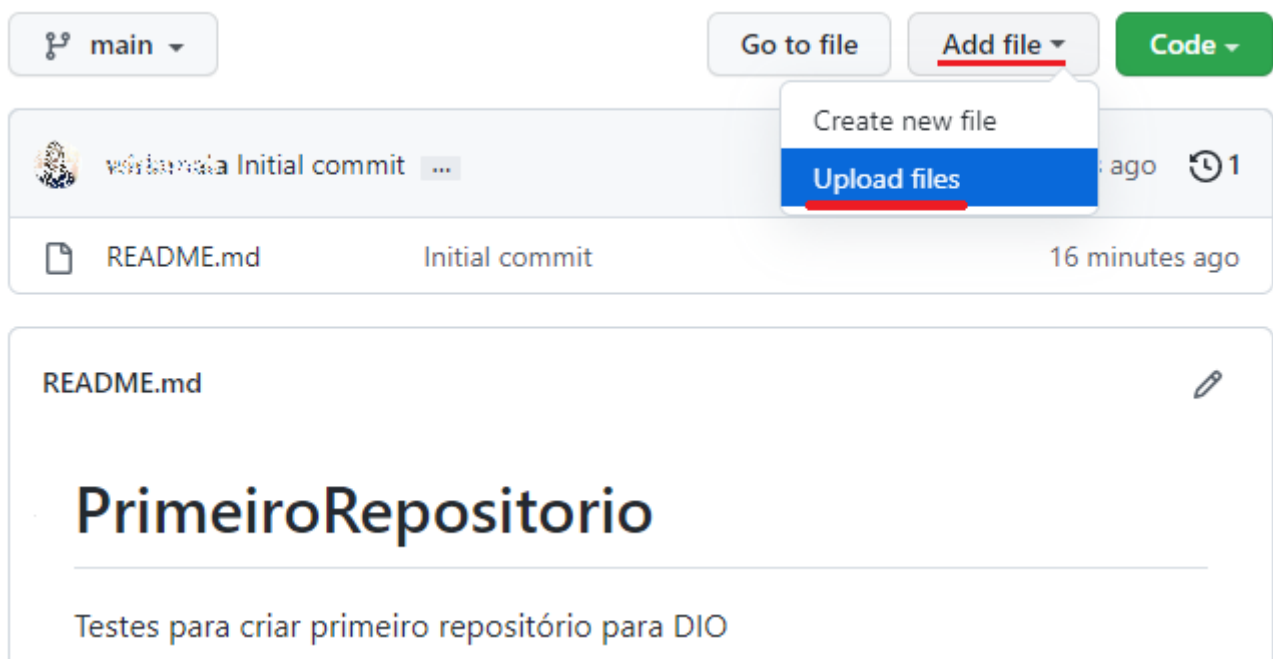
 You are creating a public repository in your personal account.

Create repository


- III. Observo que a branch principal tem o nome 'main';
No botão 'Code' 1, aba HTTPS 2, copieie o link do repositório 3.




- IV. Adicionei uma imagem, no botão 'Add file', sub-botão 'Upload files', de imagem JPG, pelo campo 'Choose your files' e botão 'Commit changes'




- V. Podendo visualizar o arquivo, clicando em cima do nome 1;
Editei o arquivo README, clicando no botão Editar 2;

 wirlamaia Add files via upload adef578 in 1 minute 2 commits

 1 Create repository.jpg 1


Add files via upload

now

 README.md

Initial commit

21 minutes ago

README.md  2

PrimeiroRepositorio

Testes para criar primeiro repositório para DIO

VI. Incluso texto, botão 'Commit changes';

Algumas frases se juntaram, editei novamente visualizando pelo 'Preview' antes de Commitar finalmente.

Editing PrimeiroRepositorio/REA x +

github.com/wirlamaia/PrimeiroRepositorio/edit/main/README.md

repository'

65 III. Observo que a branch principal tem o nome 'main';

66 No botão 'Code' 1, aba HTTPS 2, copieie o link do repositório 3.

67 IV. Adicionei uma imagem, no botão 'Add file', sub-botão 'Upload files', de imagem JPG, pelo campo 'Choose your files' e


botão 'Commit changes'

68 V. Podendo visualizar o arquivo, clicando em cima do nome 1;

69 Editei o arquivo README, clicando no botão Editar 2 (Salvando o texto até aqui), botão 'Commit changes';

70

Attach files by dragging & dropping, selecting or pasting them.



