

# Tutorial GitHub Desktop

## Crie uma conta no GitHub

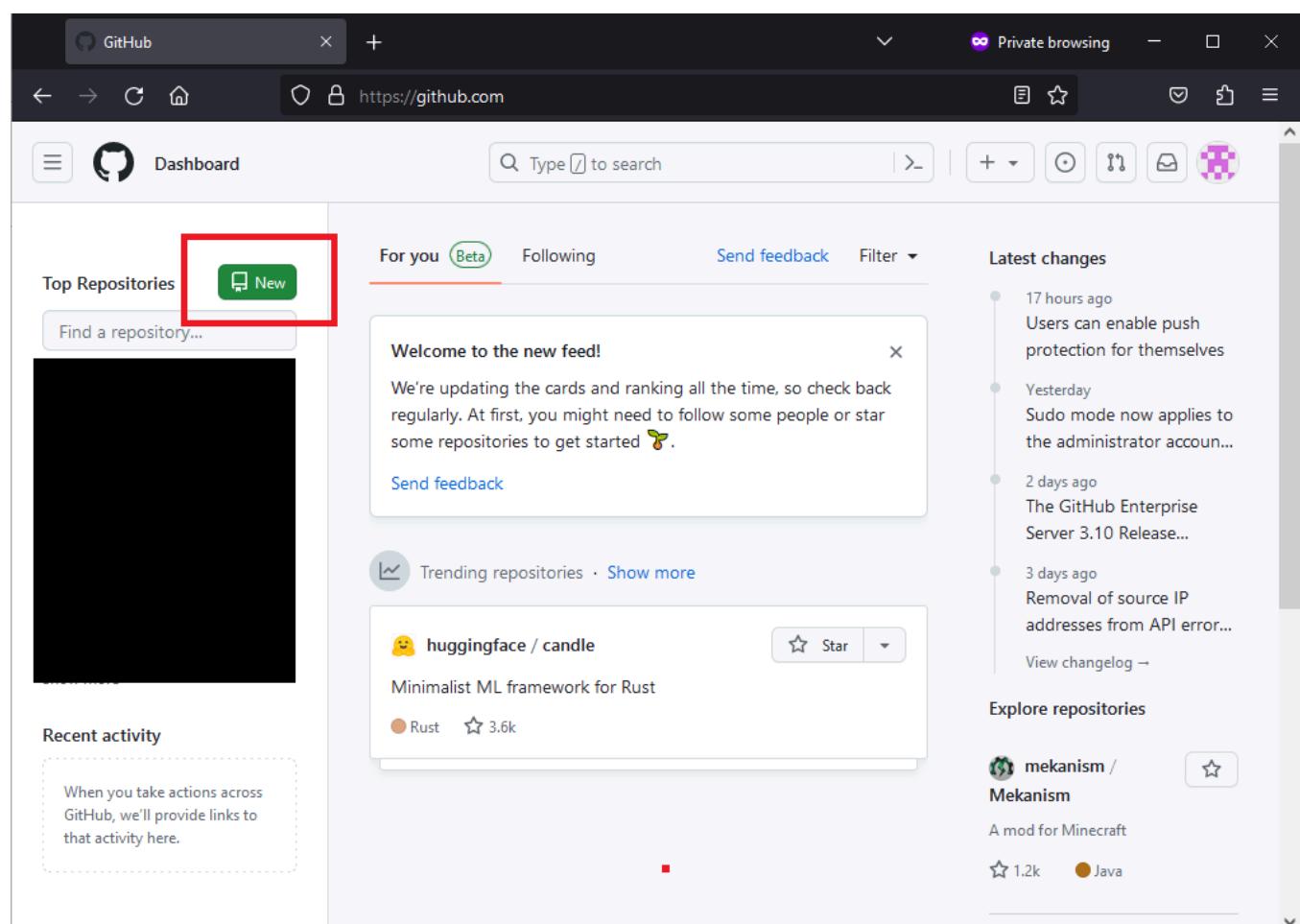
<https://github.com/signup?source=login>

## Faça Login

<https://github.com/login>

## Criar novo repositório

1. Inserir o nome do repositório
2. Marcar a opção **Público**
3. Marcar a opção **ADD README**
4. Selecionar a opção **Create Repository**



## Create a new repository

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

Required fields are marked with an asterisk (\*).

Owner \*



Repository name \*

Ipro

Ipro is available.

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

Description (optional)

atividade de lógica de programação

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:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template:None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

