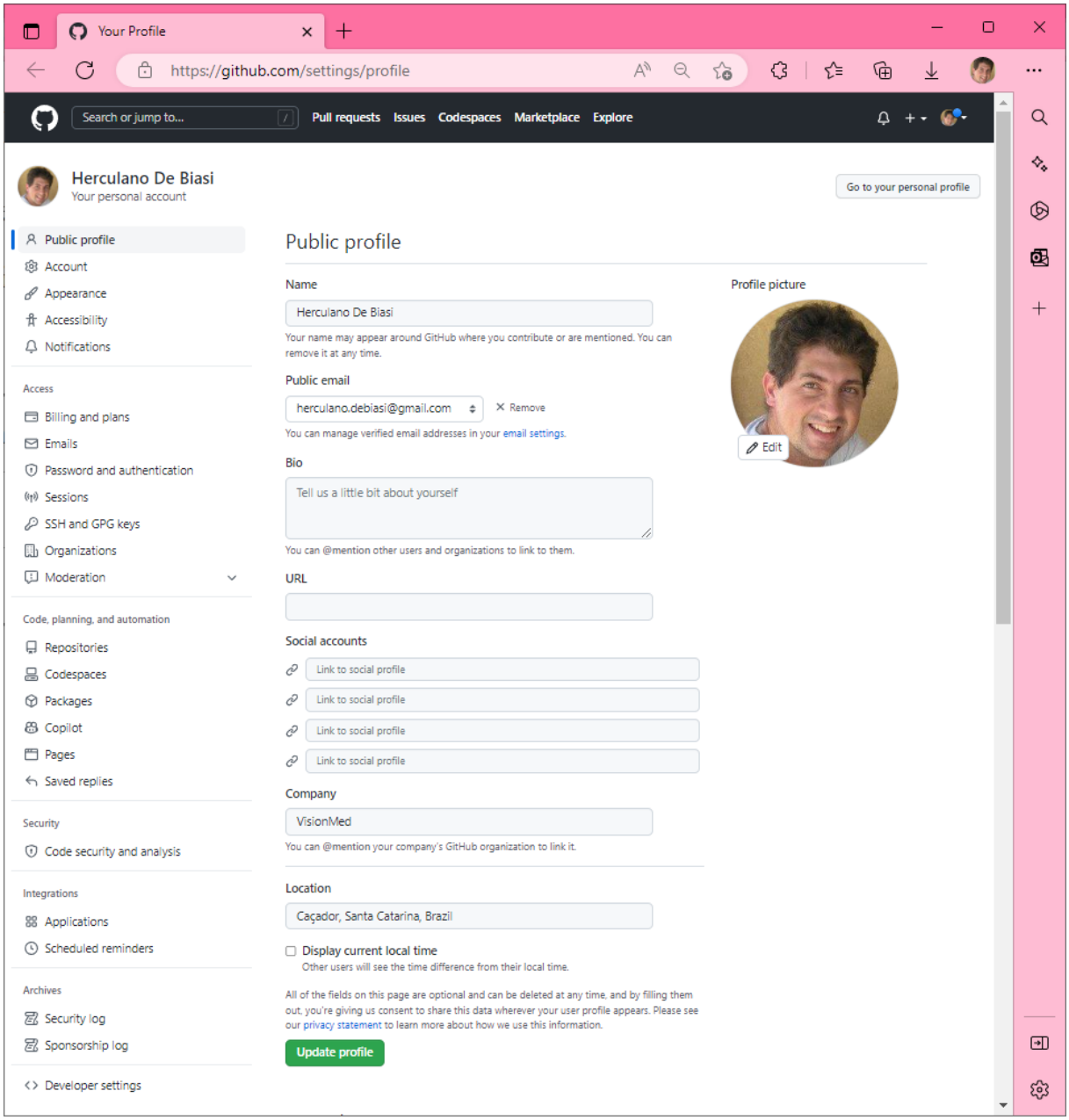
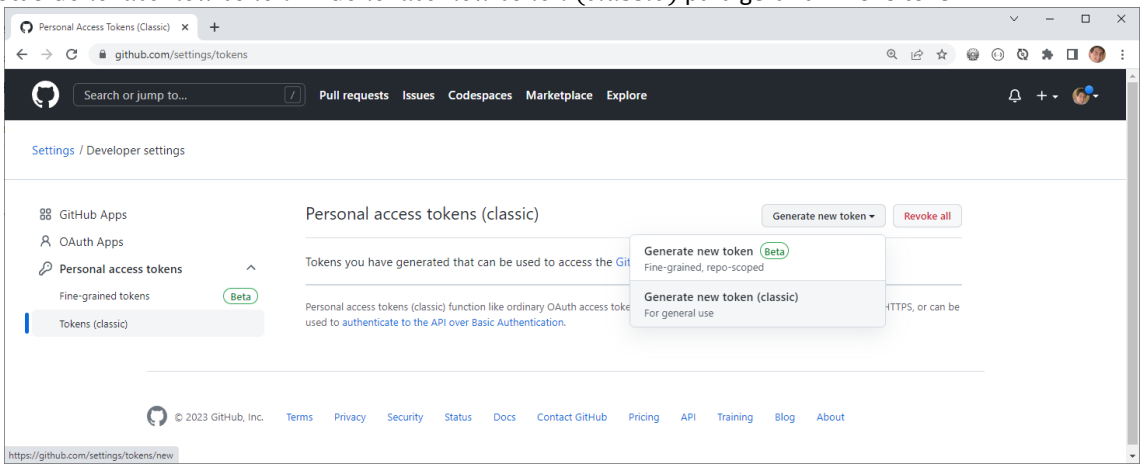
1 GitHub

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.

