

Aula #16: Desvendando o Github

- Parte 1

Github	1
Aba Overview	2
Aba Repositories	3
Criação de Repositório	4
Dentro de um Repositório	7
Clonando um repositório	8
Issues	9
Pull Requests	11
Próxima Aula	12
Exercícios	12
Fontes e Links Complementares	12

Github

Como já vimos nas aulas anteriores, o Github é uma ferramenta de armazenamento de repositórios Git, onde podemos armazenar e compartilhar nossos projetos com outras pessoas. Ela é uma das primeiras ferramentas do tipo a serem criadas e uma das mais utilizadas. Atualmente ela é mantida pela Microsoft.

Conseguimos, além de armazenar nossos projetos e contribuir com projetos de terceiros, fazer toda a parte de CI/CD utilizando o Github Actions e gerenciar nossas

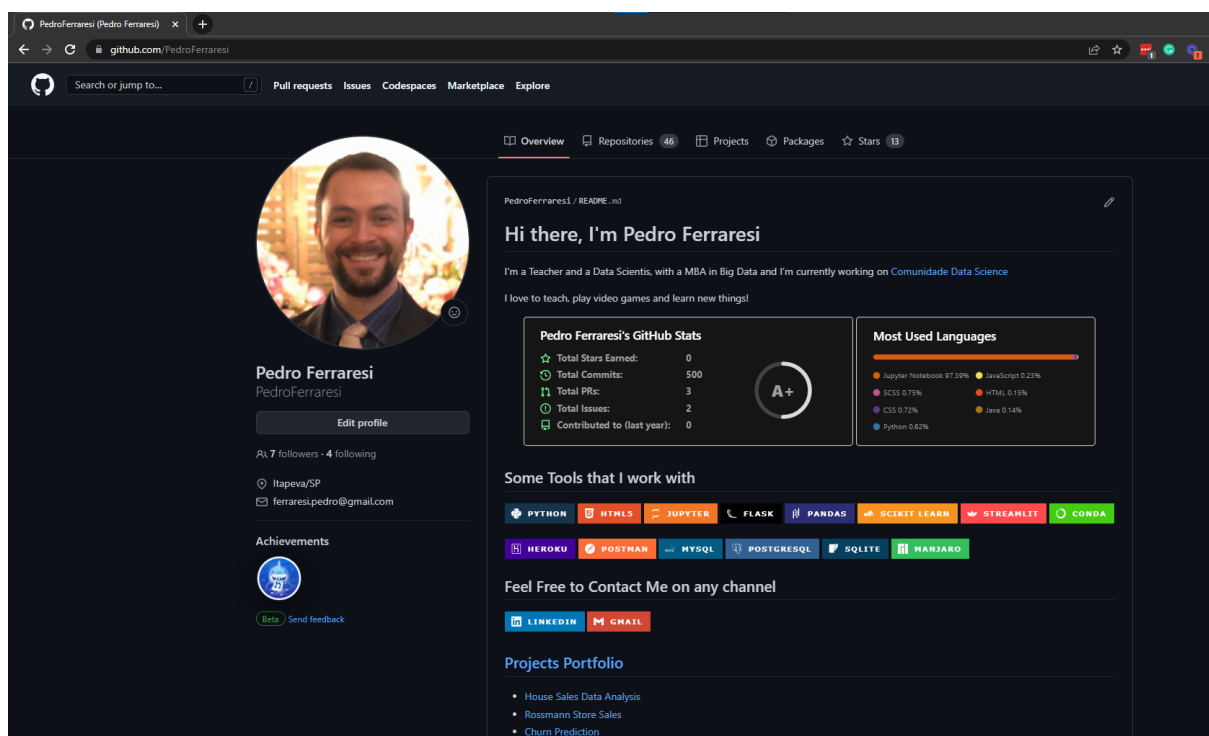
issues dentro da própria ferramenta. Infelizmente, Github Actions está além do escopo do curso, bem como utilizamos as issues do Github para gerenciar nossos projetos.

Nesta aula iremos verificar as principais funcionalidades da ferramenta, com o enfoque de te auxiliar a encontrar as funcionalidades que iremos demonstrar, bem como o uso básico da ferramenta do Github.

Lembrando que o Github Desktop também está fora do escopo do curso.