License:None ▾

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

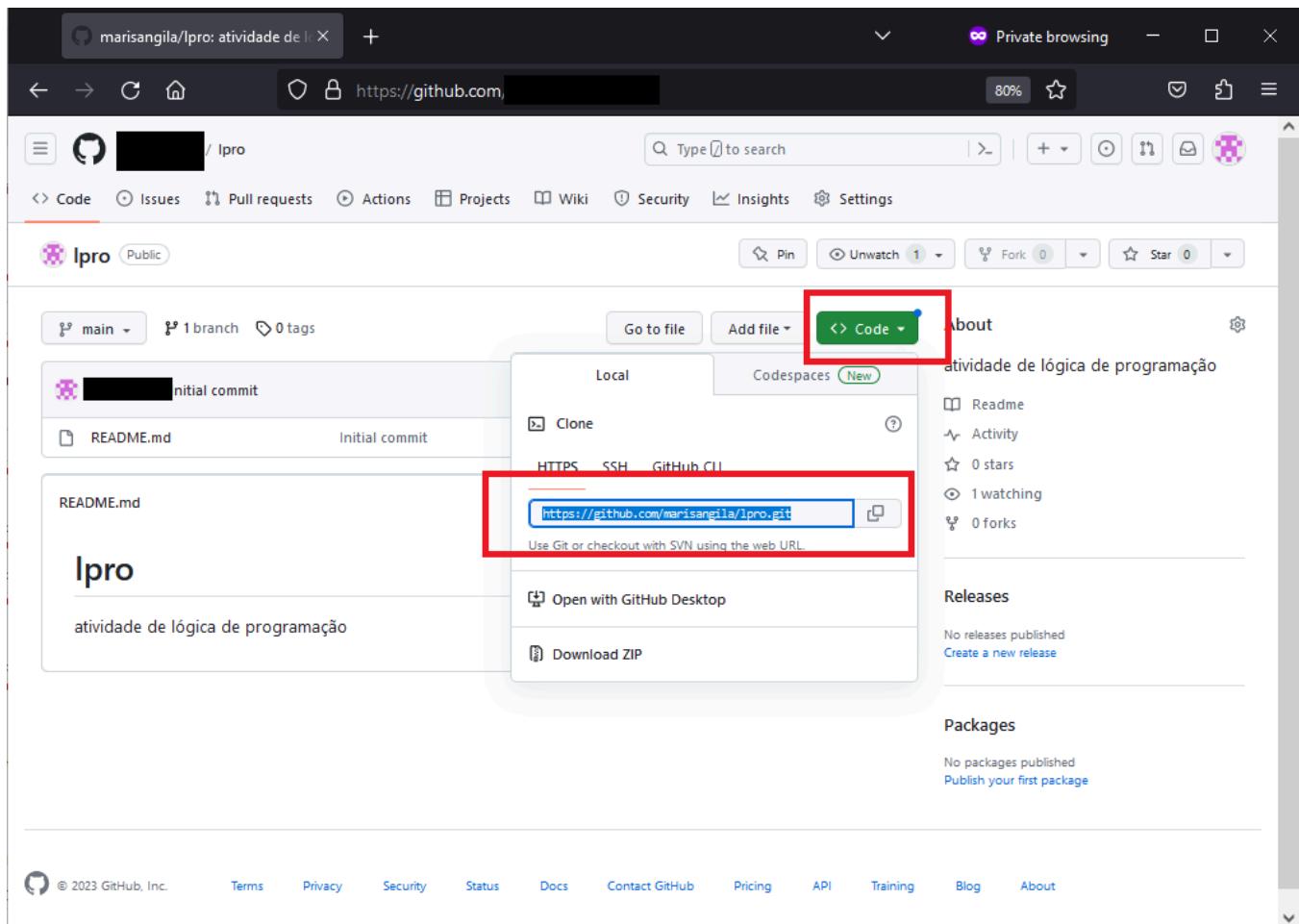
This will set `main` as the default branch. Change the default name in your [settings](#).

You are creating a public repository in your personal account.

**Create repository**

## Obter link de clone

Para obter o link de clone do seu repositório, selecione **Code** e depois copie o link.



marisangila/ipro: atividade de lógica de programação

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

ipro Public

main 1 branch 0 tags

Initial commit

README.md Initial commit

README.md

ipro

atividade de lógica de programação

about

Readme Activity 0 stars 1 watching 0 forks

Clone

Local Codespaces (New)

HTTPS SSH GitHub CLI

https://github.com/marisangila/ipro.git

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

Releases

No releases published Create a new release

Packages

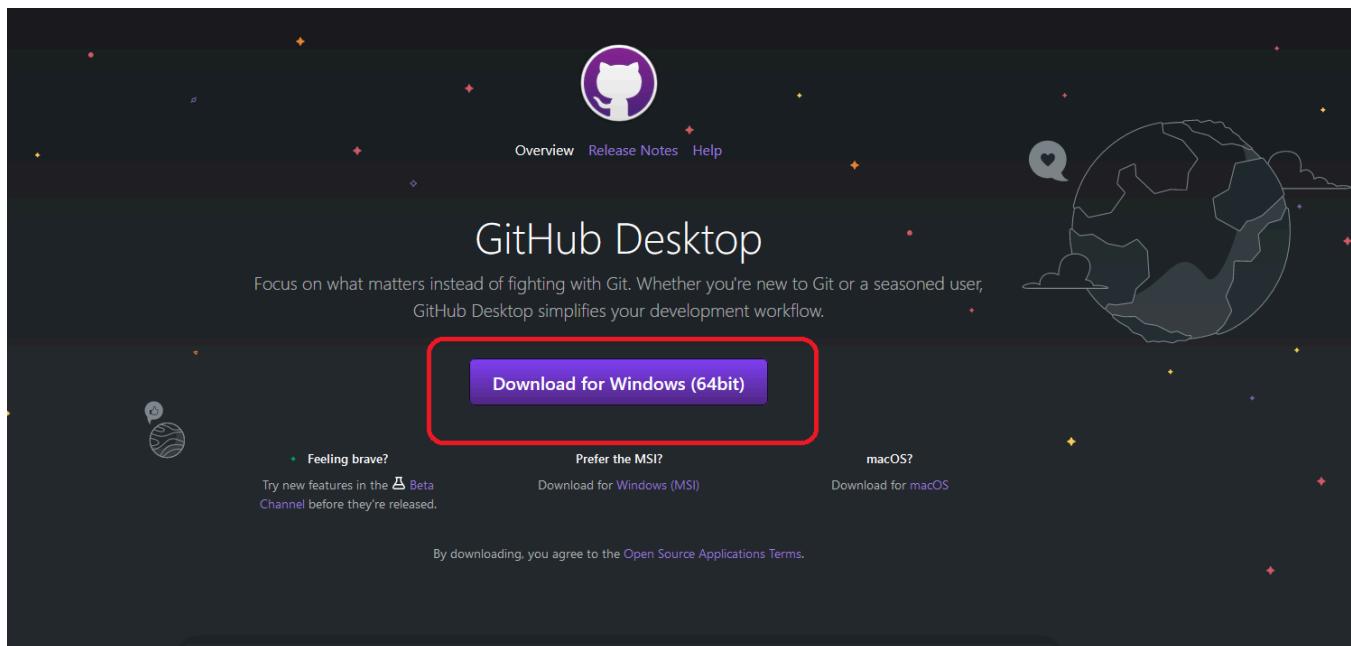
No packages published Publish your first package