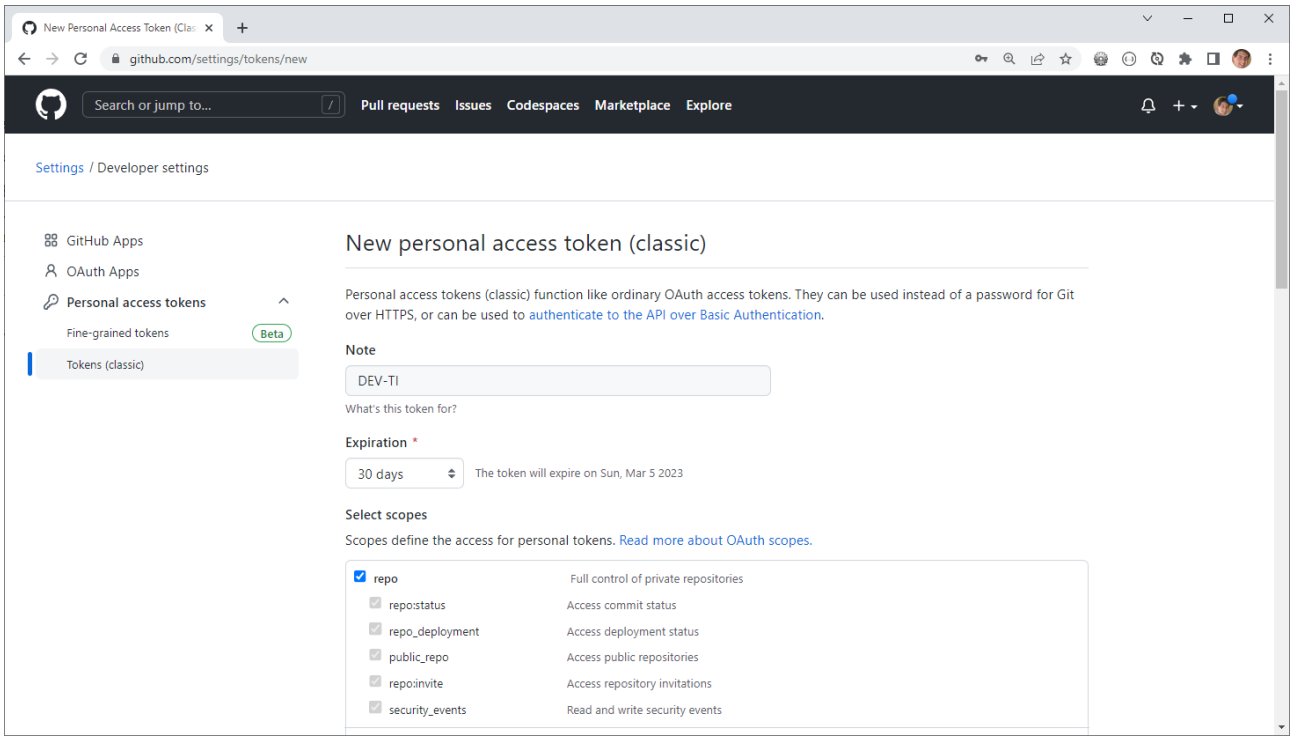
Para fazer isso primeiramente entre na sua conta no GitHub e vá para a configurações do perfil. Role a página até o final para entrar na opção *Developer settings*.



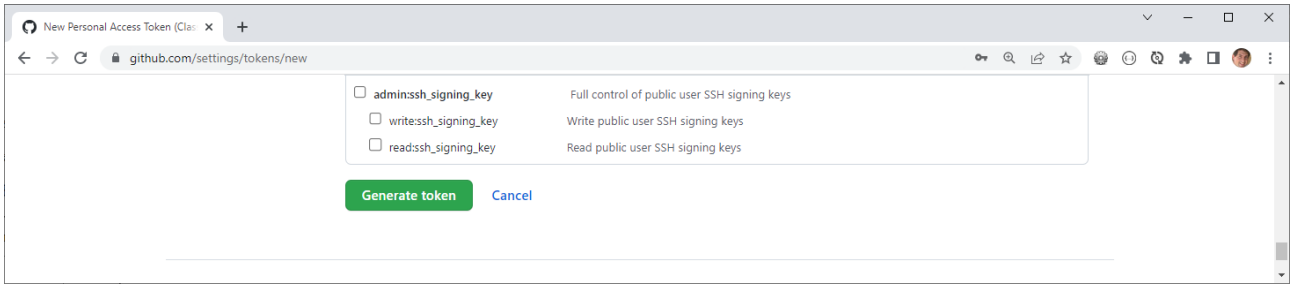
Clique no botão *Generate new token* → *Generate new token (classic)* para gerar um novo *token*.