Aba Overview

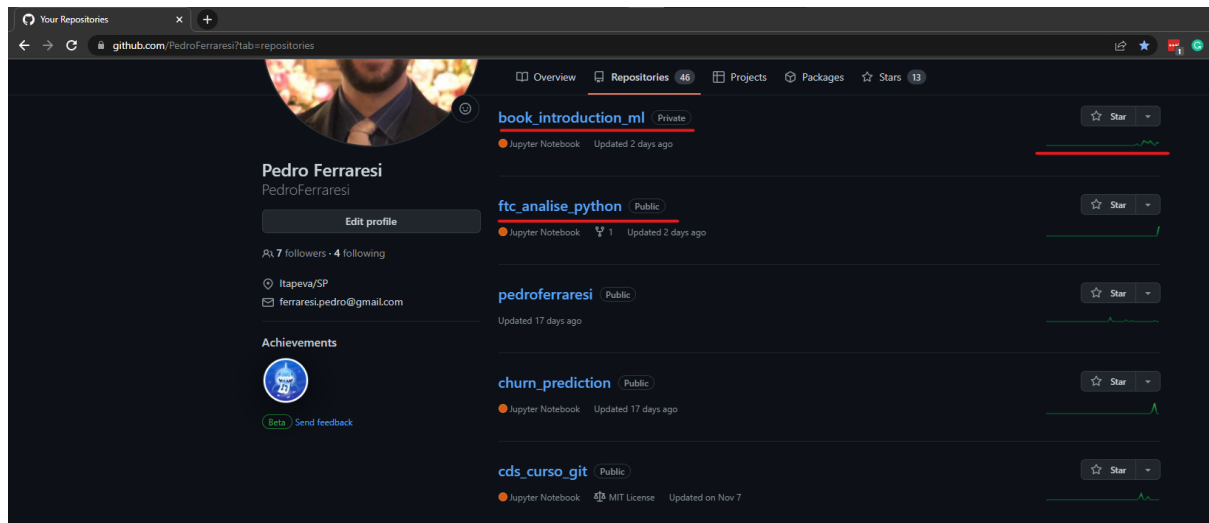
A aba de overview é uma das principais abas do Github. Quando alguém entrar em seu perfil, é ela que o navegador irá carregar.



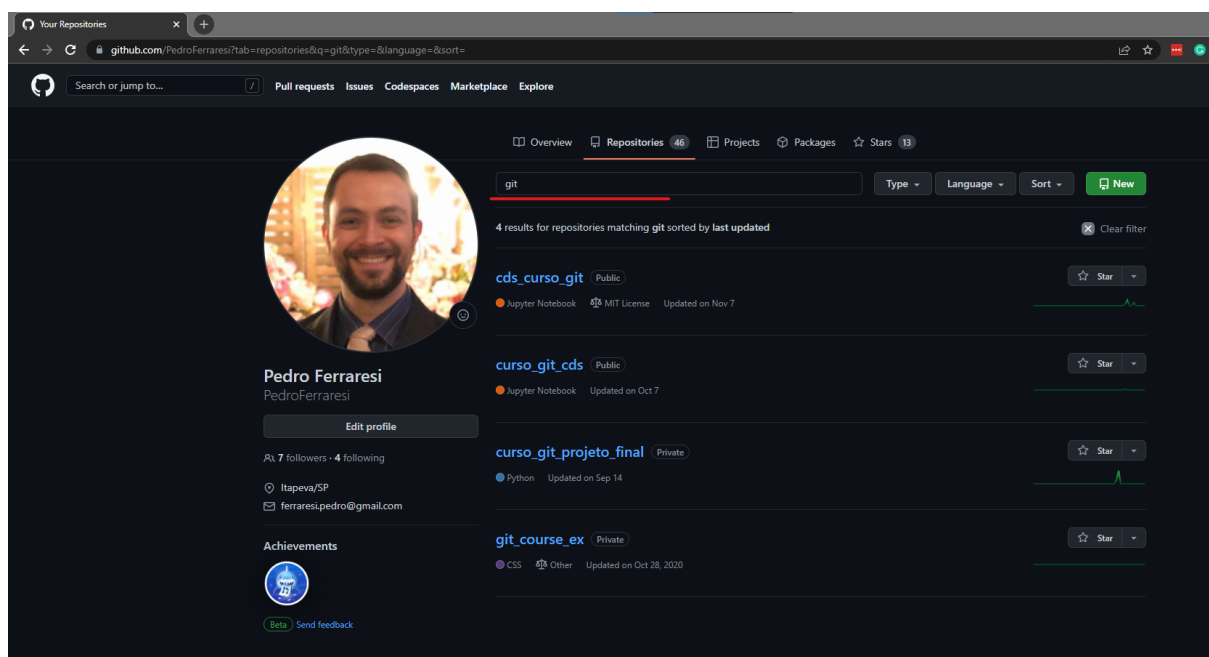
Dessa forma, manter uma página de perfil bem escrita e organizada é essencial na utilização do Github. Novamente, como construir e organizar essa página está fora do escopo do curso, porém existem inúmeros tutoriais na internet demonstrando como ajustar a página principal do Github.