© 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

## GitHub Desktop

Faça o download no link:

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



GitHub Desktop

Focus on what matters instead of fighting with Git. Whether you're new to Git or a seasoned user, GitHub Desktop simplifies your development workflow.

Download for Windows (64bit)

Feeling brave? Try new features in the [Beta Channel](#) before they're released.

Prefer the MSI? [Download for Windows \(MSI\)](#)

macOS? [Download for macOS](#)

By downloading, you agree to the [Open Source Applications Terms](#).

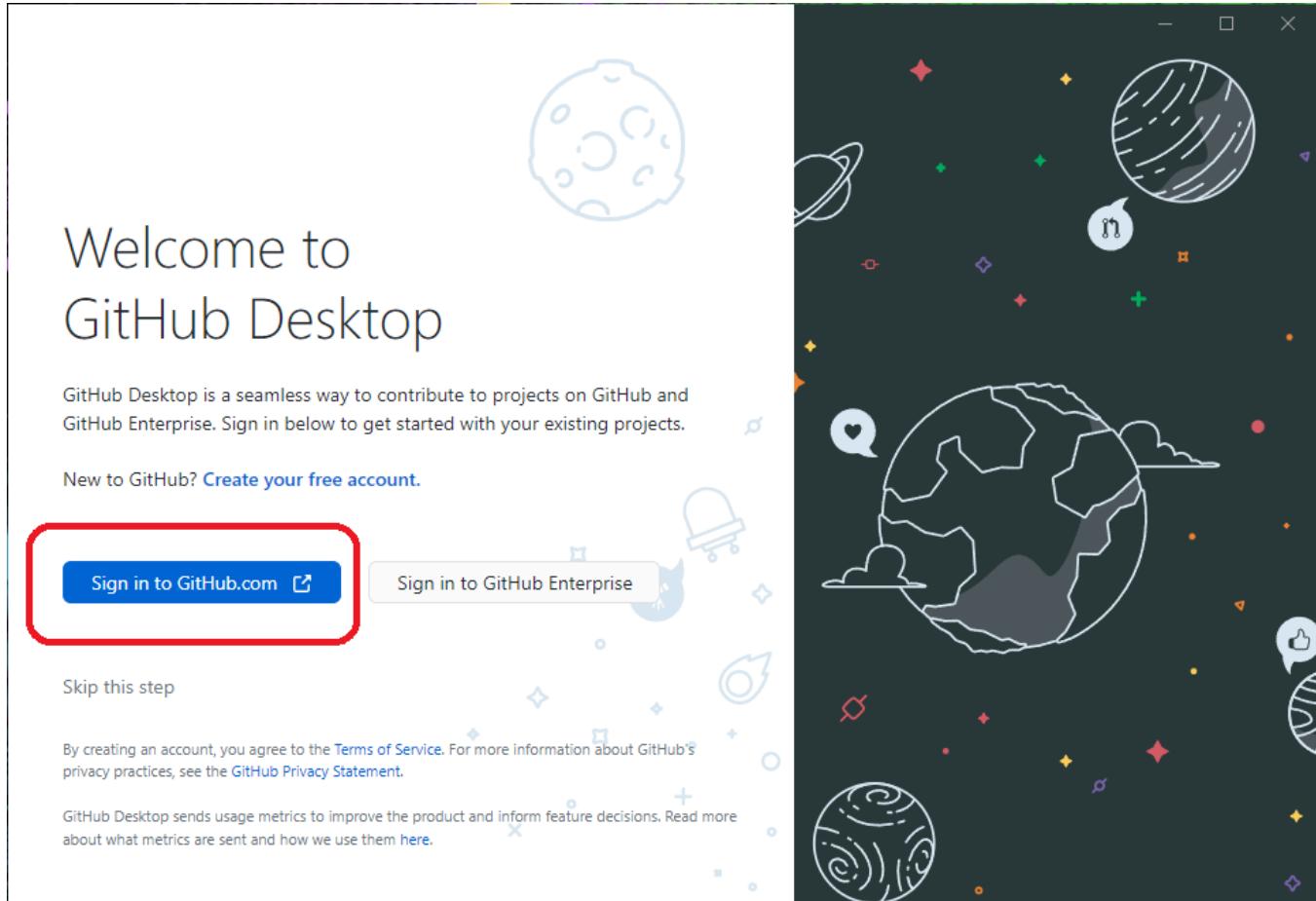
## Instale o GitHub Desktop

Processo de instalação comum do Windows!