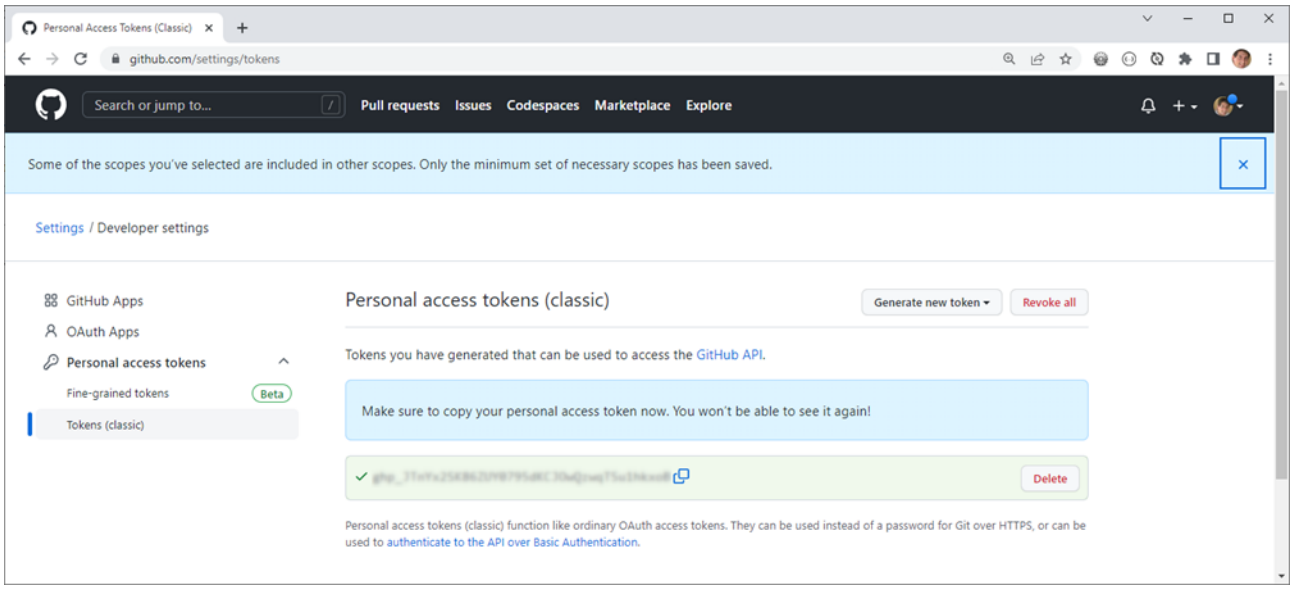
Forneça um nome qualquer no campo $Note$ e marque a opção $repo$.



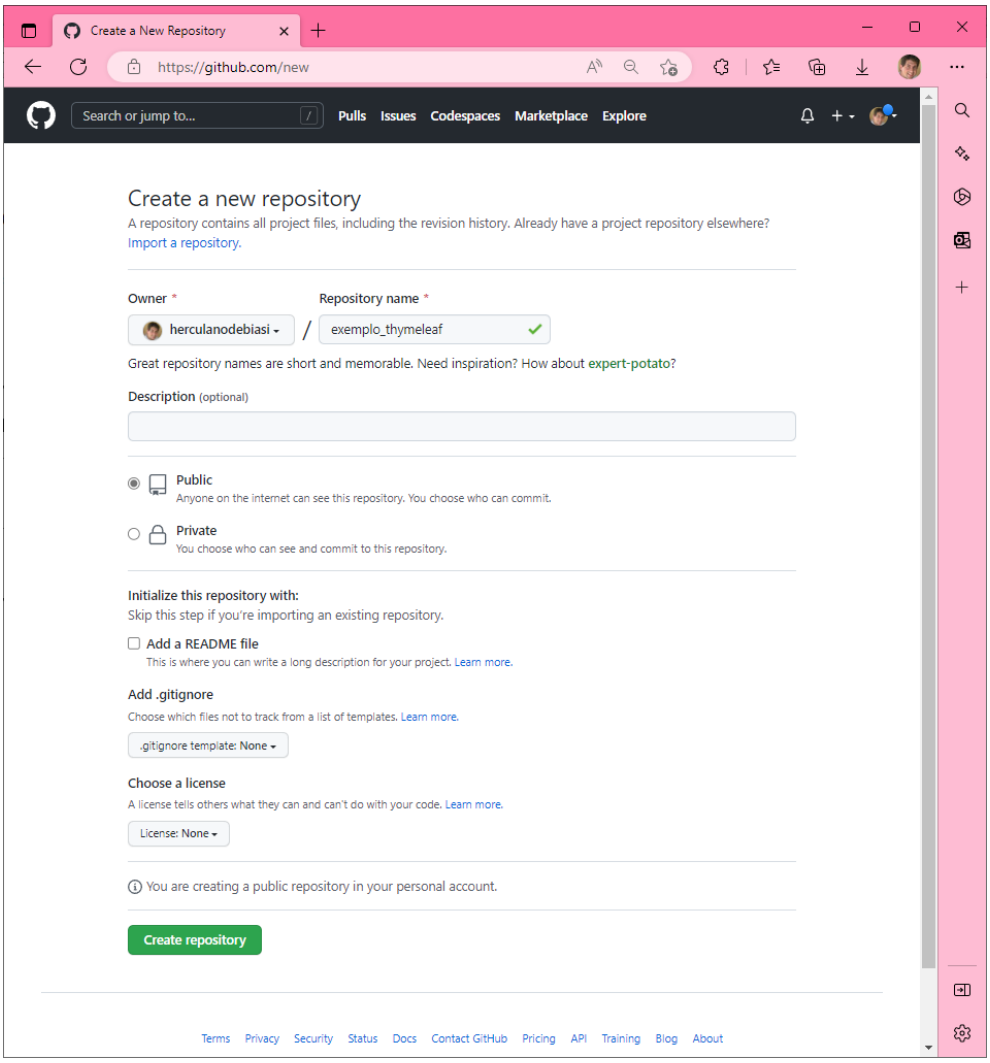
Role a página até o final e clique em $Generate\ token$.