Aba Repositories

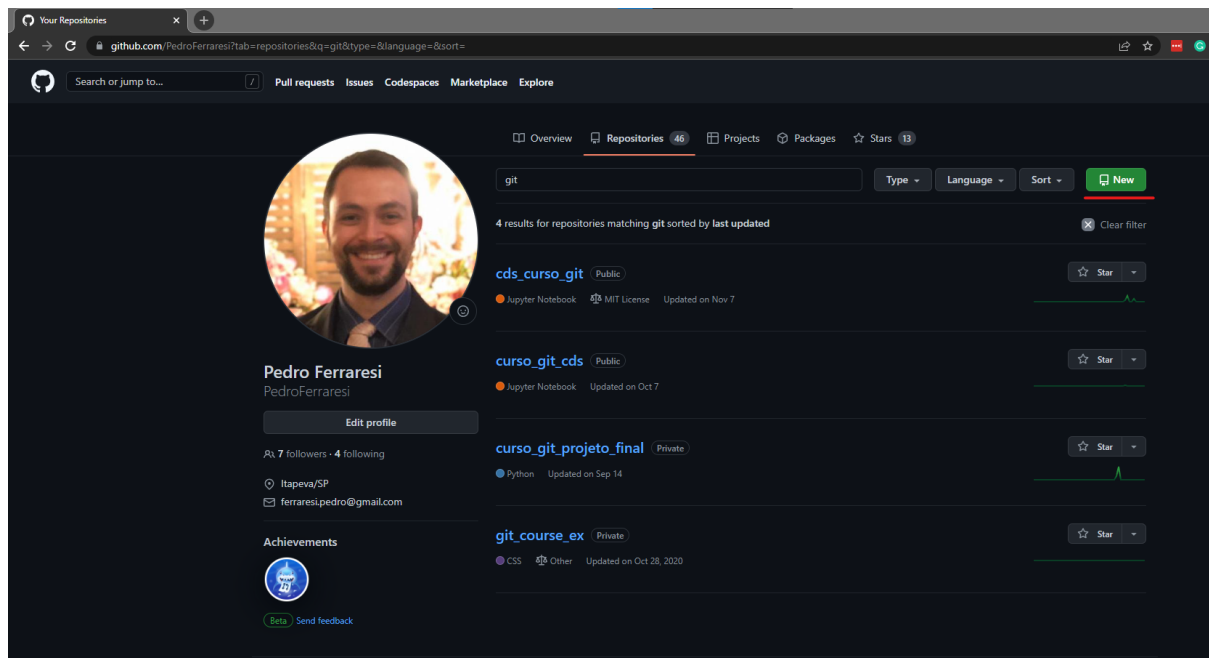
A aba de é onde estão todos os repositórios cadastrados na sua conta. Nesta aba podemos ver o nome dos repositórios, se eles são públicos ou privados e a frequência com a qual “atualizamos” o repositório.



Além disso, caso seja necessário buscar um repositório específico, podemos utilizar o campo de busca para isso.



Outra funcionalidade importante desta aba é a criação de novos repositórios. Para isso, basta clicar no botão verde, no canto superior direito, escrito new:



Embora já tenhamos feito esse processo de criação anteriormente, iremos refazê-lo para explicar melhor certas etapas configurações que ainda não foram explicadas.


Criação de Repositório

Para criar um repositório dentro do Github é muito simples: Basta clicar no botão verde escrito New, conforme imagem anterior, para ser direcionado na página de criação. Dentro da página de criação, será necessário inserir alguns detalhes do repositório, como nome, se ele é público ou privado, se iremos ou não utilizar um `README`, `.gitignore` e licença para o nosso projeto.

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 ***

 **PedroFerraresi** /

Great repository names are short and memorable. Need inspiration? How about [upgraded-octo-guide?](#)

Description (optional)

☒  **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 ▾

 You are creating a public repository in your personal account.

