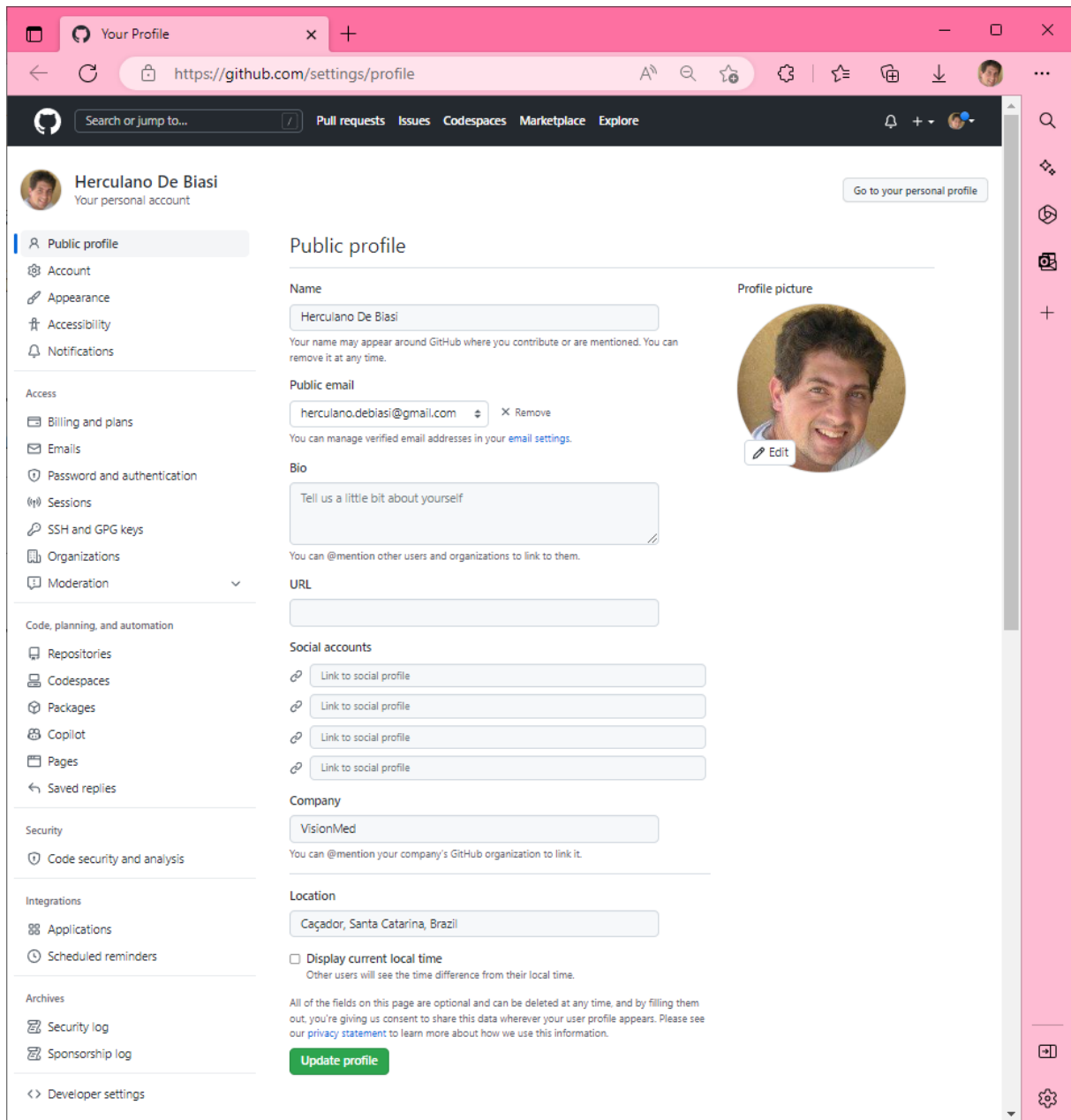


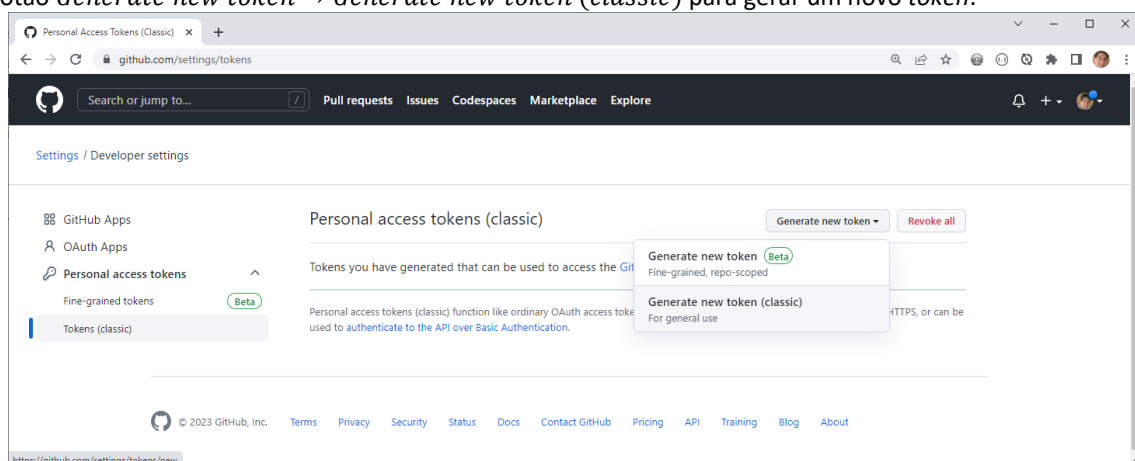
# 1 GitHub

O processo a seguir irá gerar um *token*. Este *token* funciona como uma senha para que você, usuário desenvolvedor, consiga se autenticar no GitHub quando estiver utilizando o STS.

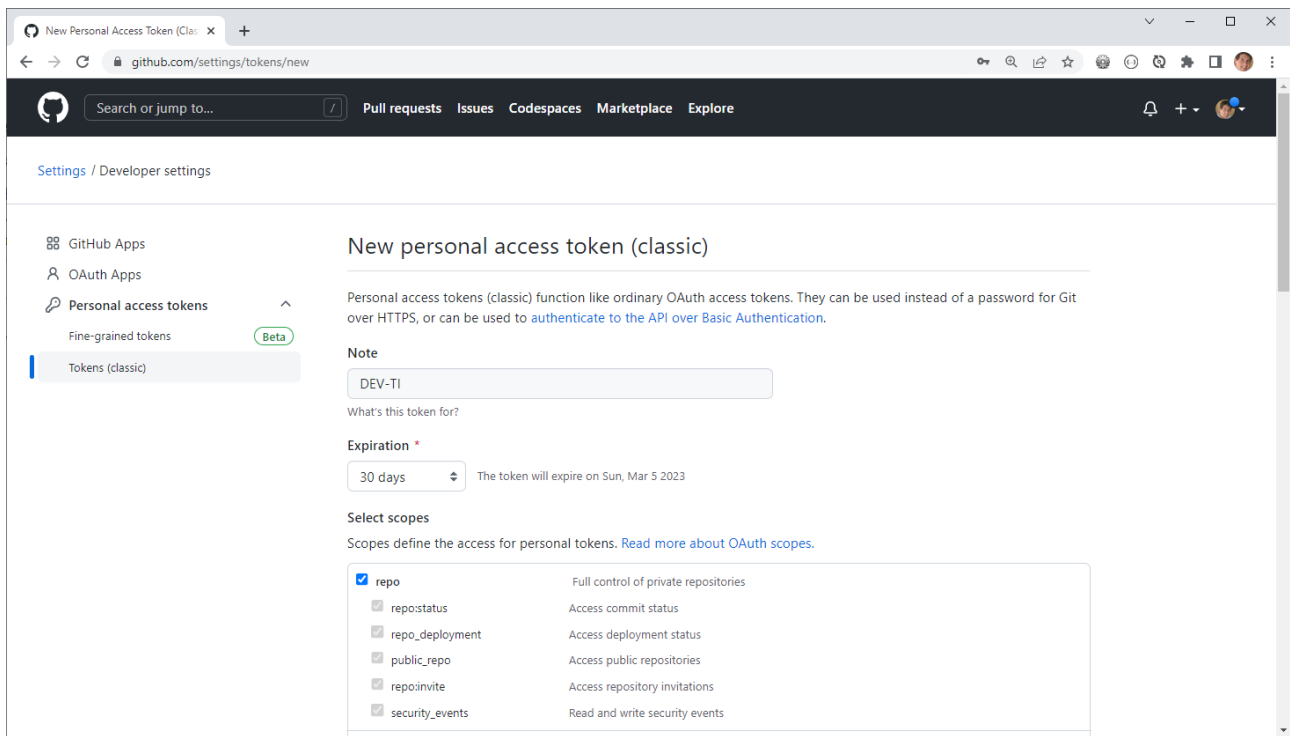
Para fazer isso primeiramente entre na sua conta no [GitHub](#) e vá para a configurações do [perfil](#). Role a página até o final para entrar na opção [Developer settings](#).



Clique no botão *Generate new token* → *Generate new token (classic)* para gerar um novo *token*.



Forneça um nome qualquer no campo *Note* e marque a opção *repo*.



New Personal Access Token (Classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

**Note**

DEV-TI

What's this token for?