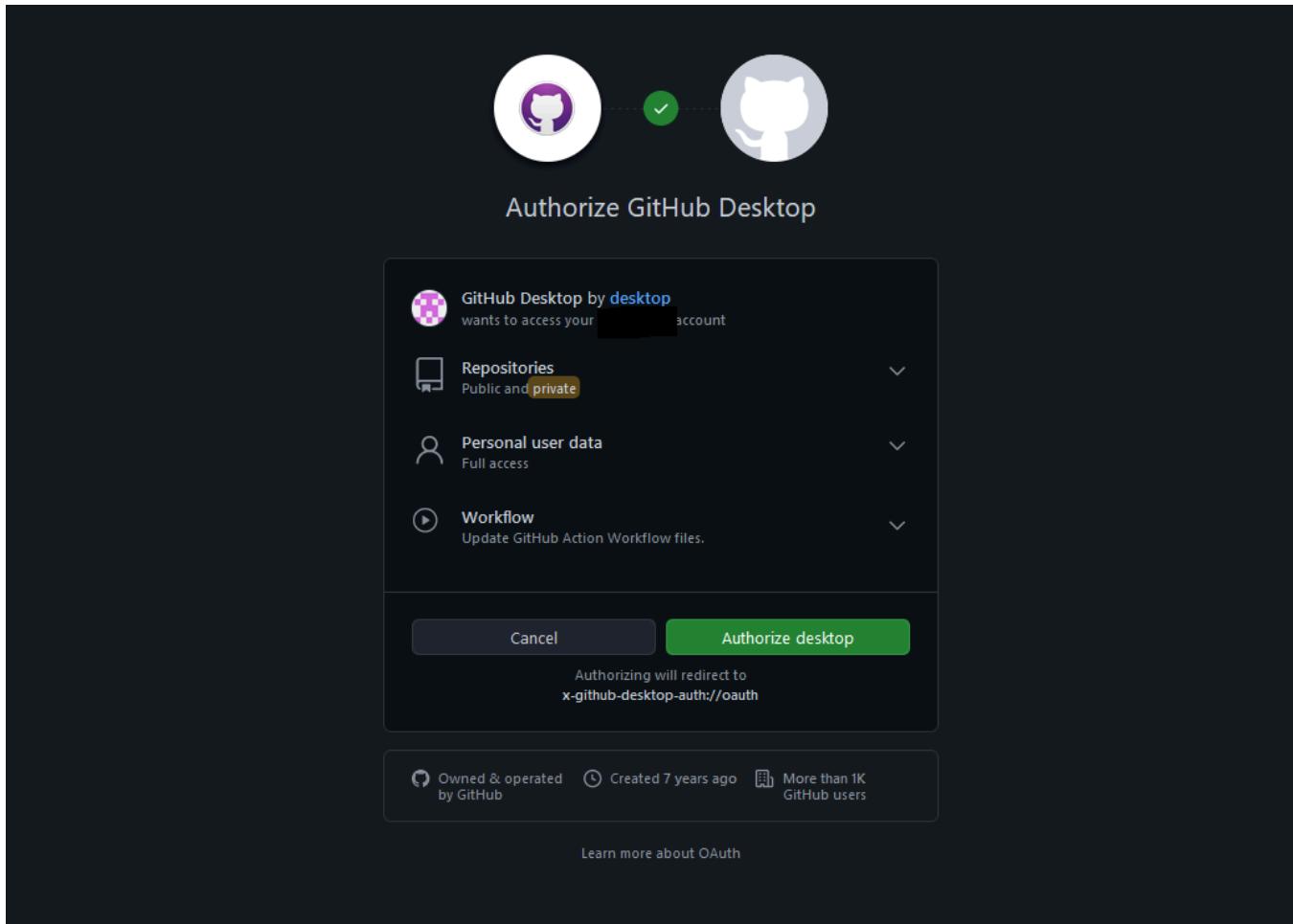
## Login no GitHub Desktop

Agora você precisa inserir e-mail e senha do GitHub.

Provavelmente irá pedir permissão para acessar sua conta.



Selecione seu e-mail e de as permissões:



## Clonar Repositório Existente

Agora você pode realizar um **clone** do repositório existente no seu GitHub.  
Use aquele link que você copiou.

# Configure Git

This is used to identify the commits you create. Anyone will be able to see this information if you publish commits.