[Create repository](#)

Vamos criar um repositório de teste para utilizarmos nesta e na próxima aula, para exemplificar o funcionamento básico do github. Para isso, crie um repositório com as seguintes informações:

- **Nome:** curso_git_teste_github
- **Visibilidade:** privado
- **Arquivo README:** Sim


- `.gitignore`: Python
- Tipo de licença: MIT

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 *

 PedroFerraresi ▾

/ curso_git_teste_github ✓

Great repository names are short and memorable. Need inspiration? How about [upgraded-octo-guide?](#)

Description (optional)

☐ 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: Python ▾

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

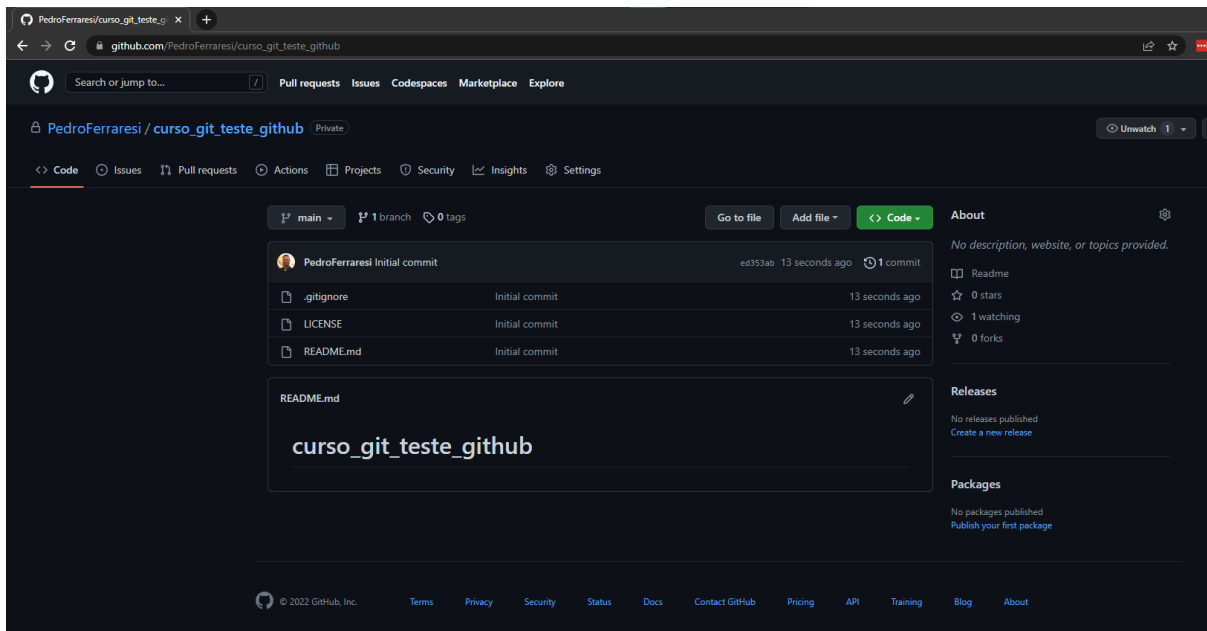
License: MIT License ▾

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

 You are creating a private repository in your personal account.