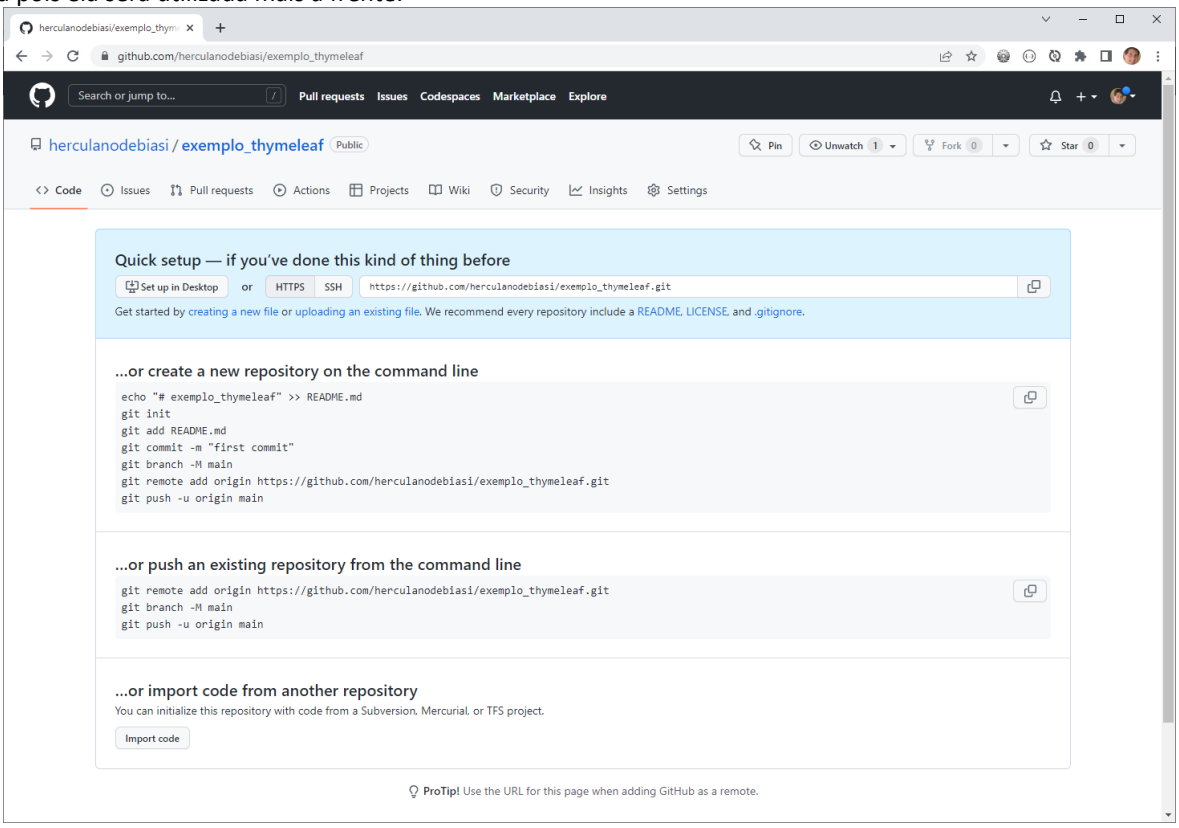
O *token* irá aparecer somente uma vez, se ele for perdido, terá que ser gerado novamente. Clique no botão copiar e armazene-o em outro local, pois será necessário utilizá-lo novamente mais tarde. Por questões de segurança o *token* não deve ser divulgado ou compartilhado, pois funciona como uma senha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



Clique neste *link* para criar um novo repositório. Preencha com o nome do repositório e clique no botão *Create repository*.