Commit changes

Update README.md

Add an optional extended description...

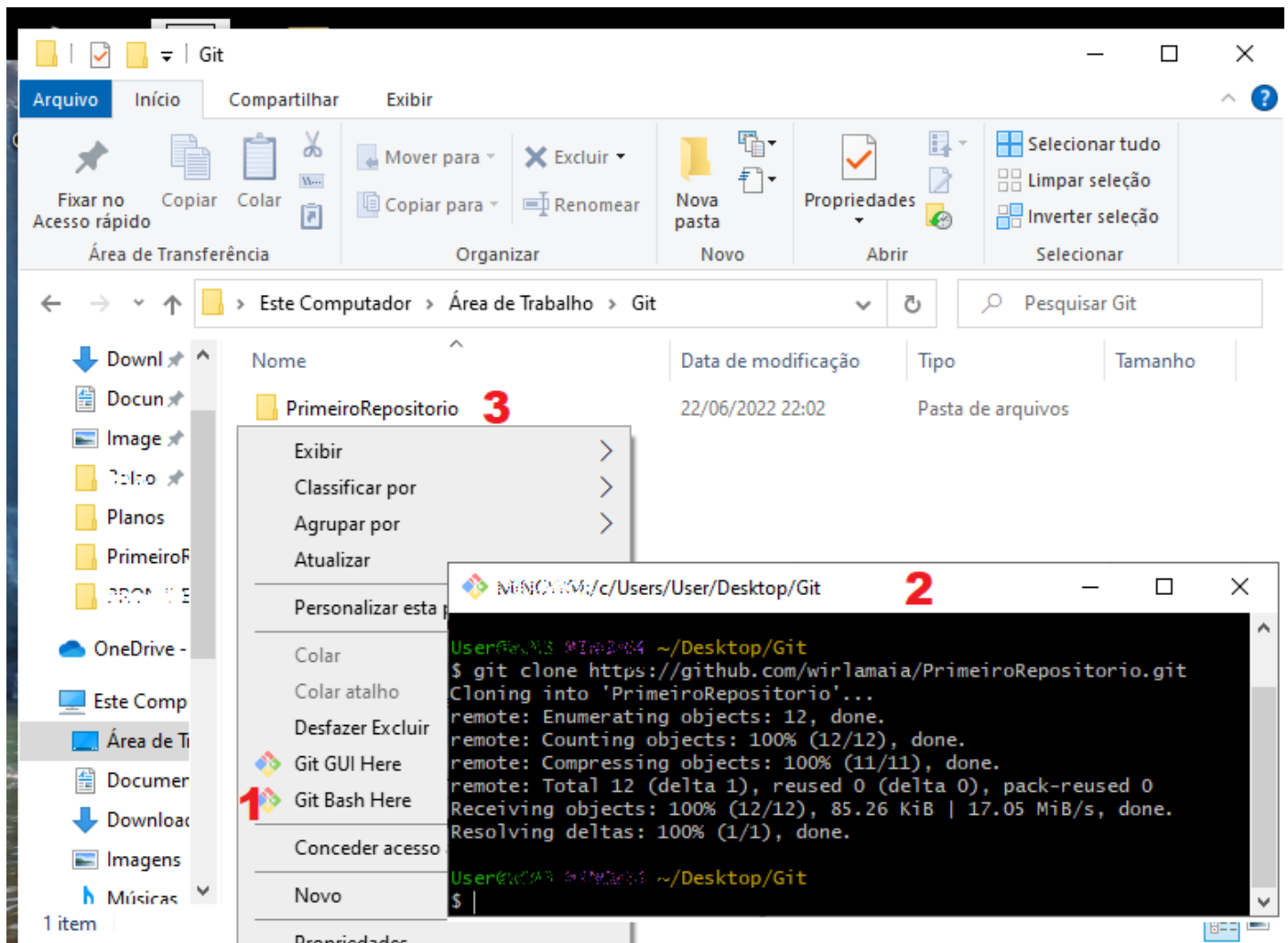
☒ Commit directly to the `main` branch.