- Use my GitHub account name and email address
- Configure manually

Name

Email

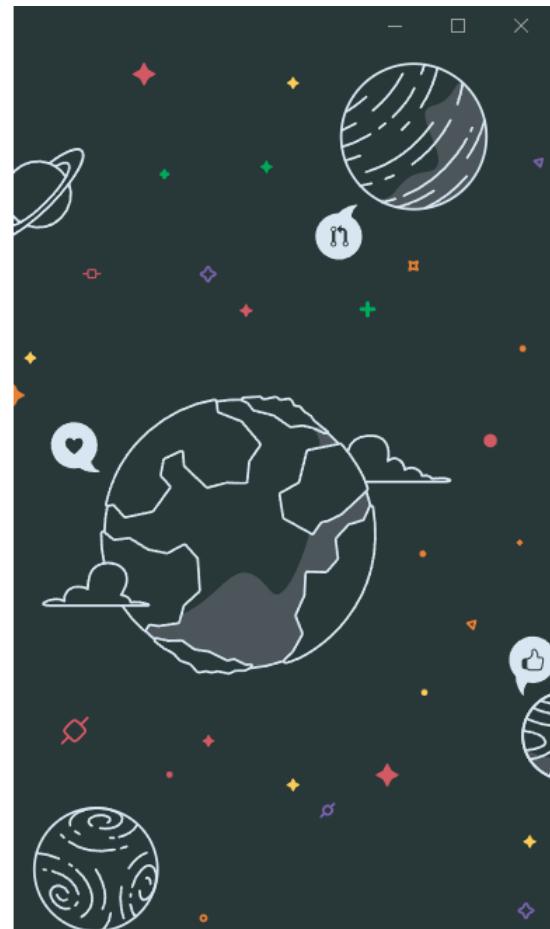
Finish

Cancel

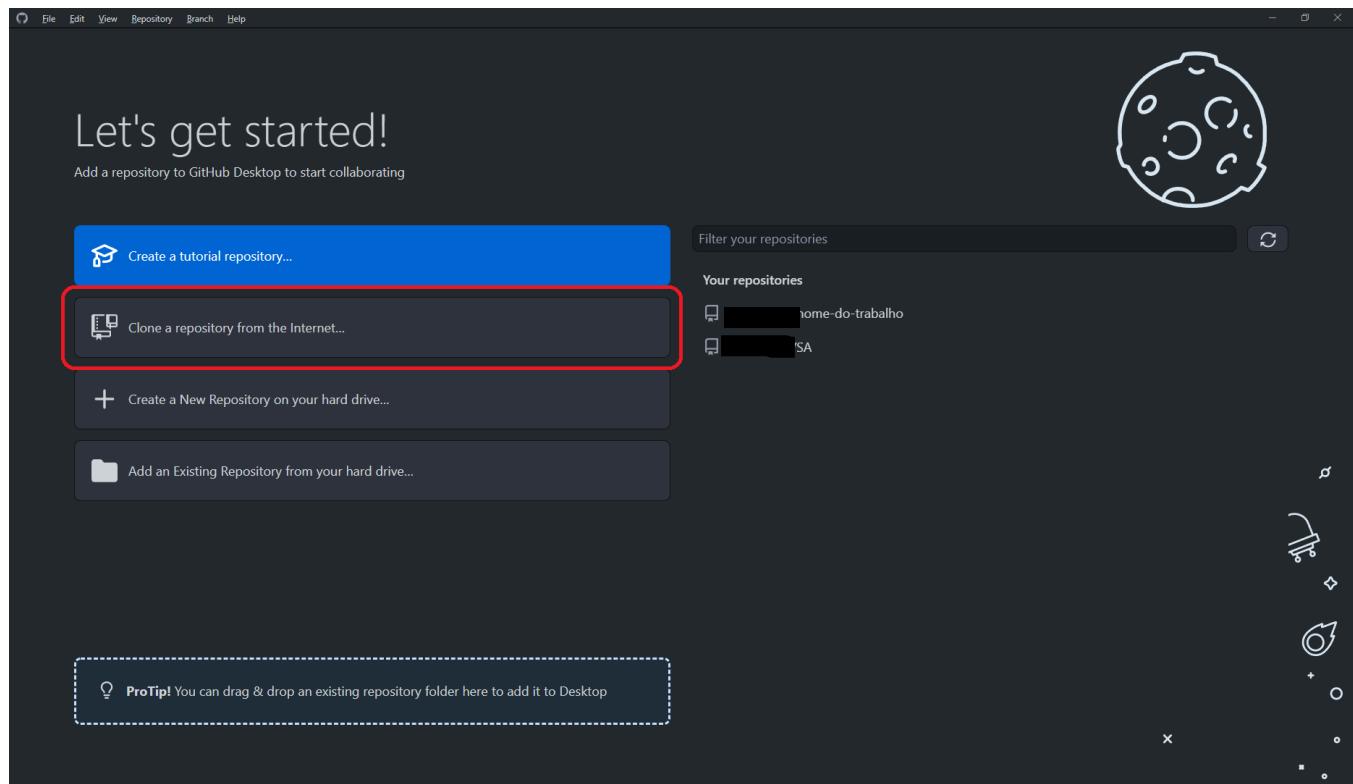
Example commit

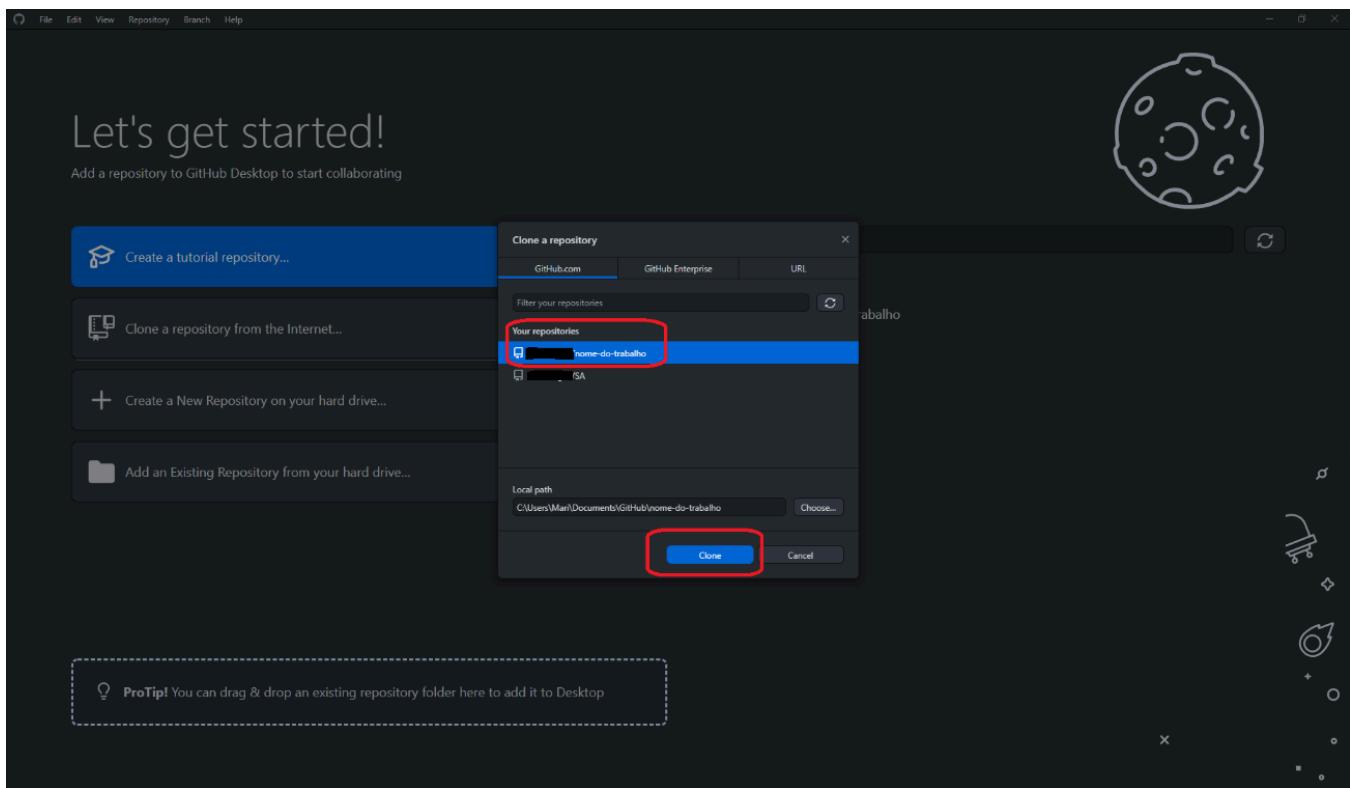
Fix all the things

 marisangila • 30 minutes ago