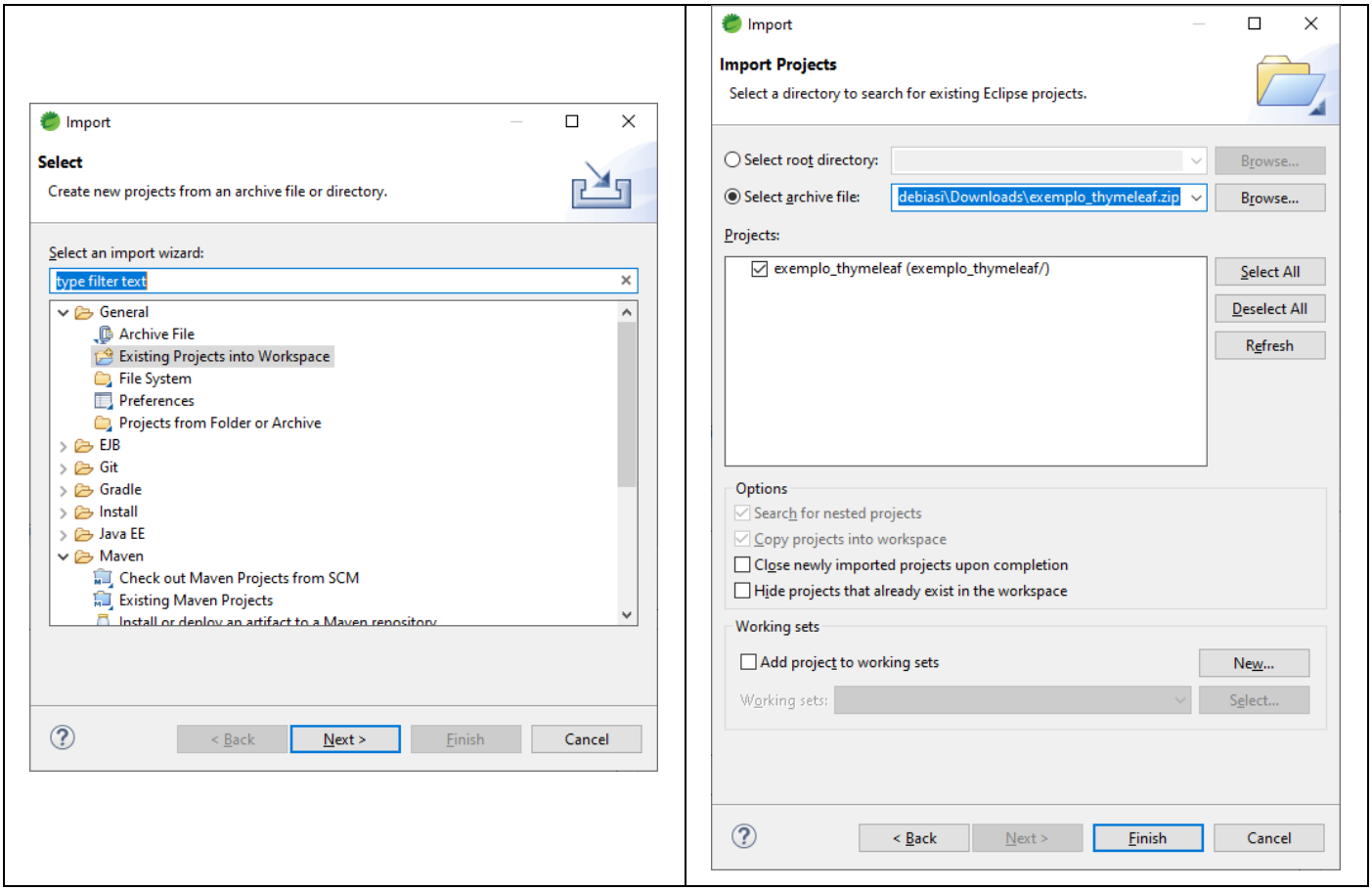


A página abaixo deverá aparecer mostrando que o repositório remoto foi criado no GitHub assim como várias instruções sobre como criar um repositório local e conectá-lo ao repositório remoto utilizando o cliente Git em linha de comando. Mantenha esta tela aberta pois ela será utilizada mais à frente.

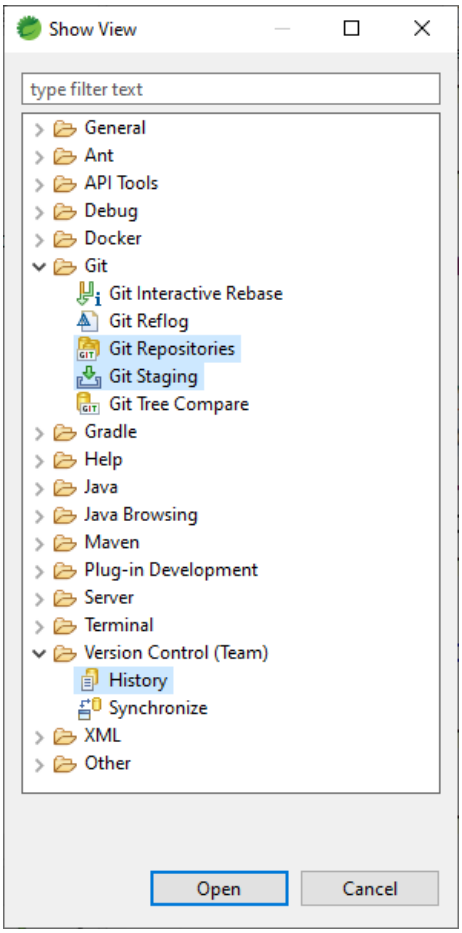


2 STS

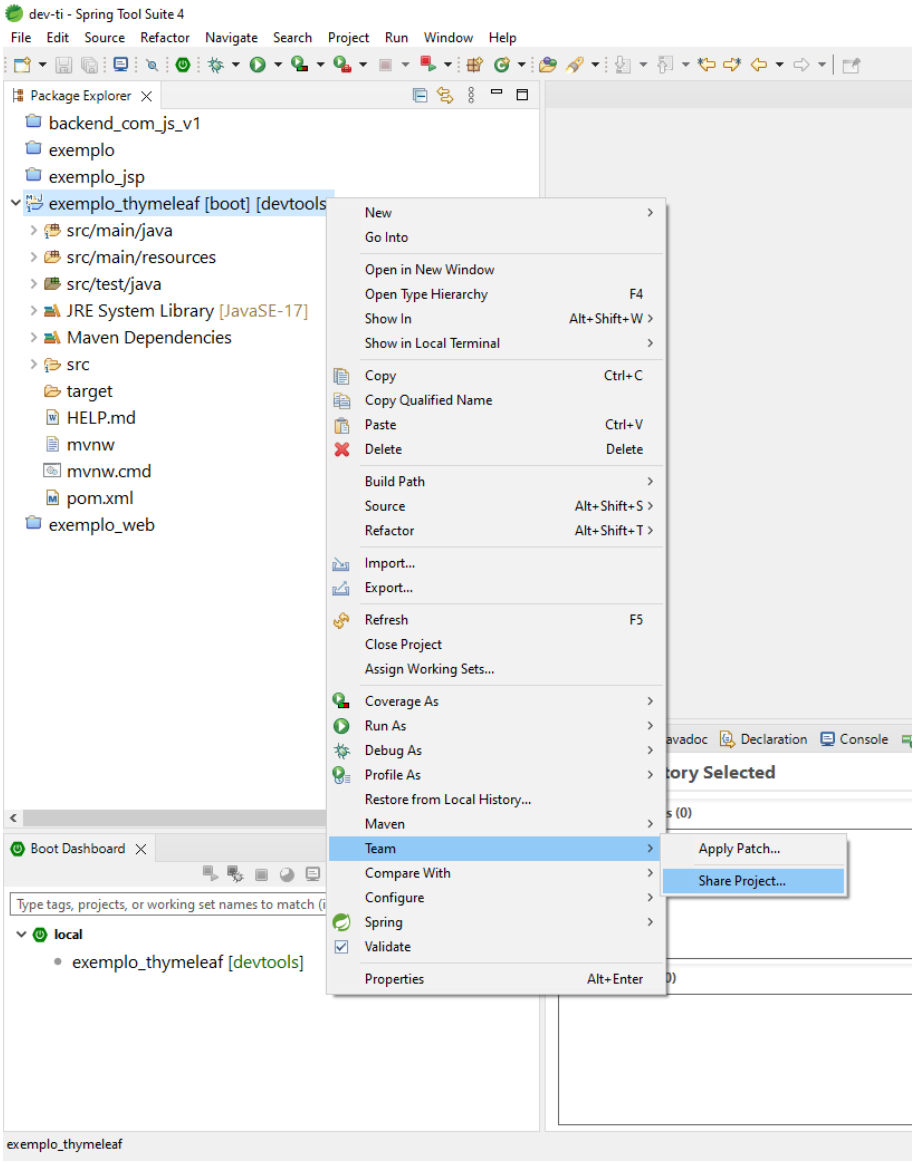
No moodle, baixe o projeto '[exemplo thymeleaf]'. Se este projeto já existir no *workspace*, apague-o. Importe o projeto no STS indo ao menu $File \rightarrow Import$ e seguindo as telas abaixo:



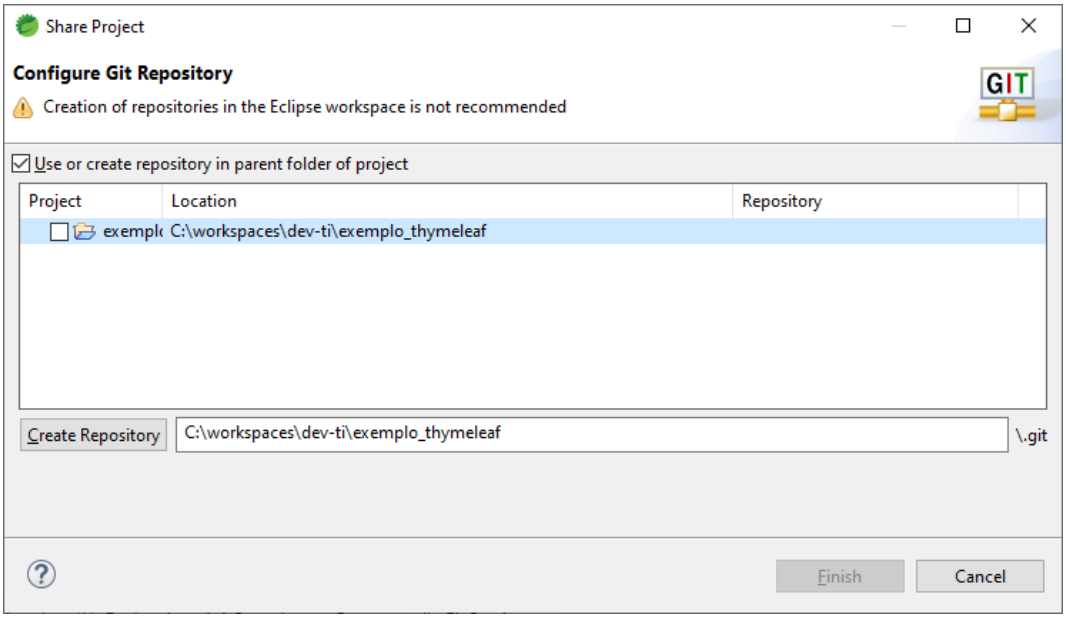
No menu $Window \rightarrow Show\ View$ selecione as 3 janelas abaixo (para isso segure a tecla $Ctrl$):



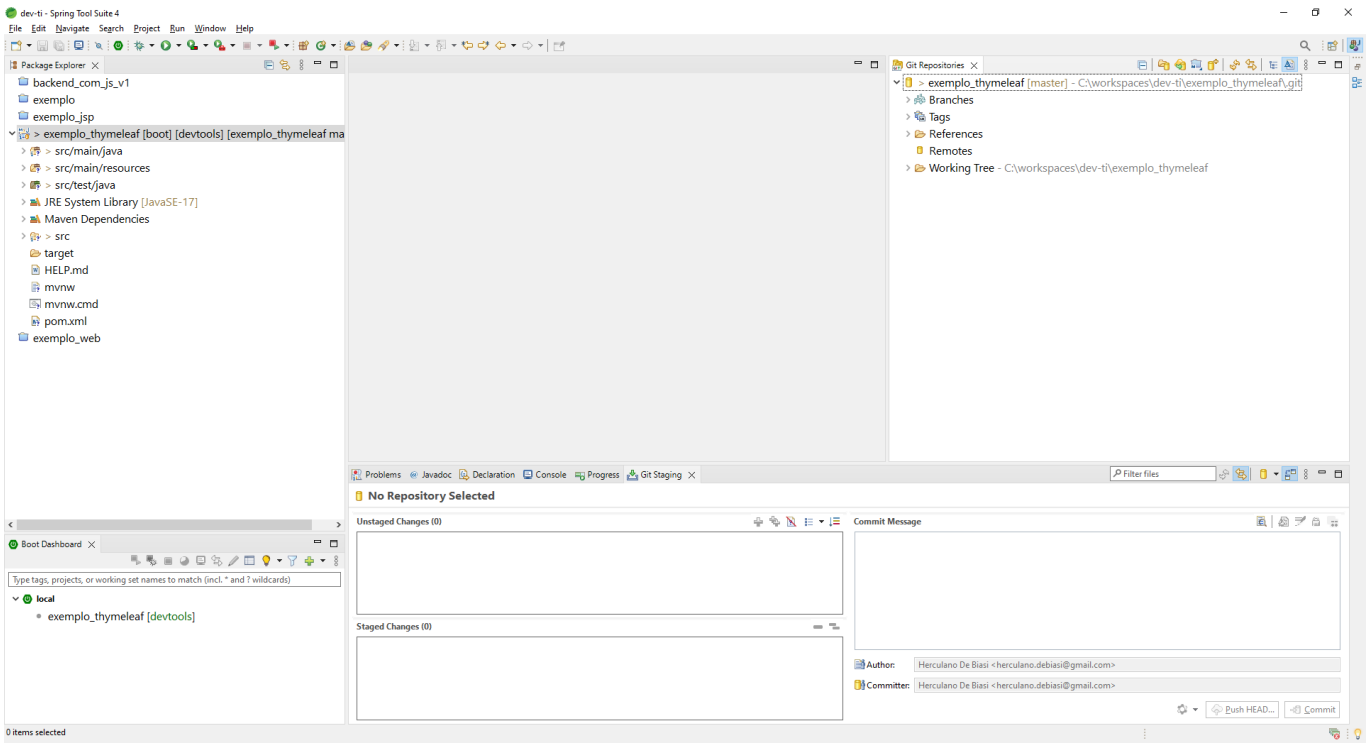
Para criar um repositório local, sob controle do Git, abra o projeto recém importado. A seguir, clique com o botão direito sobre ele e escolha a opção $Team \rightarrow Share\ Project$ ...