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

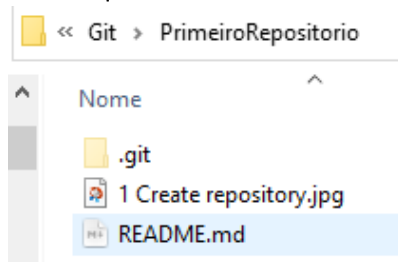
Commit changes

Cancel

8. Criei no meu computador uma pasta na área de trabalho, nome 'Git';
 - I. Dentro da pasta criada para o repositório, cliquei com botão direito do mouse e selecionei o 'Git Bash here';
 - II. Comando '\$ git clone <https://github.com/wirlamaia/PrimeiroRepositorio.git>', link do repositório;
 - III. Comando '\$ CTRL+L' para limpar tela;



i. Na pasta há os arquivos corretamente:



IV. Comando `$ cd`, testei voltando e indo para a pasta do repositório local, ao entrar nela é demonstrado a branch (main)

```
User@WIN10: ~/Desktop/Git
$ cd

User@WIN10: ~
$ cd PrimeiroRepositorio
bash: cd: PrimeiroRepositorio: No such file or directory

User@WIN10: ~
$ cd Desktop

User@WIN10: ~/Desktop
$ cd Git

User@WIN10: ~/Desktop/Git
$ cd PrimeiroRepositorio

User@WIN10: ~/Desktop/Git/PrimeiroRepositorio (main)
$
```

V. Adicionei uma pasta Introdução com o arquivo docx (este);

Comando `$ git status`;

```
User@W3: ~/Desktop/Git/PrimeiroRepositorio (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        "Introdu\303\247\303\243o/"

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

VI. Comando `$ git add .`, adicionando o arquivo na versão local;

```
User@W3: ~/Desktop/Git/PrimeiroRepositorio (main)
$ git add .