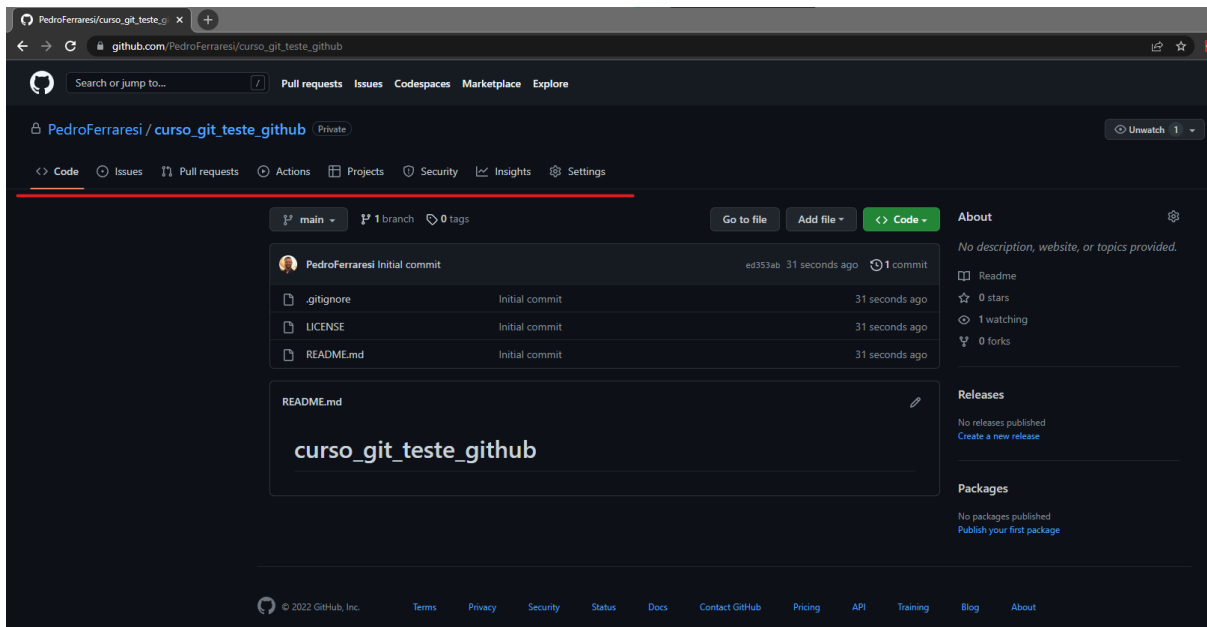
Create repository

Feito isso, basta clicar no botão `Create Repository` para criar o repositório em questão na sua conta do Github.



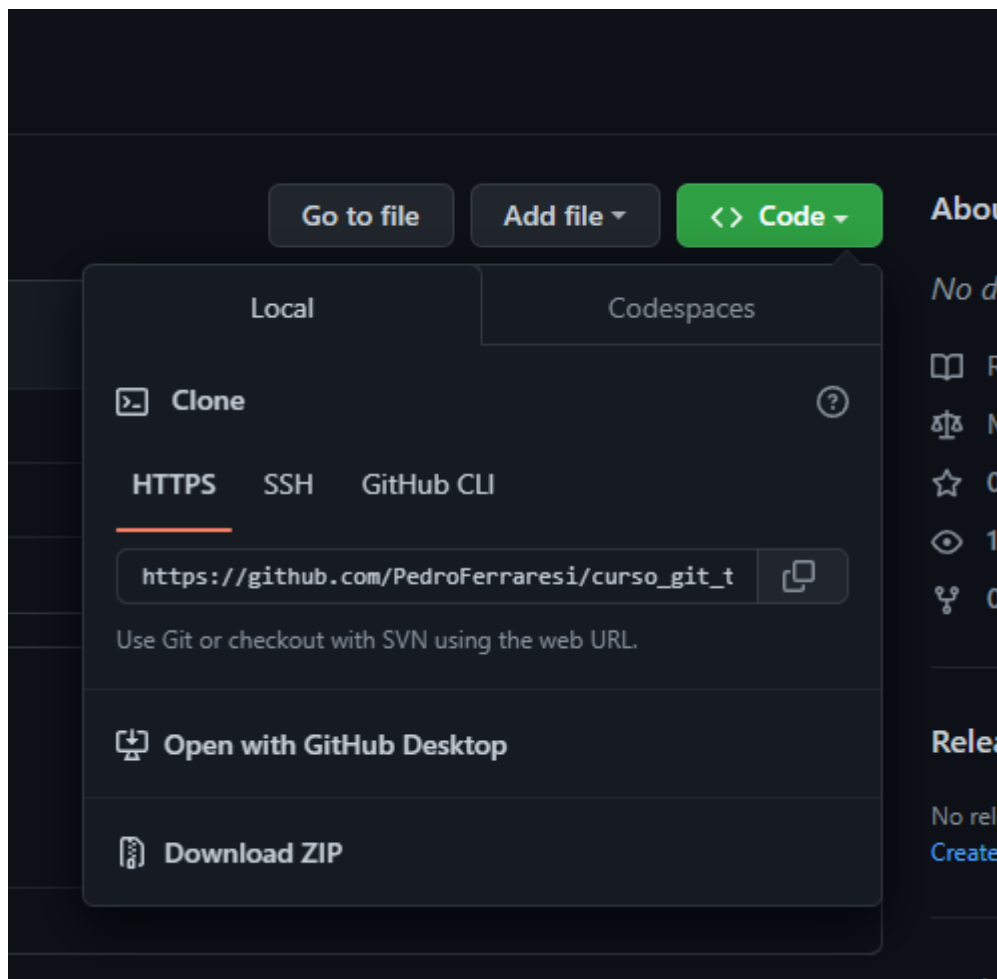
Dentro de um Repositório

Dentro de todos os repositórios do Github, existem ainda outras abas e funcionalidades. Vamos verificar as mais importantes.