Selecione o repositório que você criou:

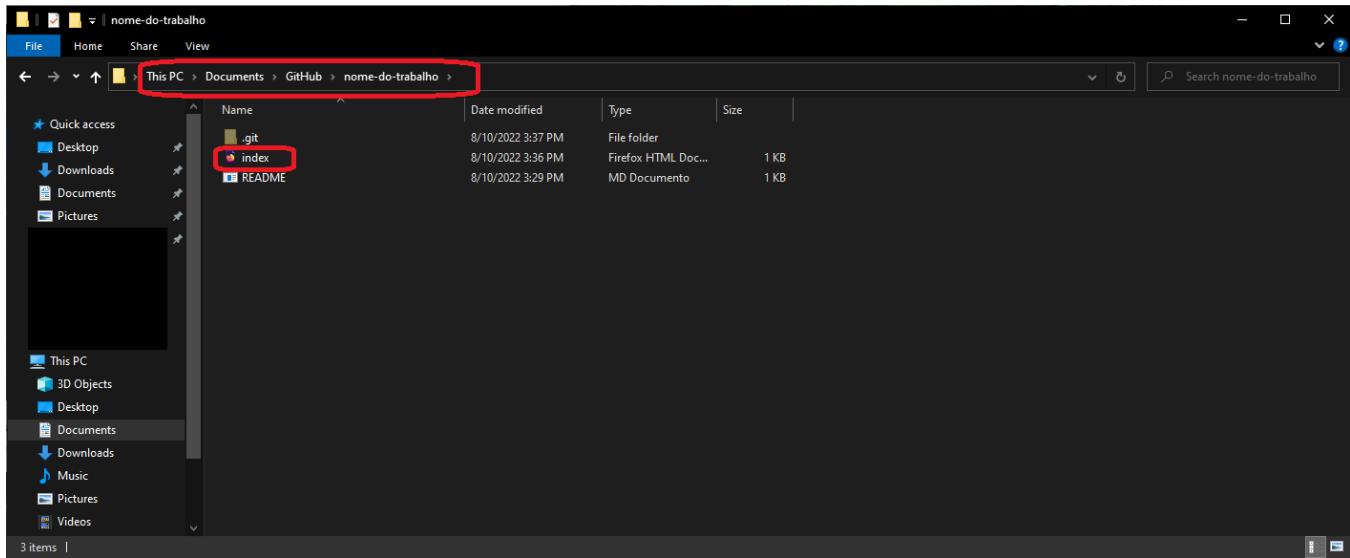




## Acesse a Pasta Local

Vá até a pasta onde seu repositório foi克隆ado  
(Normalmente dentro de Documentos/GitHub)

Crie um arquivo qualquer de teste (no exemplo foi criado o arquivo `index.html`).