User@W3: ~/Desktop/Git/PrimeiroRepositorio (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   "Introdu\303\247\303\243o/PassoAPasso_PrimeiroRepositorio_WirlaMaia.docx"
```

VII. Comando `$ git commit -m "Commit Inicial Nome dado a versão"`, Informando as mudanças ocorridas para organizar os commits;

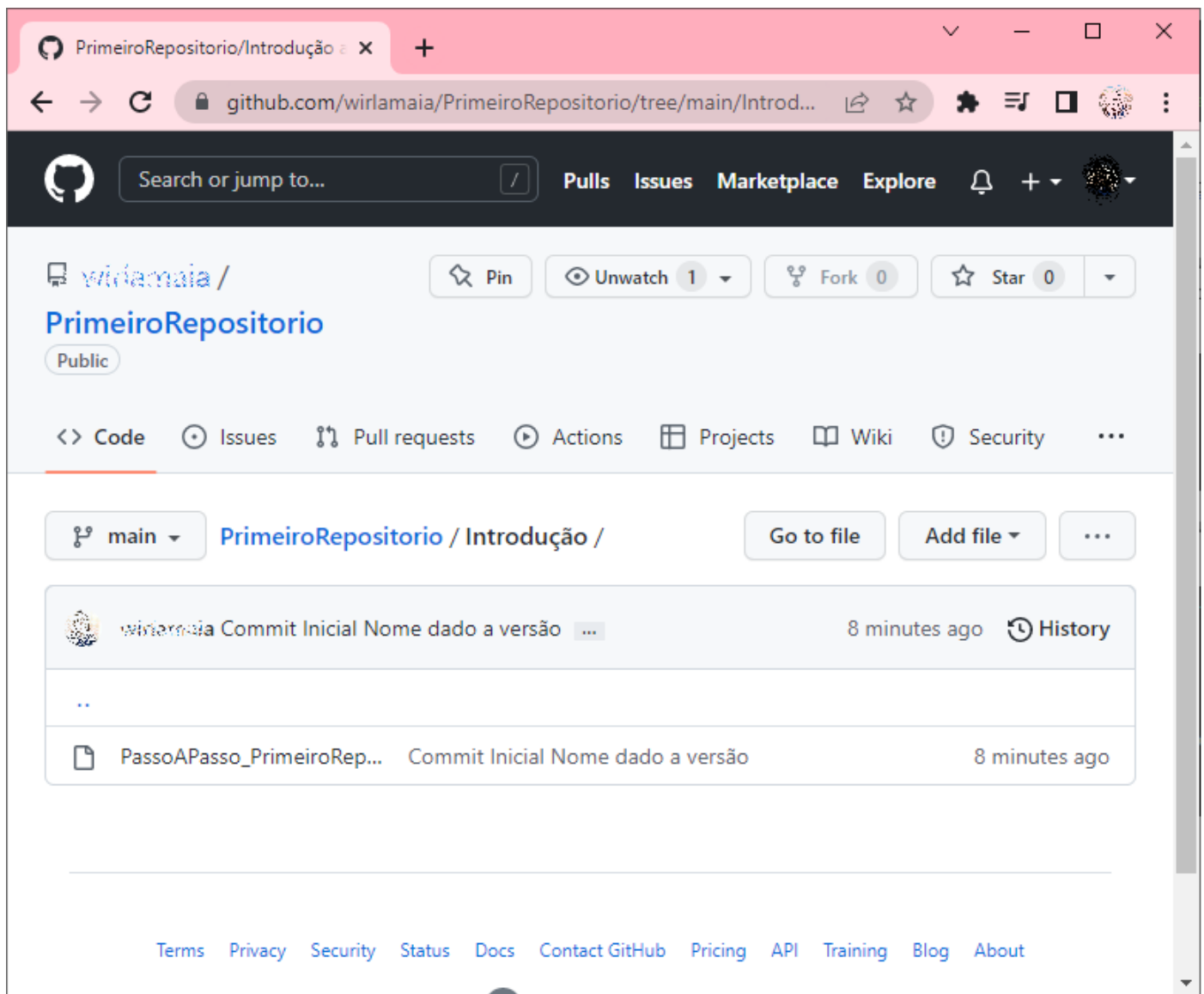
Obs: para colar o texto na tela de comandos, preciso clicar com o botão direito do mouse e clicar em 'paste', devido o atalho informado SHIFT+Ins não funcionar; e não consigo copiar do word quando o comando possui aspas, pois desconfigura na tela de comando, então copio do bloco de notas;

```
User@W3: ~/Desktop/Git/PrimeiroRepositorio (main)
$ git commit -m "Commit Inicial Nome dado a versão"
[main a77cd48] Commit Inicial Nome dado a versão
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "Introdu\303\247\303\243o/PassoAPasso_PrimeiroRepositorio_WirlaMaia.docx"
```

VIII. Comando `$ git push origin main`, Enviando as alterações local da máquina para o servidor do repositório no GitHub online;

```
User@W3: ~/Desktop/Git/PrimeiroRepositorio (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 636.89 KiB | 14.81 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/wirlamaia/PrimeiroRepositorio.git
9682107..a77cd48 main -> main
```

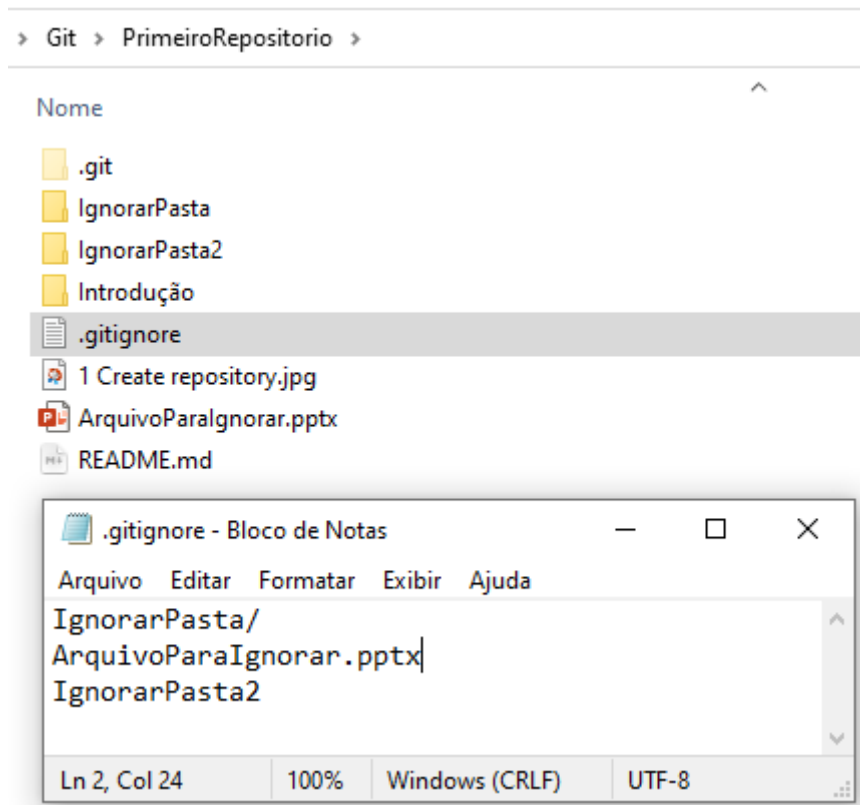
GitHub, F5 para atualizar, já está com as alterações:



- IX. Comando `$ touch .gitignore` – Criar um arquivo na pasta para ignorar o que não desejar que seja enviado a versão do repositório;