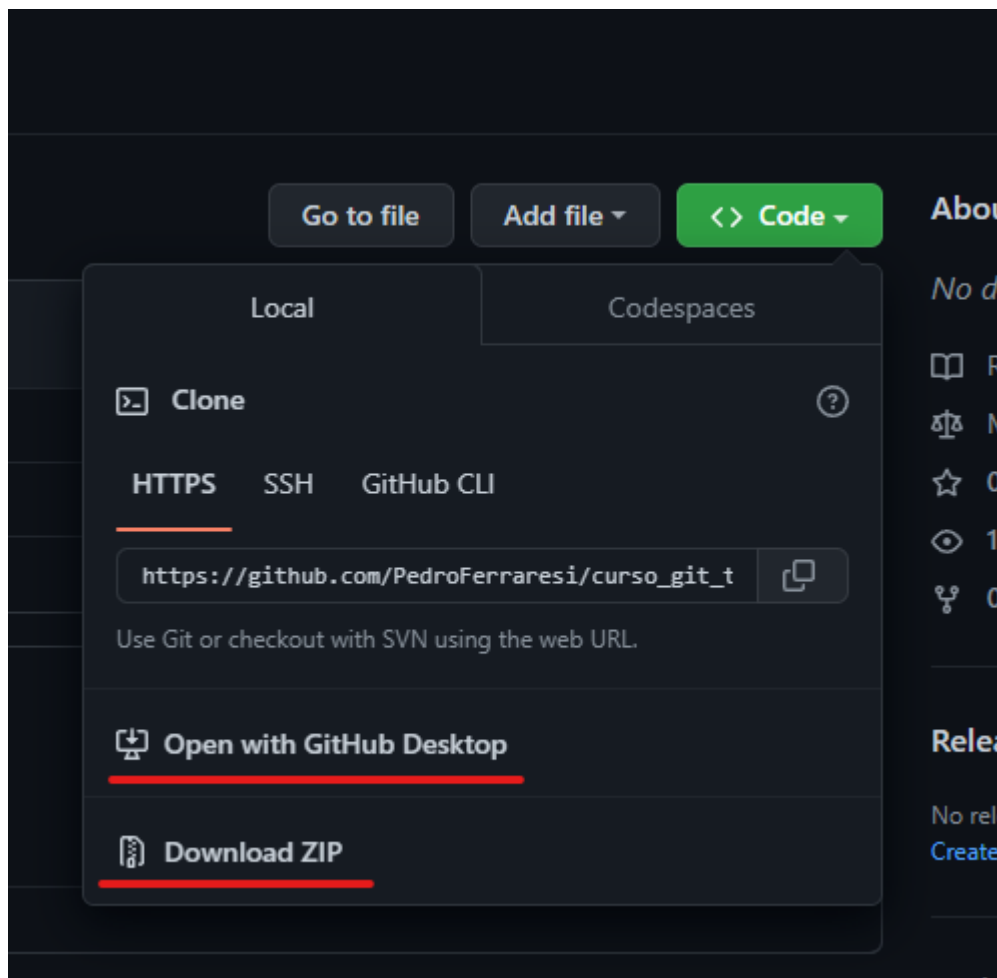
Clonando um repositório

Embora já tenhamos feito esse processo, para clonarmos um repositório do Github em nossa máquina local, clicamos no botão verde chamado code, e selecionamos a maneira que desejamos clonar o projeto.



Quando utilizamos a opção **HTTPS**, que é a mais comum, estamos baixando o projeto em nossa máquina utilizando o protocolo web **HTTPS**. Nele, precisamos informar nosso usuário e senha para nos autenticarmos e baixarmos nosso projeto. Porém, podemos ainda utilizar as opções SSH e Github CLI para download e autenticação. Ambas estão fora do escopo do curso.