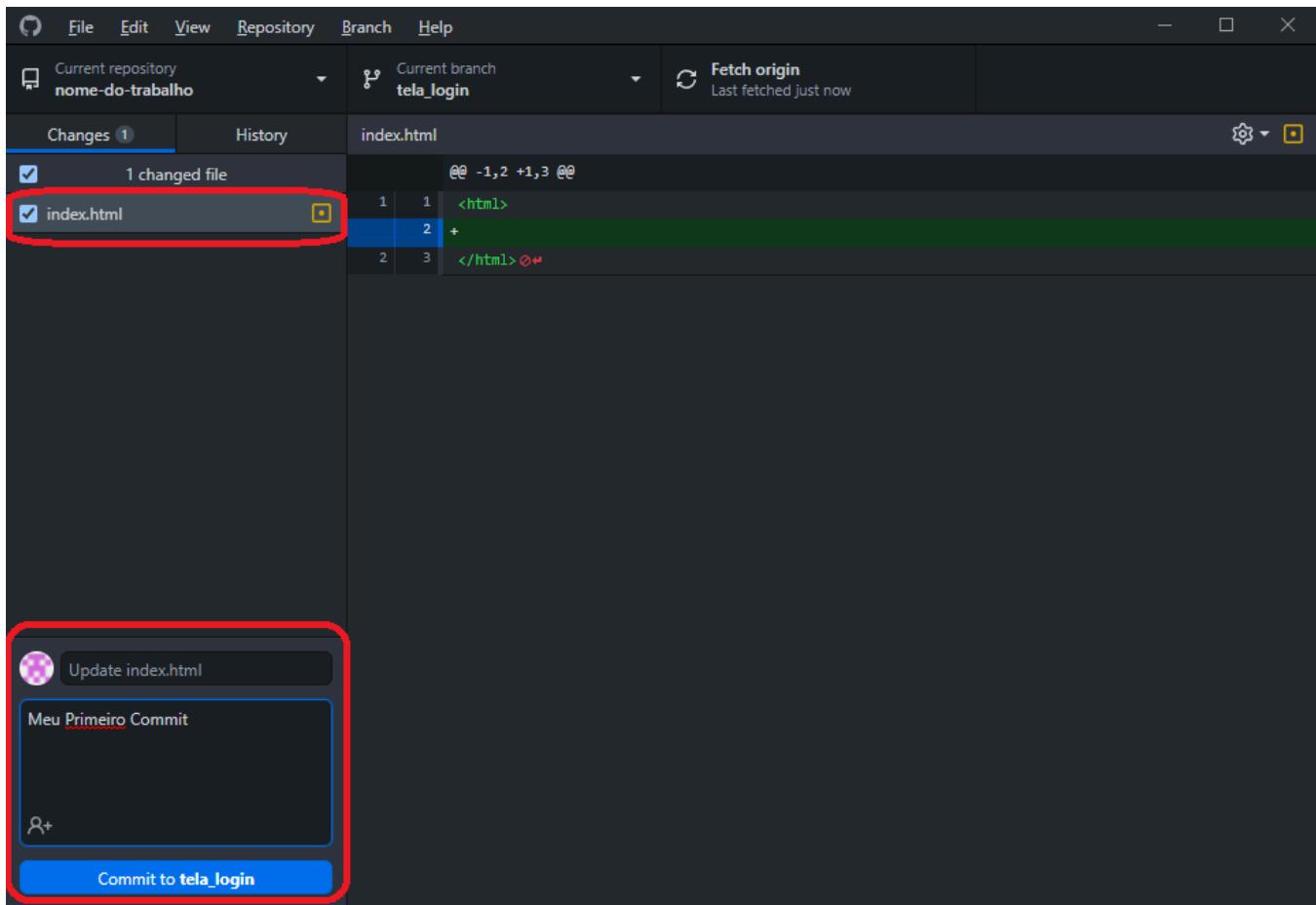


## Realizar um Commit

Agora é necessário realizar um **commit**.

1. Selecione os arquivos os quais você deseja realizar o commit;

2. Defina uma mensagem — lembre-se: use uma mensagem que sinalize **qual alteração foi feita**, isso facilita o mapeamento de alterações futuras.