**Expiration \***

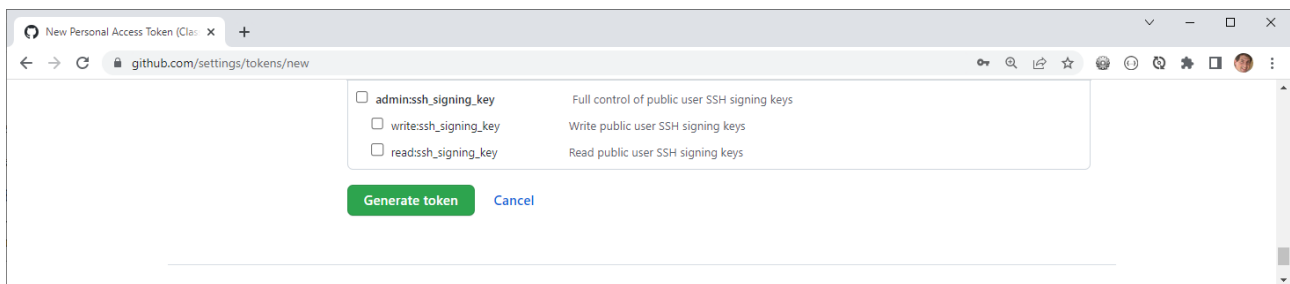
30 days The token will expire on Sun, Mar 5 2023

**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

- ☒ **repo** Full control of private repositories
  - ☒ repo\_status Access commit status
  - ☒ repo\_deployment Access deployment status
  - ☒ public\_repo Access public repositories
  - ☒ repo\_invite Access repository invitations
  - ☒ security\_events Read and write security events

Role a página até o final e clique em *Generate token*.



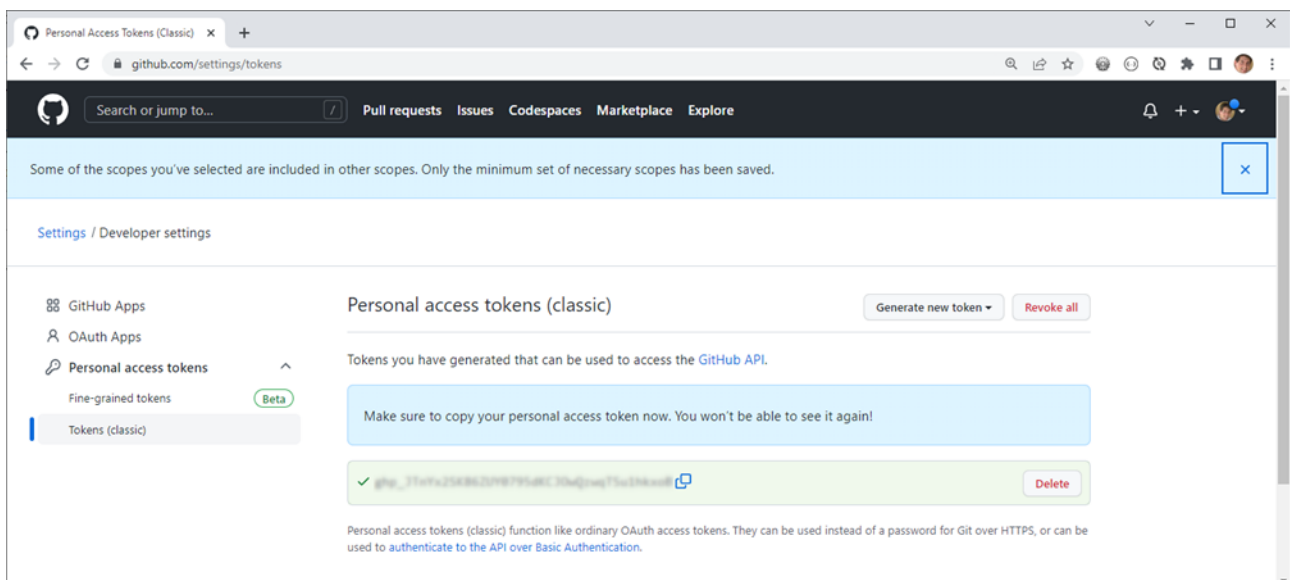
☐ admin:ssh\_signing\_key Full control of public user SSH signing keys

☐ write:ssh\_signing\_key Write public user SSH signing keys

☐ read:ssh\_signing\_key Read public user SSH signing keys

**Generate token** Cancel

O *token* irá aparecer somente uma vez, se ele for perdido, terá que ser gerado novamente. Clique no botão copiar e armazene-o em outro local, pois será necessário utilizá-lo novamente mais tarde. Por questões de segurança o *token* não deve ser divulgado ou compartilhado, pois funciona como uma senha pessoal do usuário.



Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved.

Settings / Developer settings

**Personal access tokens (classic)** **Generate new token** **Revoke all**

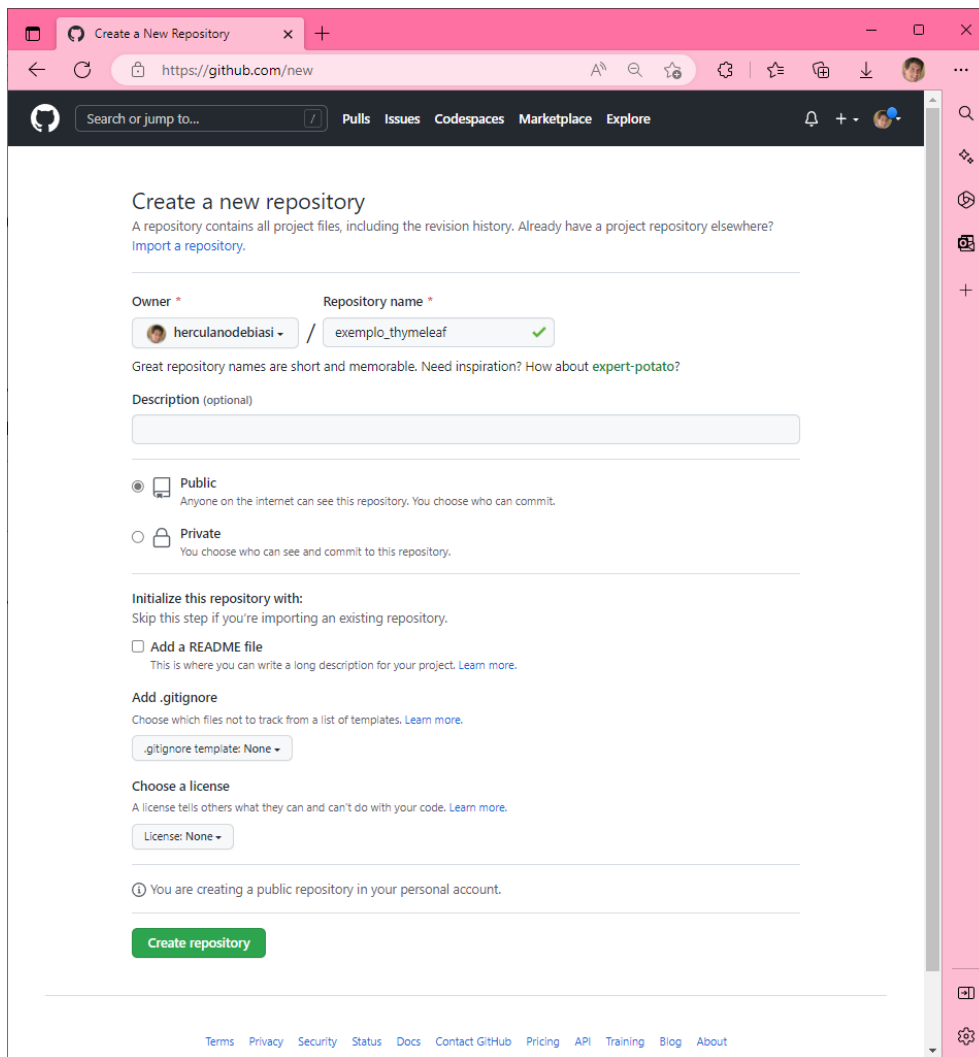
Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp\_1TnFx25X8B6Z0Y8P75u8K30u4GwqP5u3Hkxw0f **Delete**



Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Clique neste [link](#) para criar um novo repositório. Preencha com o nome do repositório e clique no botão *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.](#)

Owner <sup>\*</sup>  herculanodebiasi / Repository name <sup>\*</sup>  

Great repository names are short and memorable. Need inspiration? How about **expert-potato**?