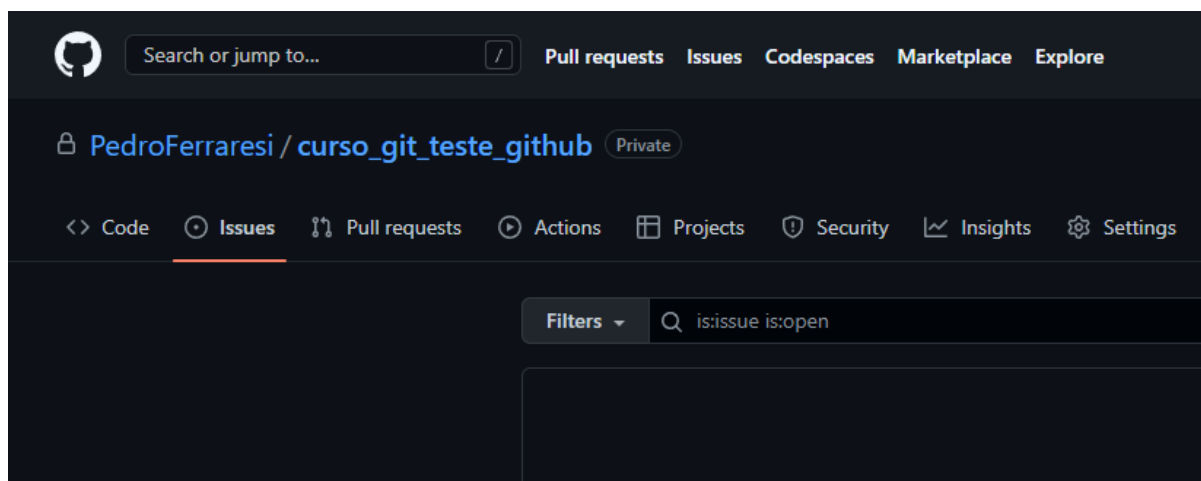
Além disso, podemos também baixar o projeto inteiro em um arquivo zip ou utilizando a aplicação Github Desktop.



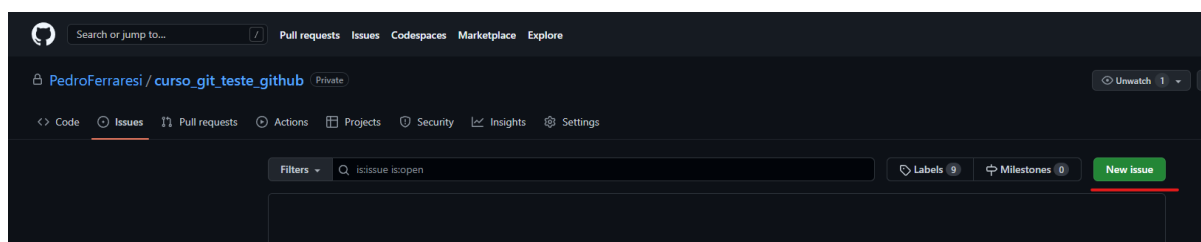
O mais comum dos processos é a utilização do protocolo [HTTPS](#), que é o mecanismo que estamos abordando neste curso.

Issues

A Aba de Issues de um repositório do Github serve para controlarmos bugs, features (funcionalidades) e milestones dos nossos projetos.