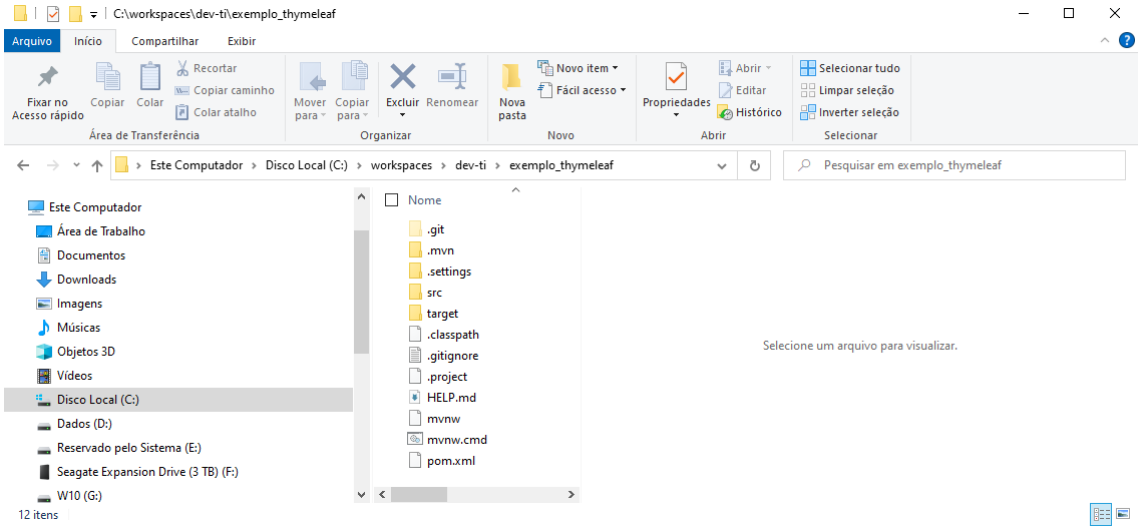
Selecione a opção $Use\ or\ create\ repository\ in\ parent\ folder\ of\ projet$. A seguir clique no botão $Create\ Repository$ e depois em $Finish$.



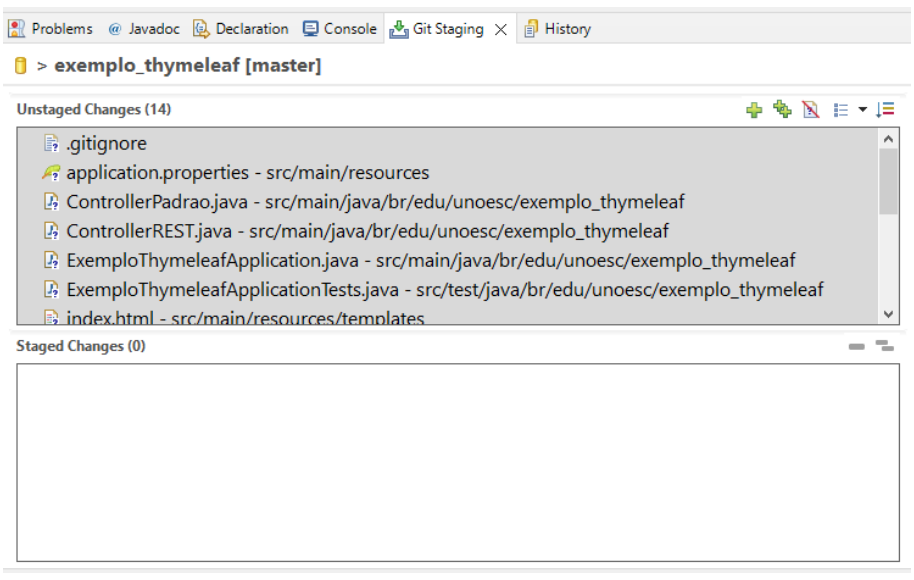
Com isso o repositório deve ter sido criado dentro do projeto atual no *workspace*.