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.](#)

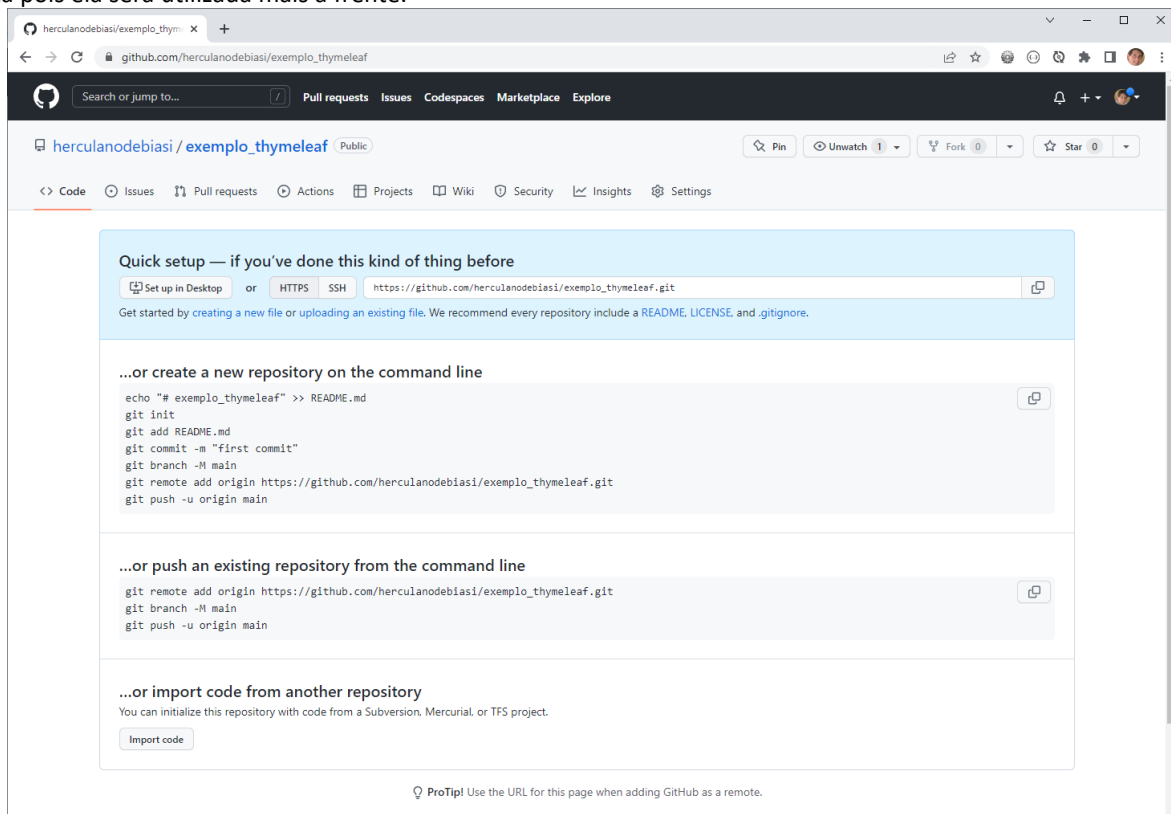
License: **None**

 You are creating a public repository in your personal account.

[Create repository](#)

[Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

A página abaixo deverá aparecer mostrando que o repositório **remoto** foi criado no GitHub assim como várias instruções sobre como criar um repositório local e conectá-lo ao repositório remoto utilizando o cliente Git em linha de comando. Mantenha esta tela aberta pois ela será utilizada mais à frente.






herculanodebiasi/exemplo\_thymeleaf

Public

Pin Unwatch 1 Fork 0 Star 0

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

Quick setup — if you've done this kind of thing before

 or **HTTPS**   

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# exemplo_thymeleaf" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/herculanodebiasi/exemplo_thymeleaf.git
git push -u origin main
```


...or push an existing repository from the command line