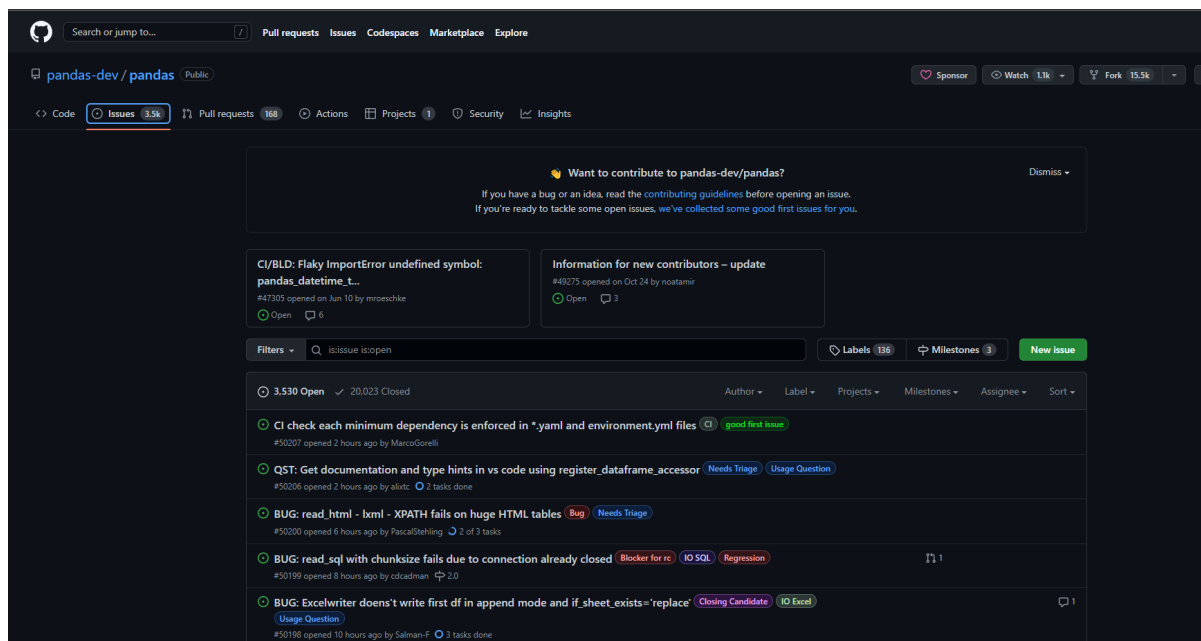


Para criar uma nova Issue, basta clicar no botão verde chamado New Issue



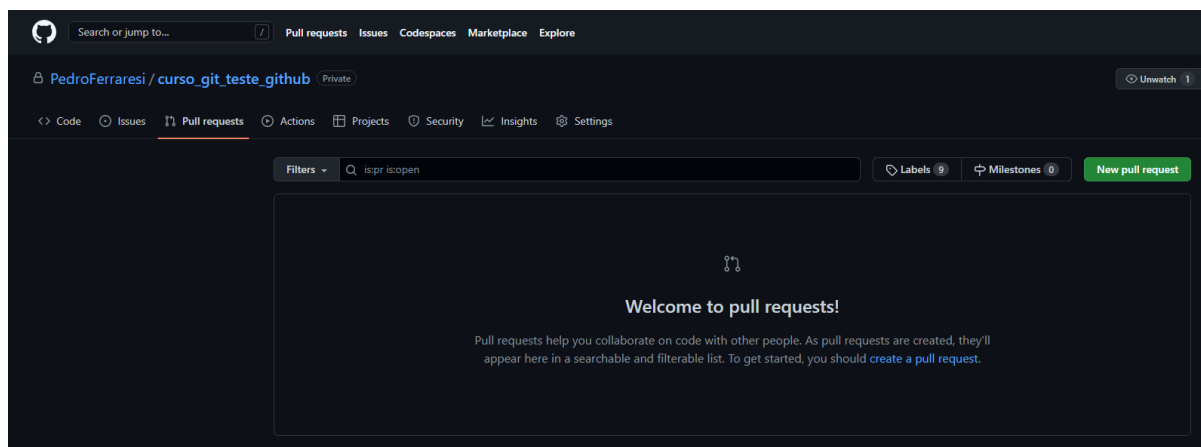
Dentro da página de criação de issue, será pedido que coloquemos o nome da Issue, a sua descrição, qual o usuário responsável por ela, seus labels, projetos a qual pertence e milestones. Como o gerenciamento de issues utilizando o Github foge do escopo do curso, não iremos criar nenhuma issue.

Mas as issues são muito utilizadas no gerenciamento do projeto, funcionando como um controle de tarefas que devem ser executadas dentro do projeto. Fora isso, quando utilizamos ferramentas que são *open-source*, como o pacote [Pandas](#) do Python, podemos abrir issues dentro do projeto no Github para que ela seja verificada



Pull Requests

Outra aba muito importante e muito utilizada dentro do Github é a de **Pull Requests**. Um **Pull Request** é quando fazemos alterações em um repositório, e enviamos nossas alterações para verificação e, caso essas alterações sejam aceitas, elas são incorporadas dentro do projeto.



Esse mecanismo de Pull Request é algo muito utilizado dentro do trabalho colaborativo dentro do Github. Porém, utilizar e demonstrar o Pull Requests está fora do escopo do curso.

Próxima Aula

Na próxima aula, finalizar o tour pela ferramenta do github.

Exercícios

Nesta aula não haverá exercícios, uma vez que é somente o tour pela ferramenta.

Fontes e Links Complementares

[Documentação - Github - Autenticação](#)

[Documentação - Github - Conexão via SSH](#)

[Documentação - Github - Issue](#)

[Documentação - Github - Pull Requests](#)