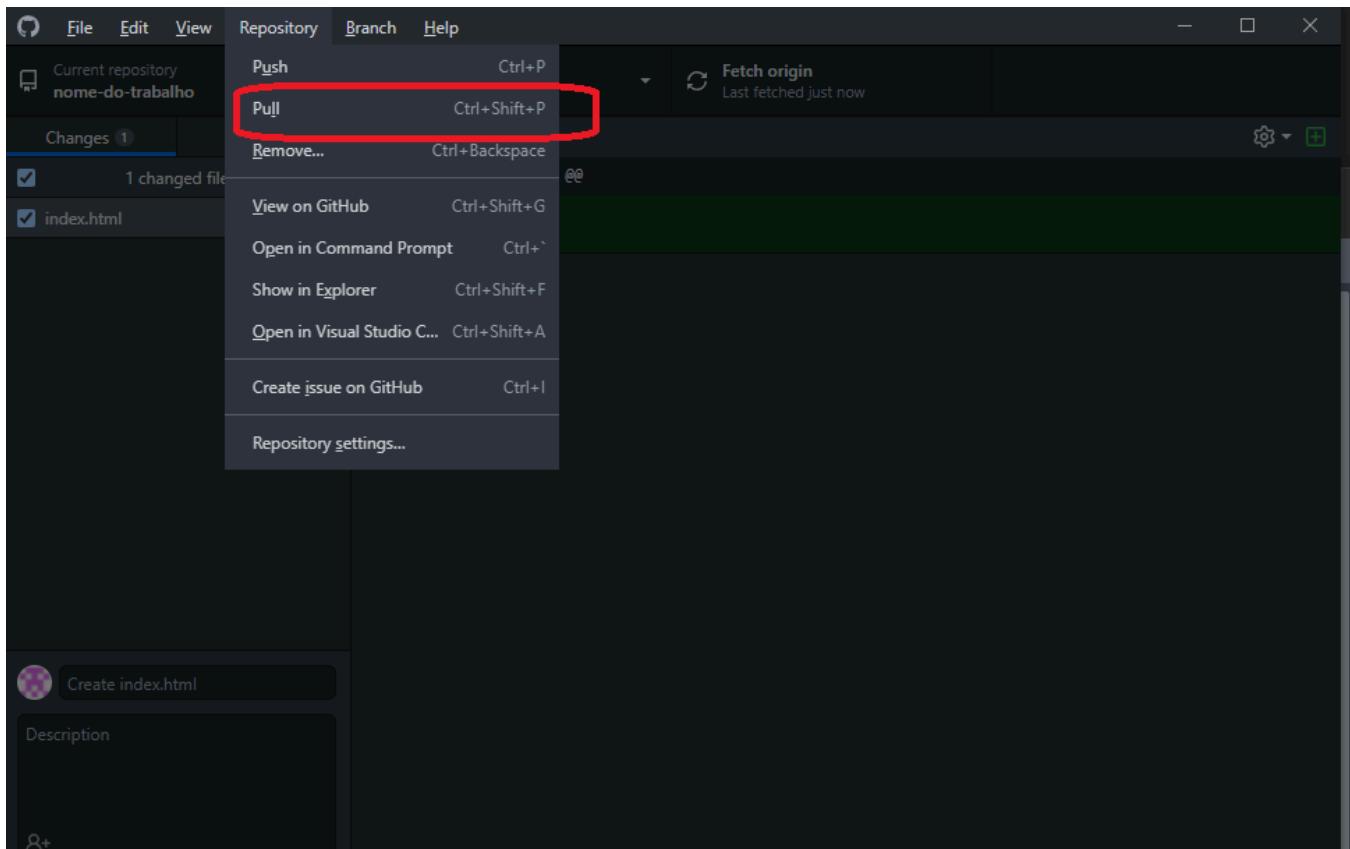


---

## Atualizar o código local com Git Pull (se necessário).

Antes de enviar o novo arquivo é necessário garantir que os arquivos em seu computador estão com as **últimas alterações existentes** no repositório do GitHub.

Para isso, use **GIT PULL**.

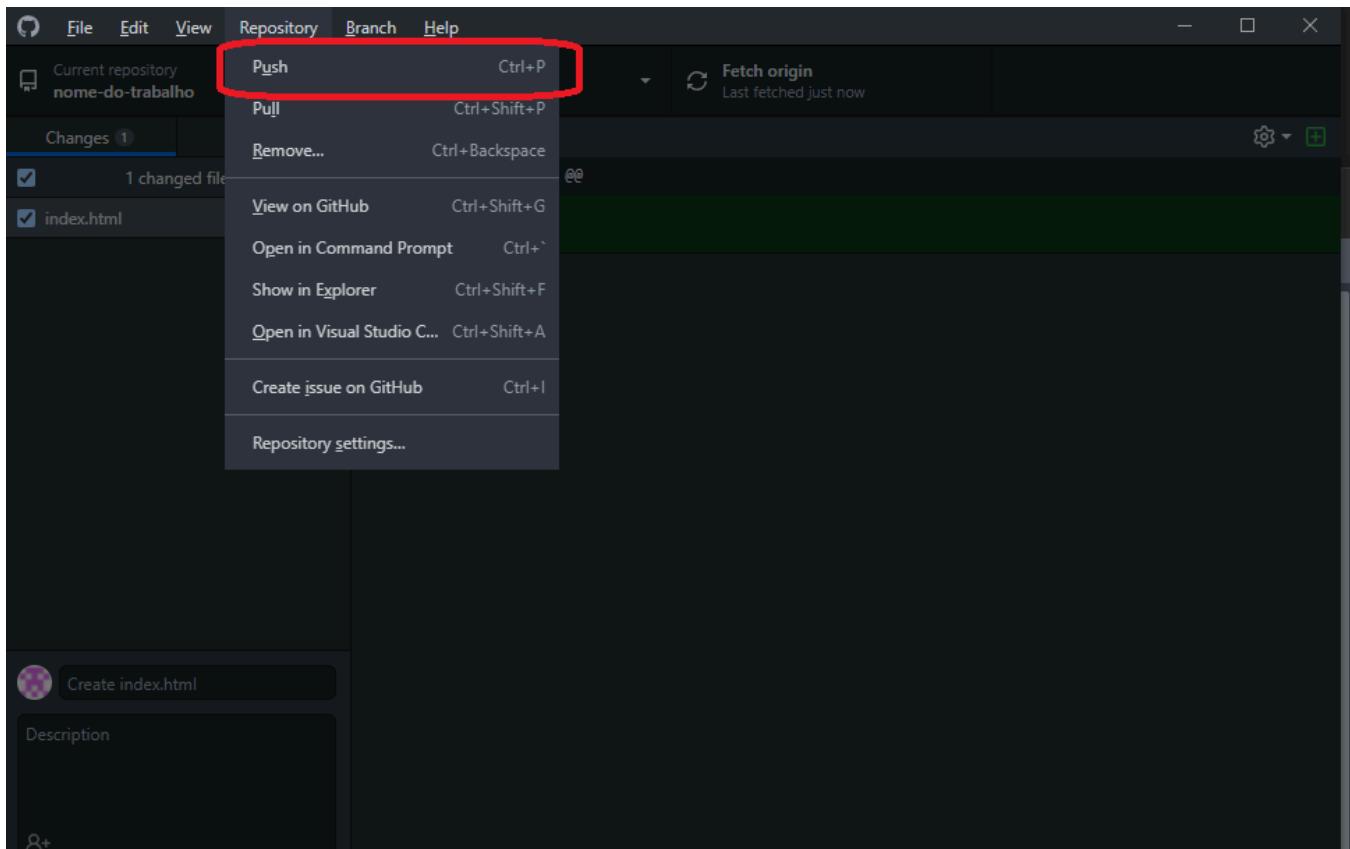


## Enviar alterações com Git Push

Finalmente, é possível enviar suas alterações para o repositório no GitHub.

Assim, suas modificações ficam salvas online.

Para isso, use **GIT PUSH**.



## Alterações no GitHub

Agora, suas alterações apareceram no GitHub!