```
git remote add origin https://github.com/herculanodebiasi/exemplo_thymeleaf.git
git branch -M main
git push -u origin main
```

...or import code from another repository

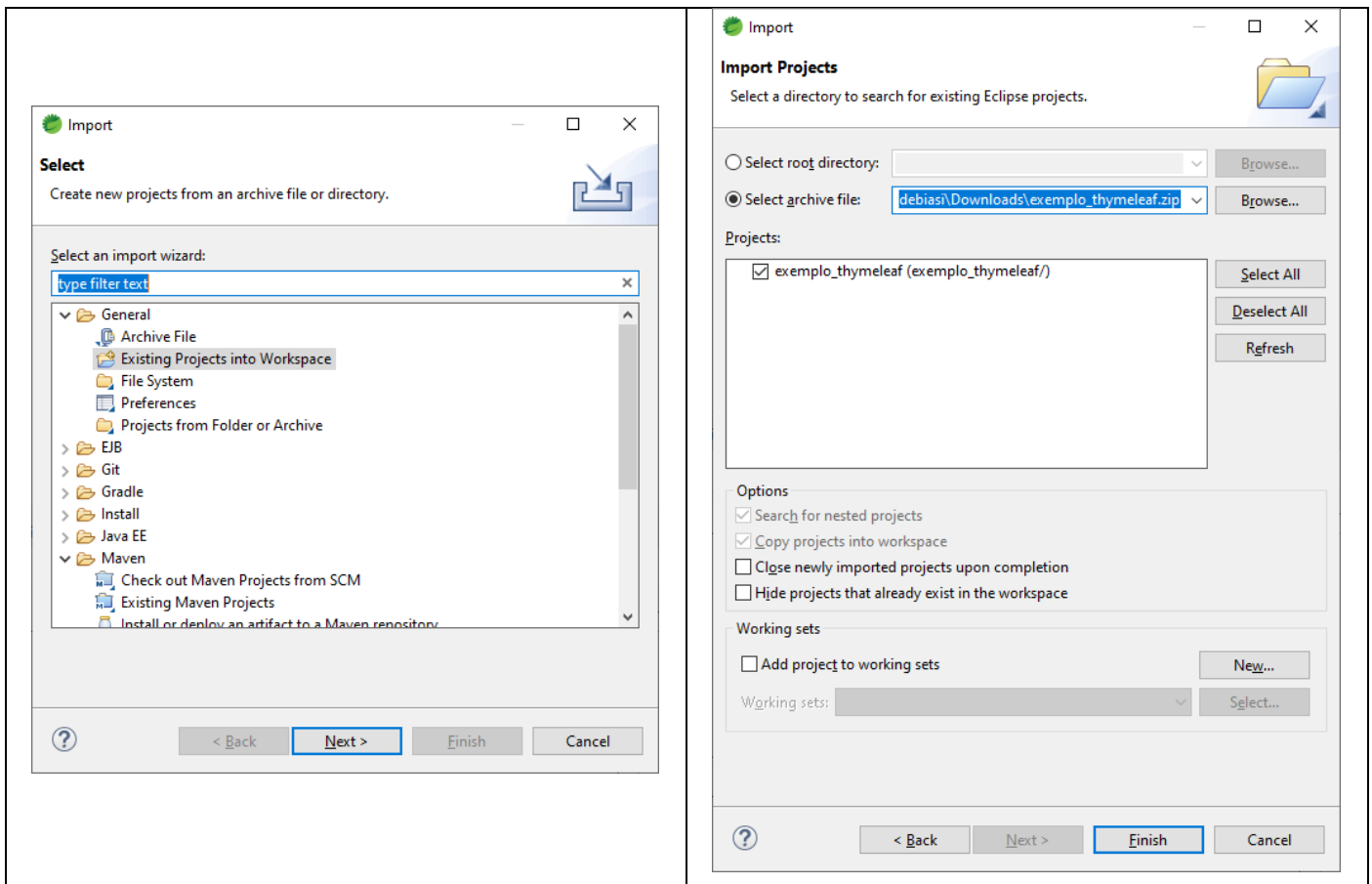
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

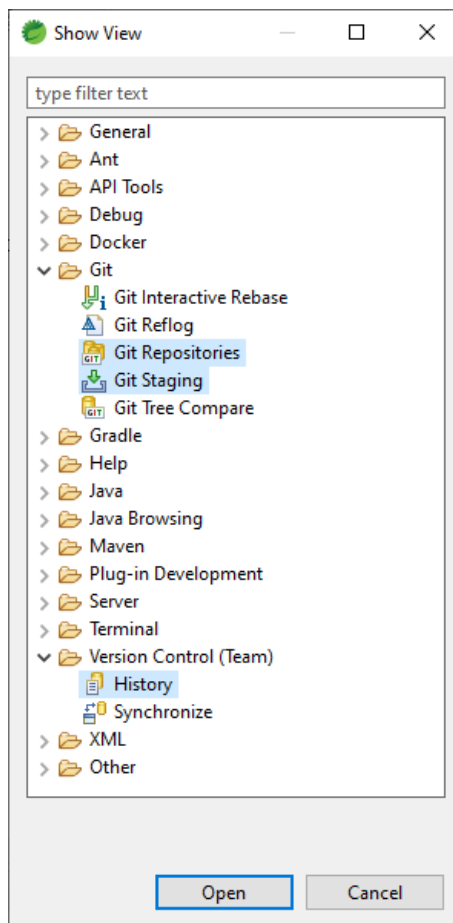
 **ProTip!** Use the URL for this page when adding GitHub as a remote.

## 2 STS

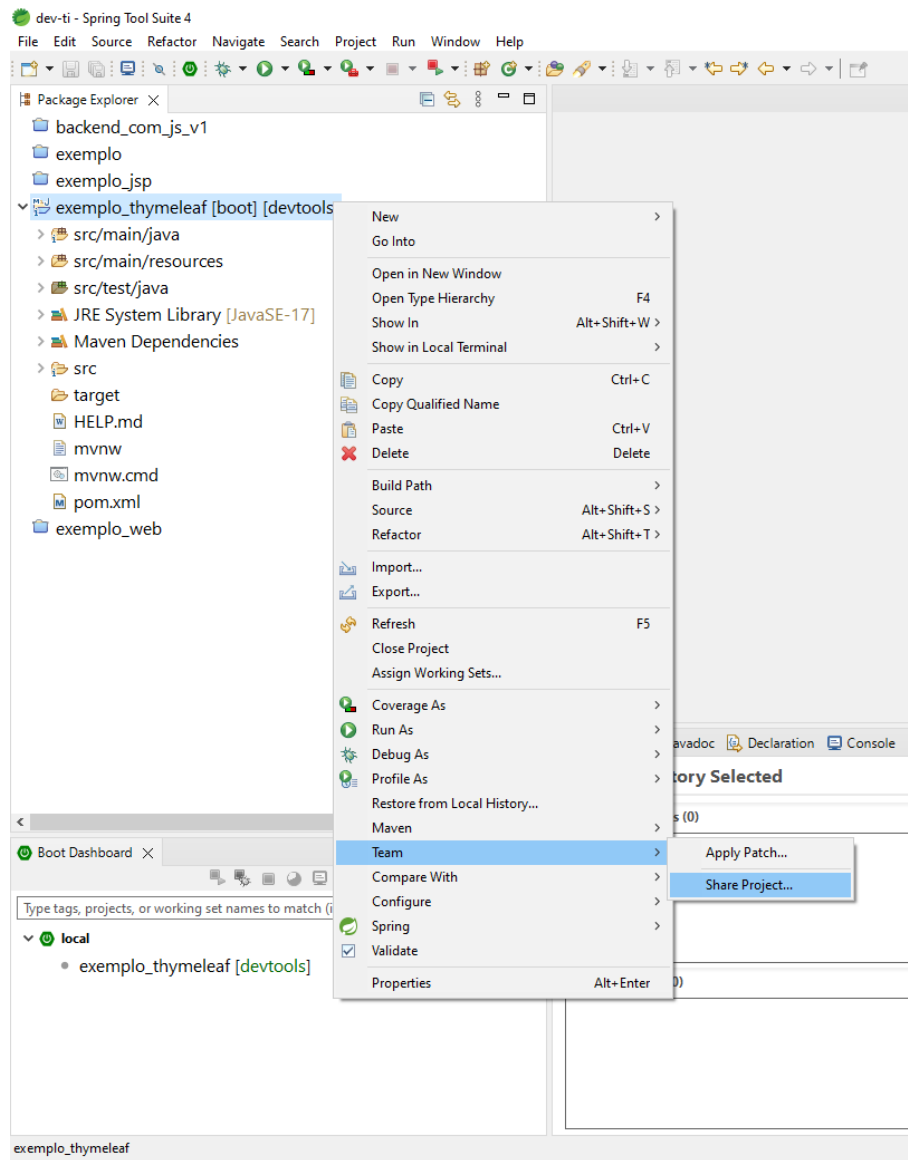
No moodle, baixe o projeto '[exemplo thymeleaf](#)'. Se este projeto já existir no *workspace*, apague-o. Importe o projeto no STS indo ao menu *File* → *Import* e seguindo as telas abaixo:



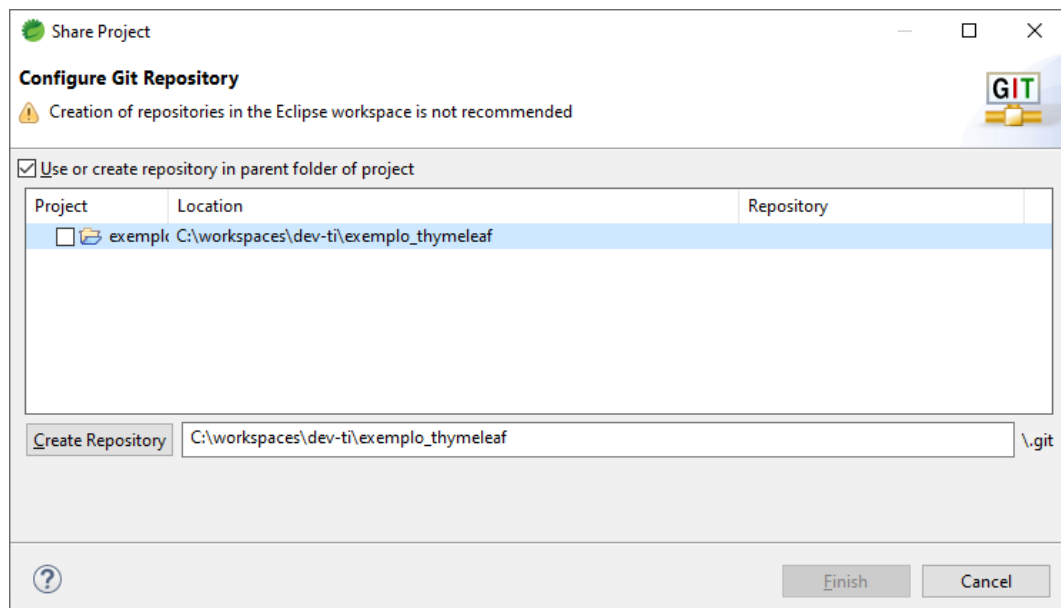
No menu *Window* → *Show View* selecione as 3 janelas abaixo (para isso segure a tecla *Ctrl*):



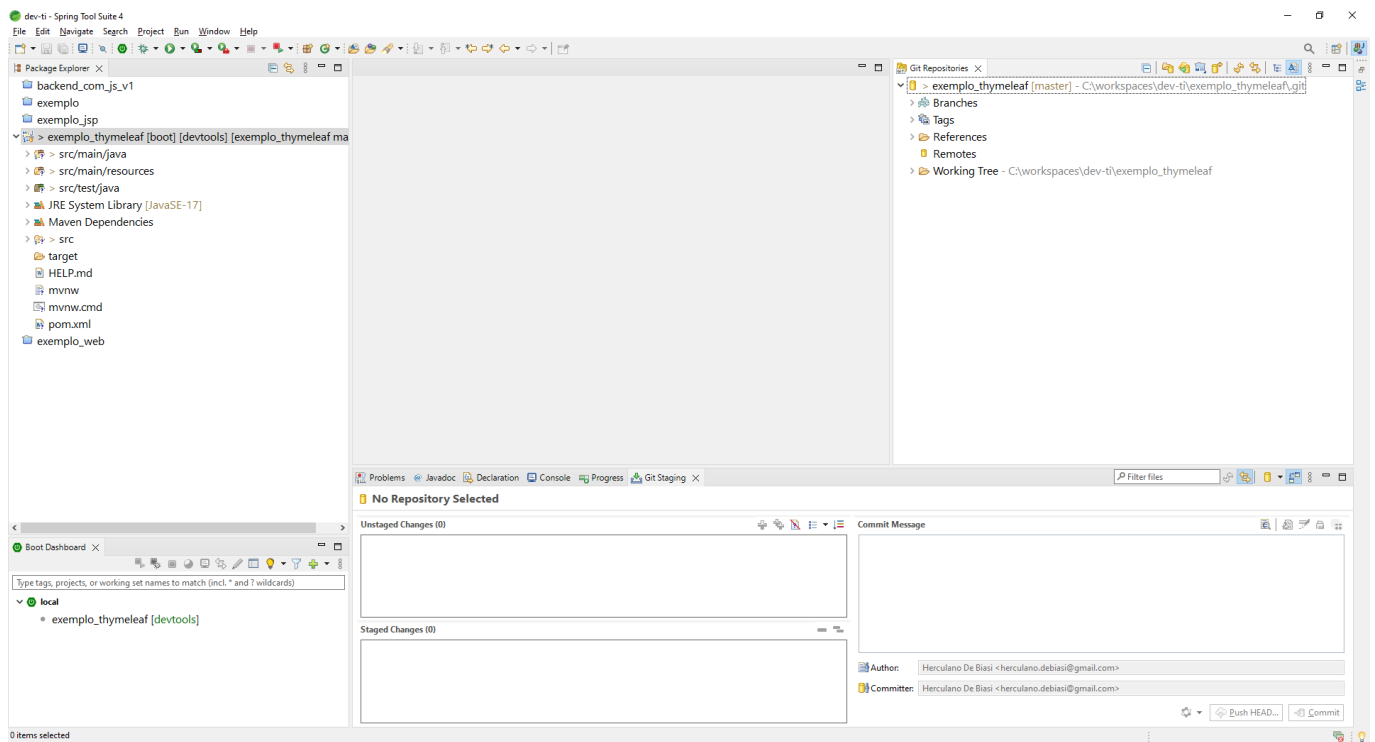
Para criar um repositório local, sob controle do Git, abra o projeto recém importado. A seguir, clique com o botão direito sobre ele e escolha a opção *Team* → *Share Project* ...



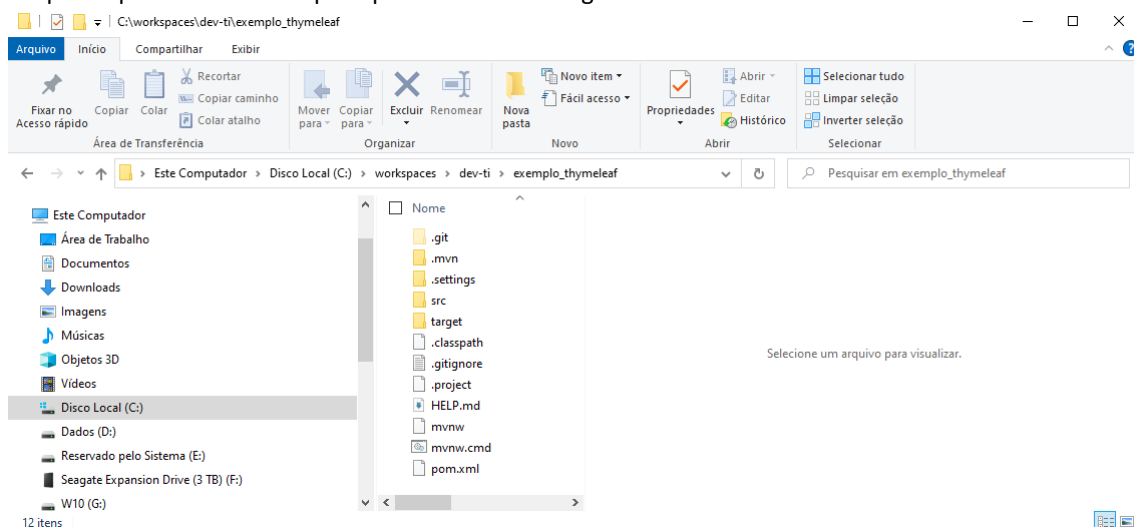
Selecione a opção *Use or create repository in parent folder of projet*. A seguir clique no botão *Create Repository* e depois em *Finish*.