```
.gitignore
~/Desktop/Git/PrimeiroRepositorio
User@GLASVB: ~/Desktop/Git/PrimeiroRepositorio (main)
$ touch .gitignore
```

Nesse arquivo coloquei uma pasta que não precisa ir para a versão:



Assim estes arquivos não são demonstrados para atualização, apenas os inseridos que podem realizar o upload aparecem. Vou salvar este arquivo em PDF, dentro da pasta Introdução para exemplo e o arquivo .gitignore é obrigatório ser enviado.

X. Comando \$ **git status**;

```
User@W4M3: ~/Desktop/Git/PrimeiroRepositorio (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  .gitignore
  "Introdu\303\247\303\243o/PassoAPasso_PrimeiroRepositorio_WirlaMaia2.pdf"

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

XI. Comando \$ **git branch Apoio** – Criar uma nova branch (partição de versão apoio a original);

Comando \$ **git branch -M "Apoio"** – Alternar o repositório para branch descrita. Perceba que muda a branch na tela escrita em azul.

```
User@W4M3: ~/Desktop/Git/PrimeiroRepositorio (main)
$ git branch Apoio

User@W4M3: ~/Desktop/Git/PrimeiroRepositorio (main)
$ git branch -M "Apoio"

User@W4M3: ~/Desktop/Git/PrimeiroRepositorio (Apoio)
$ |
To https://github.com/wirlamaia/PrimeiroRepositorio.git
 9682107..a77cd48 main -> main
```

XII. Comando \$ **git push origin Apoio**, Enviar as alterações para a branch Apoio no Git Hub

Essa alteração na branch apoio pode ser levada para a principal Main, pelo comando \$ **git merge main** – Porém não irei realizar para ficar arquivos diferentes em cada branch.

The screenshot shows the GitHub interface for the repository 'wirlamaia/PrimeiroRepositorio'. The browser address bar shows the URL 'github.com/wirlamaia/PrimeiroRepositorio/tree/Apoio'. The repository is public and has 2 branches and 0 tags. The 'Apoio' branch is selected and highlighted with a red box. A message states 'This branch is up to date with main.' Below this, a commit history table is visible:

Commit	Author	Message	Time
a77cd48	wirlamaia	Commit Inicial Nome dado a versão	1 hour ago
		5 commits	

Below the commit history, a file list shows:

- Introdução: Commit Inicial Nome dado a versão, 1 hour ago
- 1 Create repository.j...: Add files via upload, 2 hours ago
- README.md: Update README.md, 2 hours ago

The README.md file is expanded, showing the title 'PrimeiroRepositorio' and the content 'Teste do Git Hub para desafio da DIO bootcamp da TQI'.

On the right side, the 'About' section shows 'Testes para criar primeiro repositório para DIO' and statistics: 0 stars, 1 watching, 0 forks. The 'Releases' section shows 'No releases published' with a link to 'Create a new release'. The 'Packages' section shows 'No packages published' with a link to 'Publish your first package'.