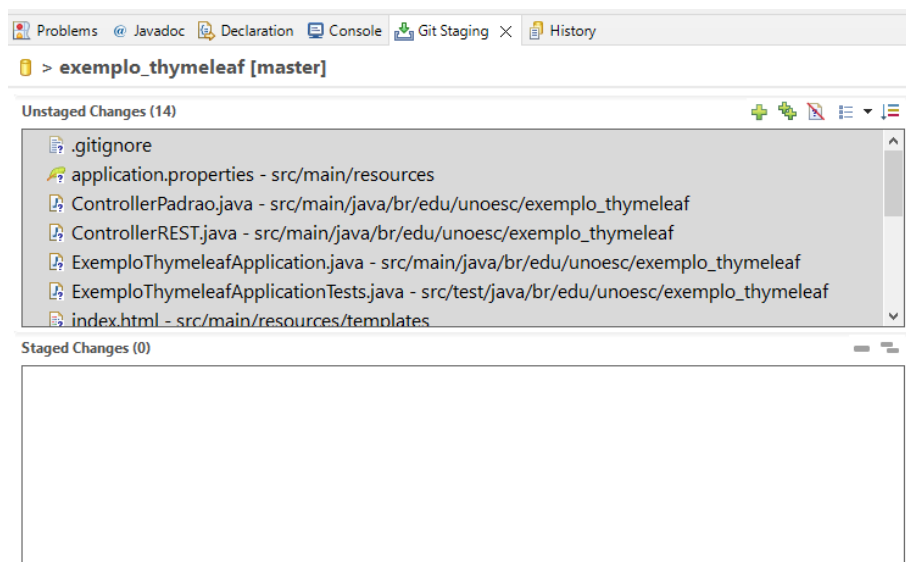
Com isso o repositório deve ter sido criado dentro do projeto atual no *workspace*.



No Windows Explorer pode-se verificar que a pasta de controle '.git' foi criada:



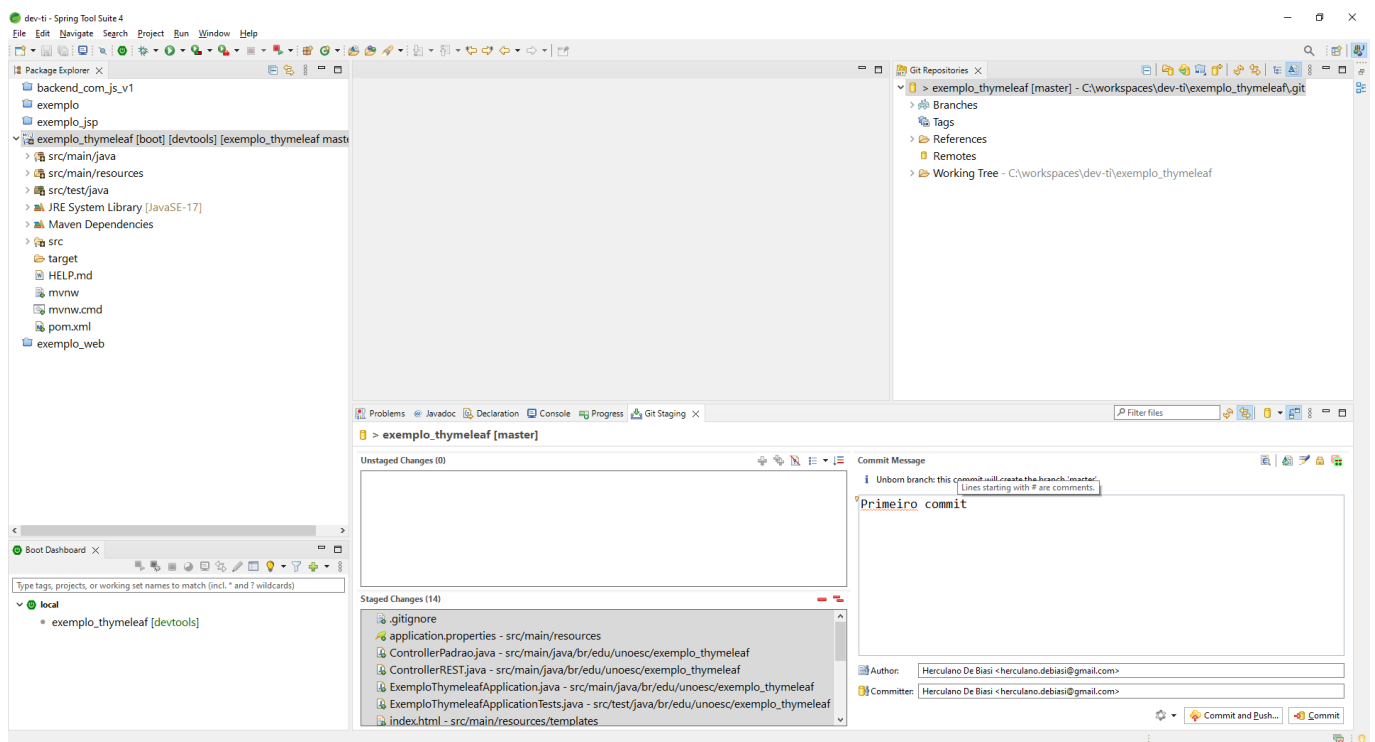
Abra a janela *Git Staging*. Pode-se notar que os arquivos do projeto estão na área *unstaged changes*. Para que as alterações sejam efetivadas no próximo *commit* é preciso que os arquivos sejam movidos para a área de *staging*. Para fazer isso clique no botão . Caso não funcione, experimente fechar e abrir novamente o STS.



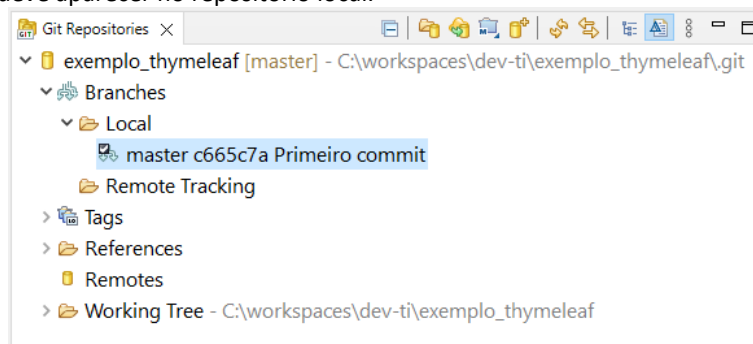
Arquivos adicionados na área de *staging* a fim que as alterações sejam efetivadas no próximo *commit*.



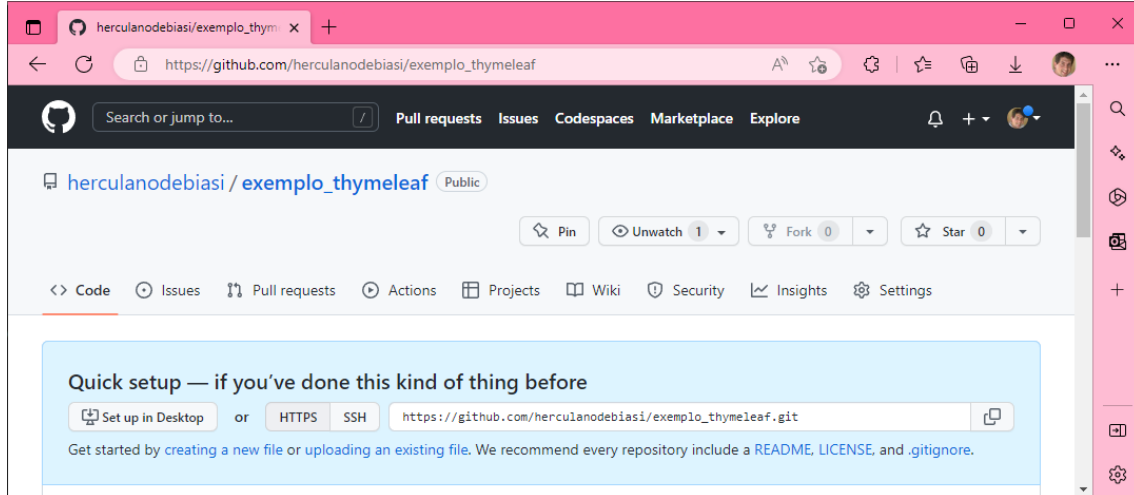
Escreva a mensagem 'Primeiro commit' na área *commit message* . Clique no botão *Commit ...*



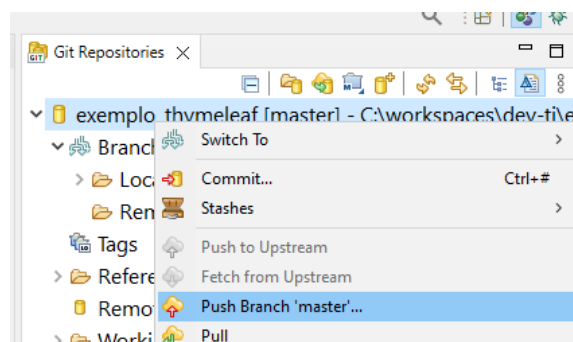
Ao fazer isso este *commit* já deve aparecer no repositório local.



O projeto local ainda não foi ligado ao repositório no GitHub. Para fazer isso volte ao GitHub e copie o *link* do repositório.



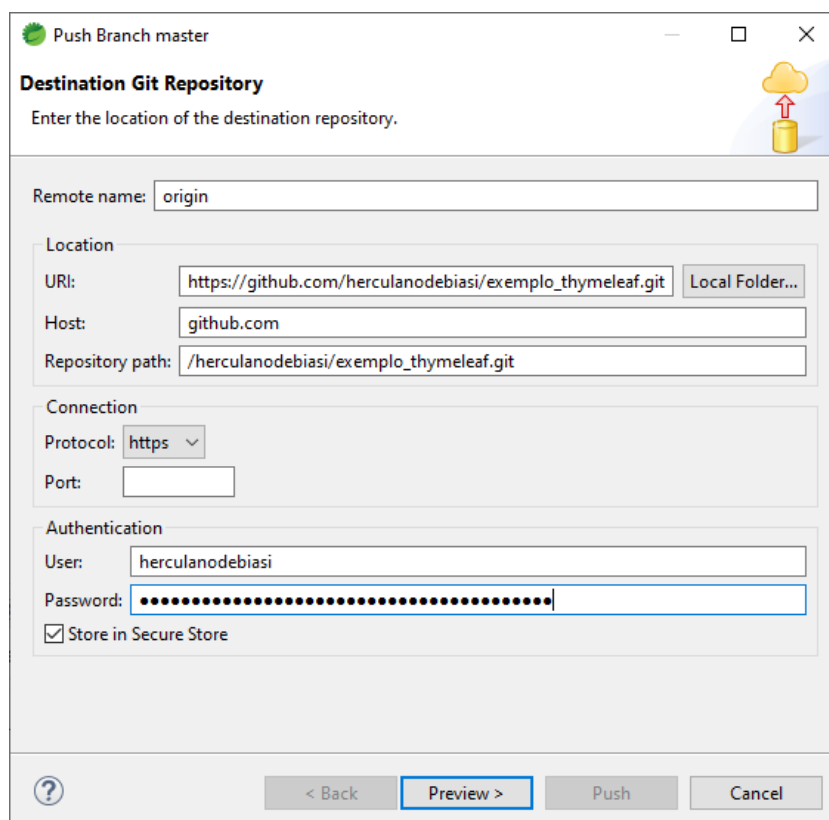
Clique agora sobre o projeto e escolha a opção *Push Branch 'master'* ...



Os seguintes dados deverão aparecer na janela:

- URI: [https://github.com/herculanodebiasi/exemplo\\_thymeleaf.git](https://github.com/herculanodebiasi/exemplo_thymeleaf.git)
- Host: github.com
- Repository path: /herculanodebiasi/exemplo\_thymeleaf.git

No campo *Password* insira o *token* gerado no início deste roteiro.



Clique em *Preview*.



**Push Branch master**

**Push to branch in remote**  
Select a remote and the name the branch should have in the remote.

Source:  
master d7571a3 Primeiro commit

Destination:  
Remote: origin: https://github.com/herculanodebiasi/exemplo\_thymeleaf.git  
Branch: master

☒ Configure upstream for push and pull

When pulling: Merge

☐ Force overwrite branch in remote if it exists and has diverged

Show [advanced push dialog](#)

< Back Preview > Push Cancel

Clique novamente em *Preview*. Caso solicitado entre novamente com o usuário e *token*.

**Login**

Repository https://github.com/herculanodebiasi/exemplo\_thymeleaf.git

User herculanodebiasi

Password .....

Store in Secure Store ☒

Log in Cancel

Clique em *Push*.

**Push Branch master**

**Push Confirmation**  
Confirm following expected push result.

master → master [new branch]

Message Details

Repository  
[https://github.com/herculanodebiasi/exemplo\\_thymeleaf.git](https://github.com/herculanodebiasi/exemplo_thymeleaf.git)

☐ Cancel push if result would be different than above because of changes on remote  
☐ Show dialog with result only when it is different from the confirmed result above

< Back Preview > Push Cancel

Clique em *Close*.

**Push Results: https://github.com/herculanodebiasi/exemplo\_thymeleaf.git**

**Pushed to https://github.com/herculanodebiasi/exemplo\_thymeleaf.git**

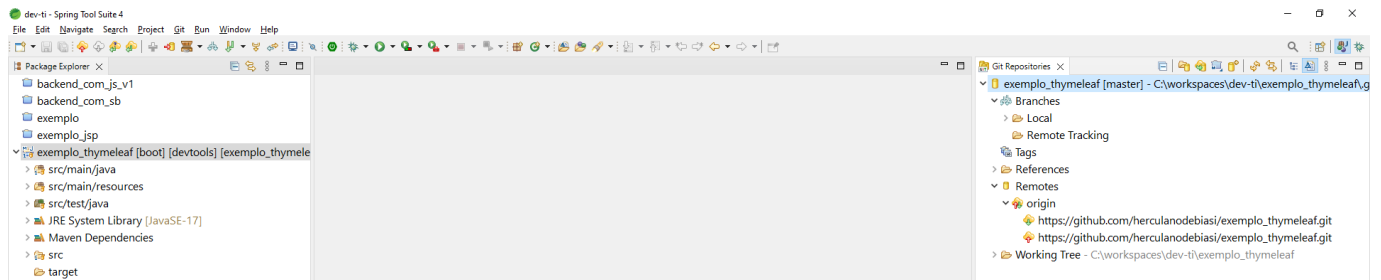
master → master [new branch]

Message Details

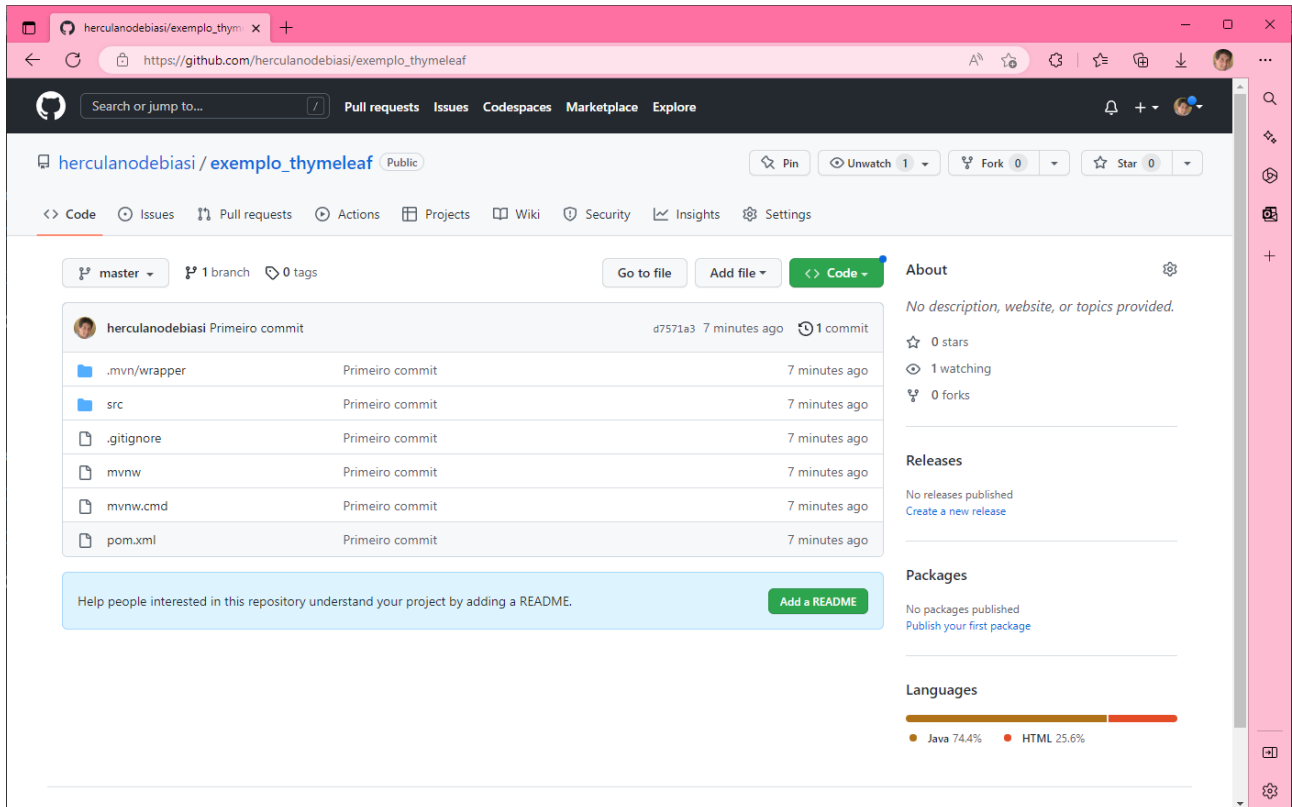
Repository  
[https://github.com/herculanodebiasi/exemplo\\_thymeleaf.git](https://github.com/herculanodebiasi/exemplo_thymeleaf.git)

Close

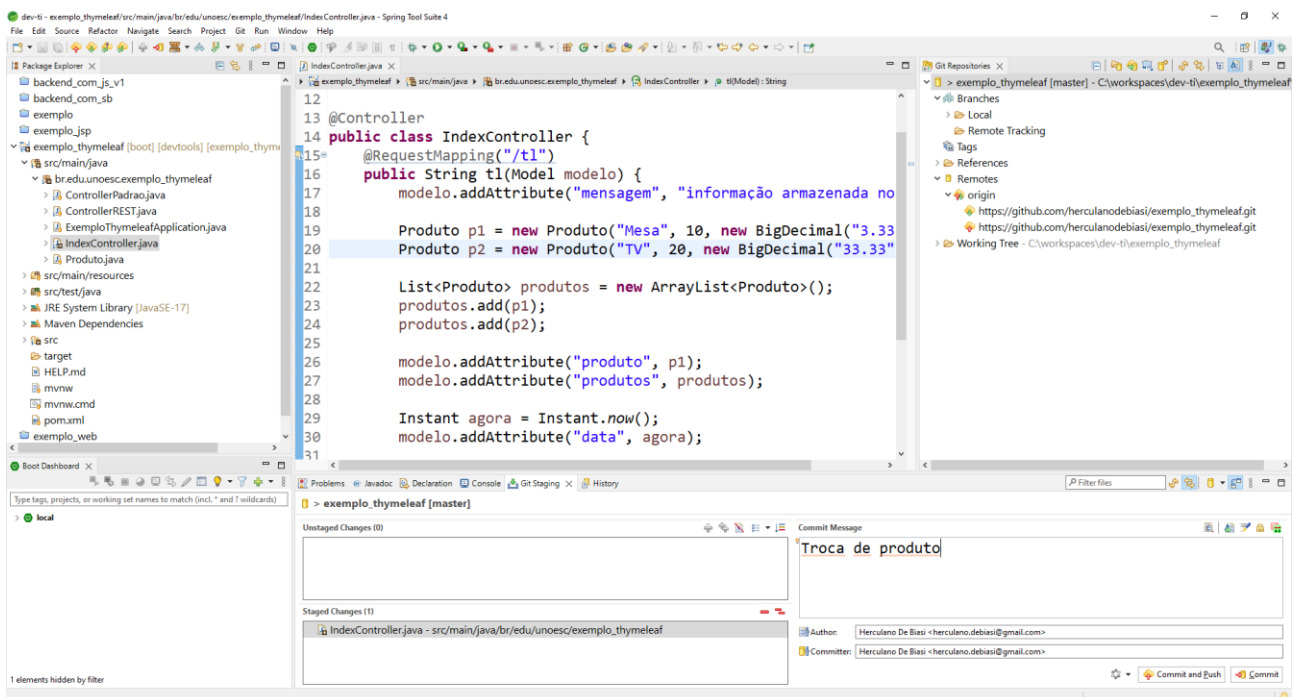
Isso deve ter configurado o repositório remoto.



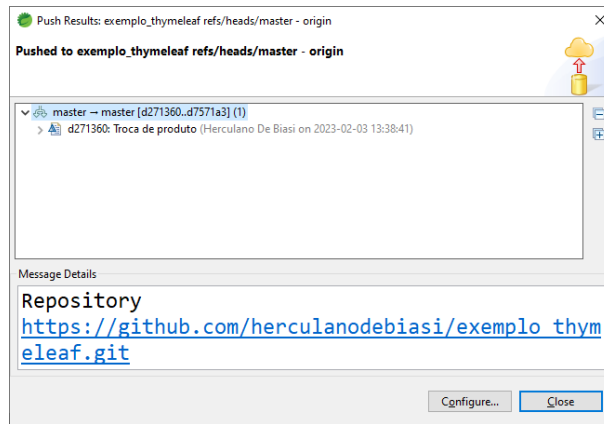
Ao atualizar a página do GitHub, a página abaixo deverá aparecer.



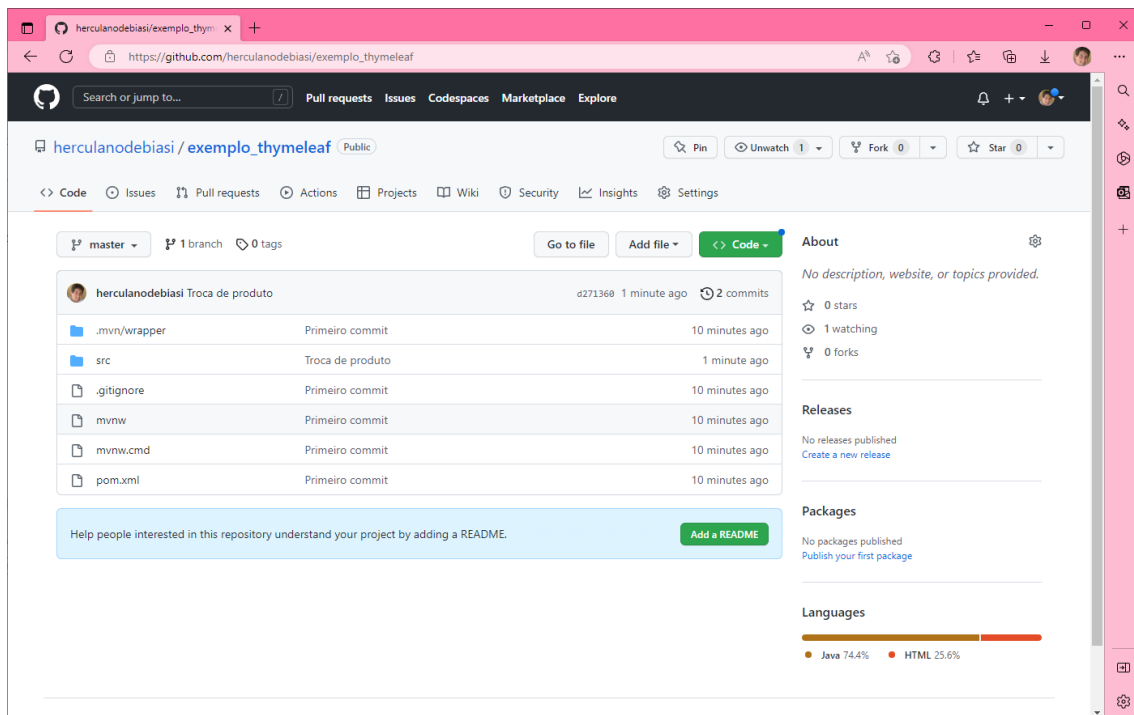
Faça agora a seguinte alteração e adicione o arquivo para ser comitado. Clique em *Commit and Push* ...



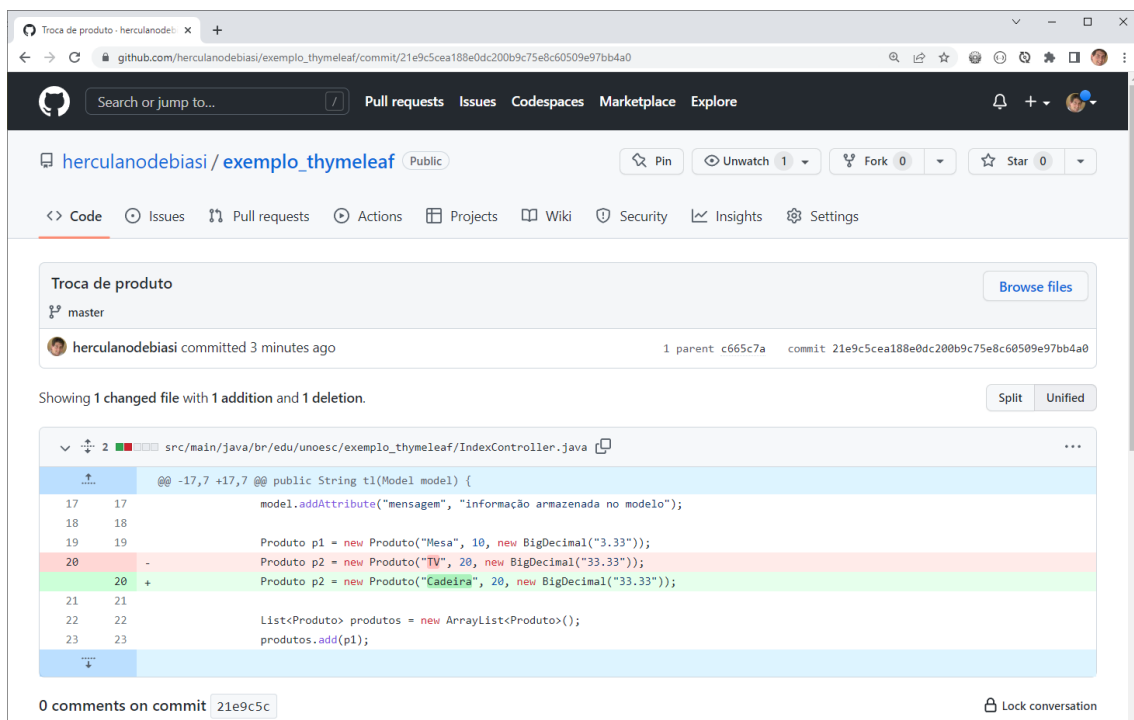
Clique em *Close*:



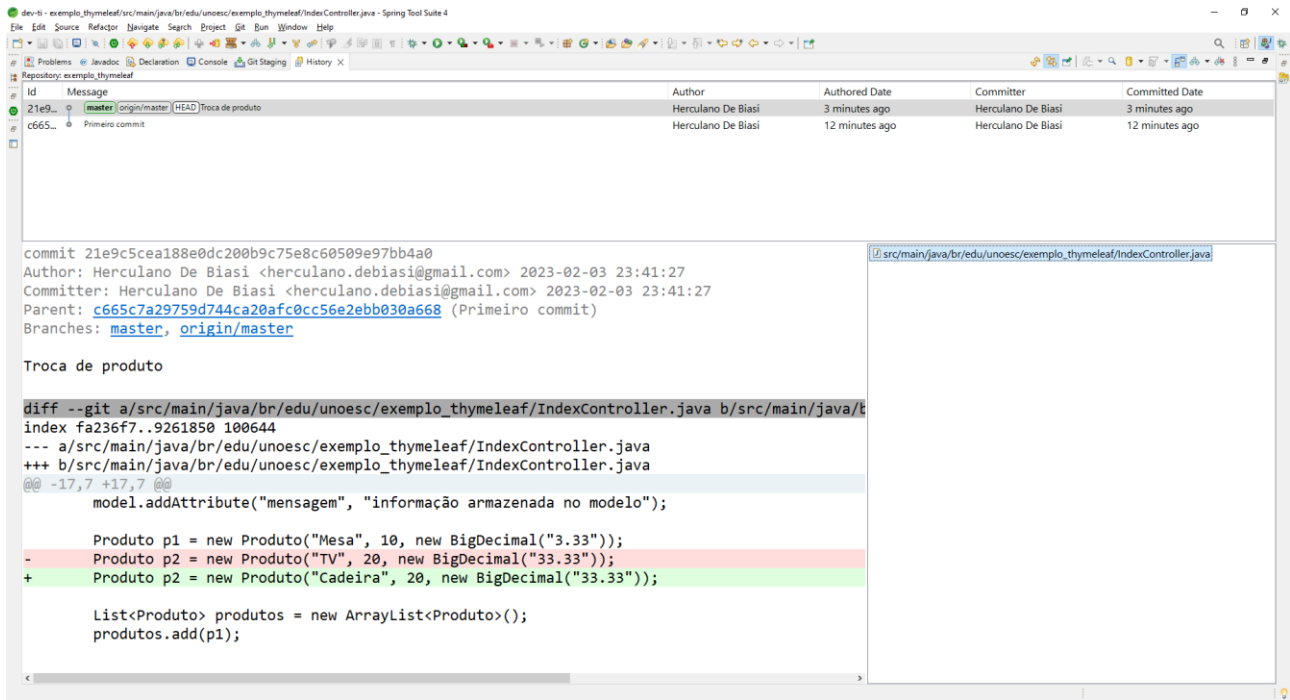
Atualize a página no GitHub para ver a atualização finalizada:



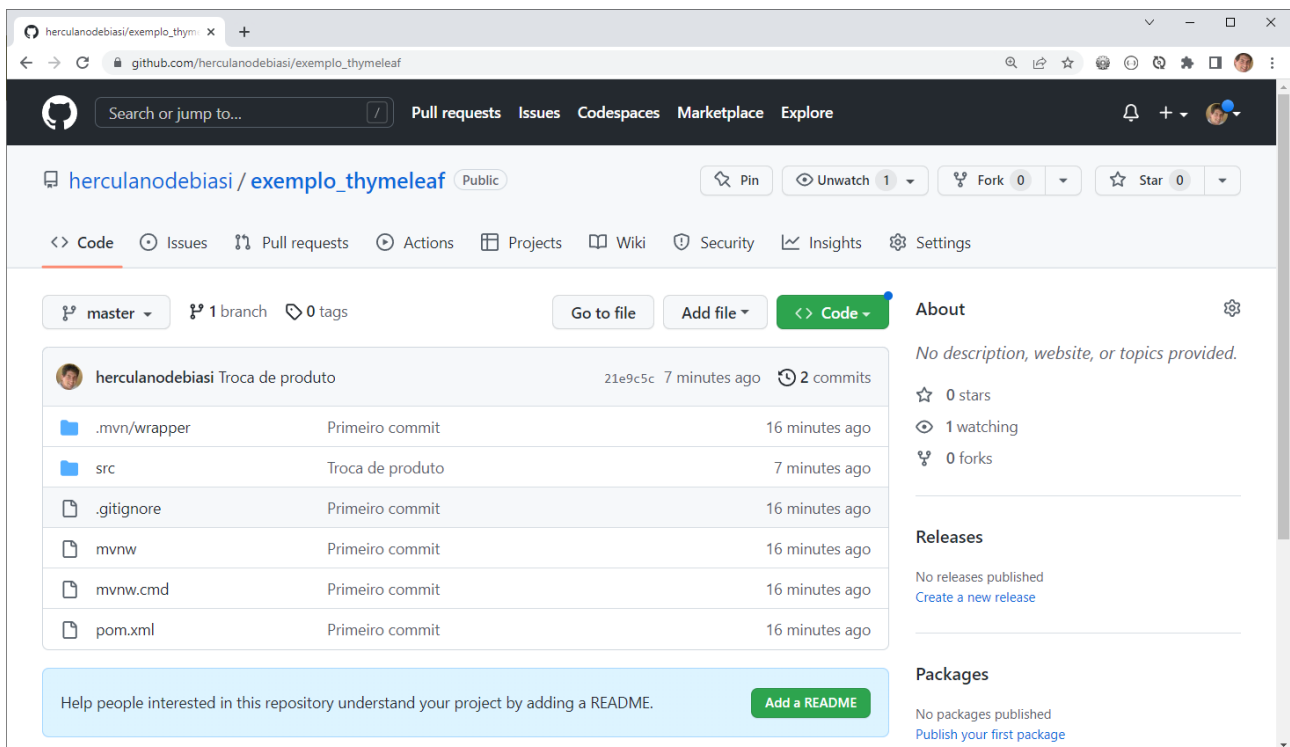
Ao clicar na mensagem 'troca de produto' é possível ver a alteração realizada neste *commit* específico.



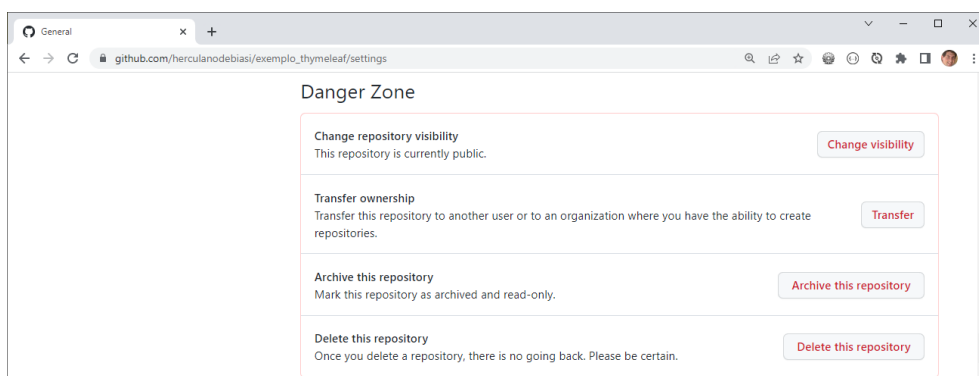
No STS é possível ver o histórico (*log*) na janela *History*. Ao selecionar um *commit* específico na parte superior da janela e um arquivo específico no lado direito, pode-se ver, na janela inferior, as diferenças entre as versões.



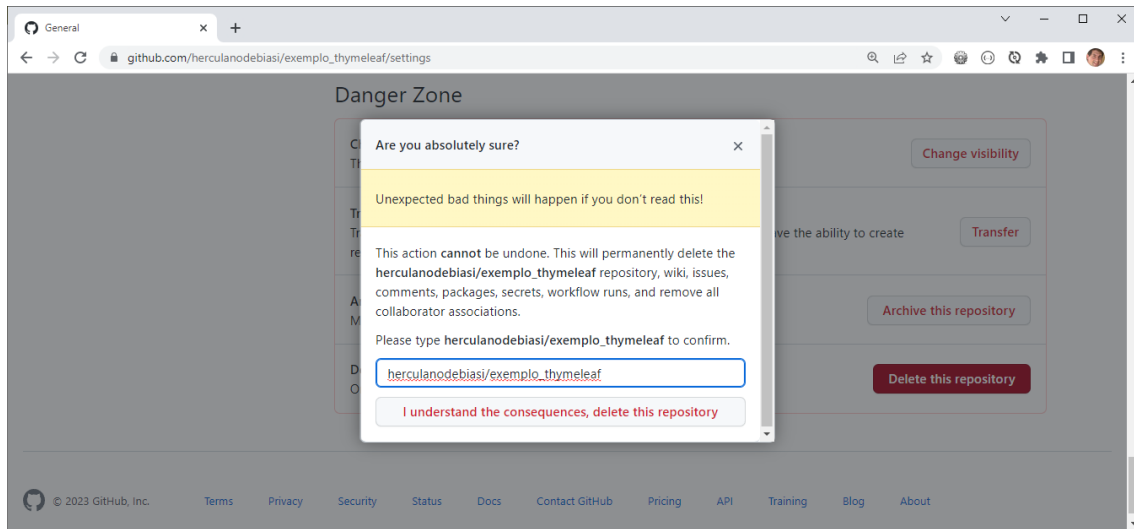
Para remover um repositório do GitHub, clique no botão *Settings* do repositório.



Role a página até o final e clique em *Delete this repository*.



Para remover definitivamente o repositório é necessário digitar o nome dele na caixa que irá aparecer.



Faça isso e clique no botão para apagar o repositório do GitHub.