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

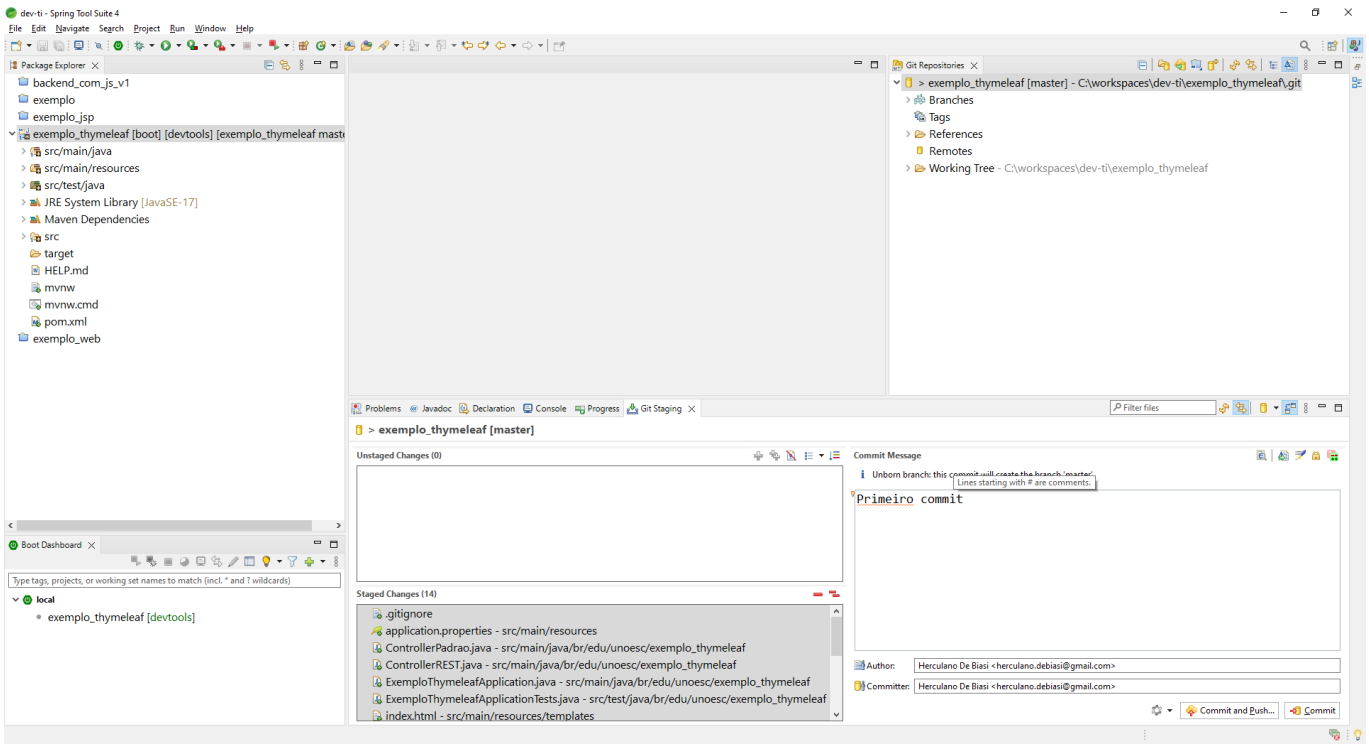


Abra a janela *Git Staging*. Pode-se notar que os arquivos do projeto estão na área *unstaged changes*. Para que as alterações sejam efetivadas no próximo *commit* é preciso que os arquivos sejam movidos para a área de *staging*. Para fazer isso clique no botão  . Caso não funcione, experimente fechar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

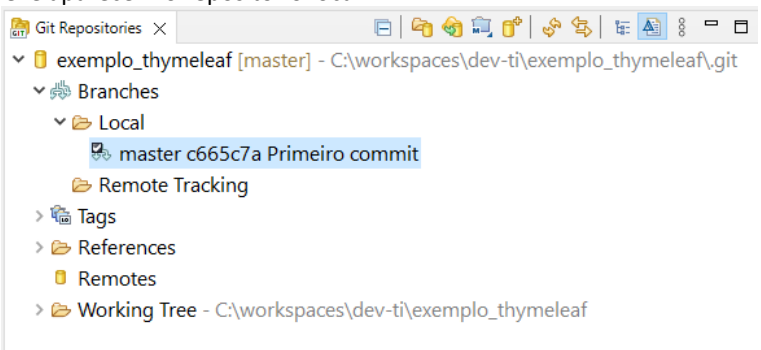


Arquivos adicionados na área de *staging* a fim que as alterações sejam efetivadas no próximo *commit.*

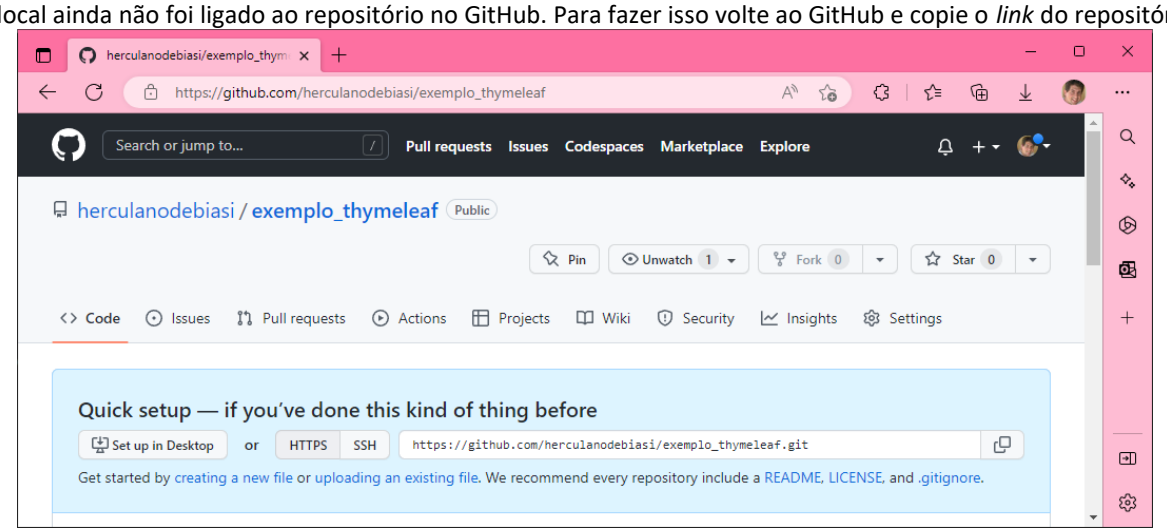
Escreva a mensagem 'Primeiro commit' na área *commit message* . Clique no botão *Commit* ...



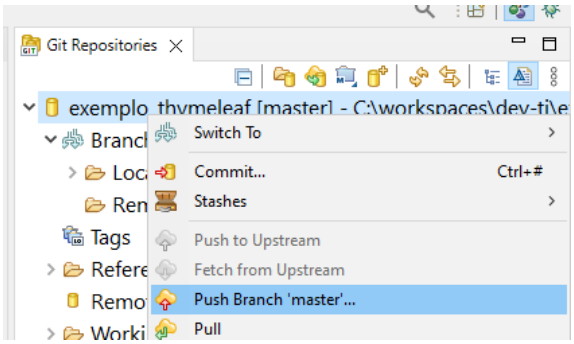
Ao fazer isso este *commit* já deve aparecer no repositório local.



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



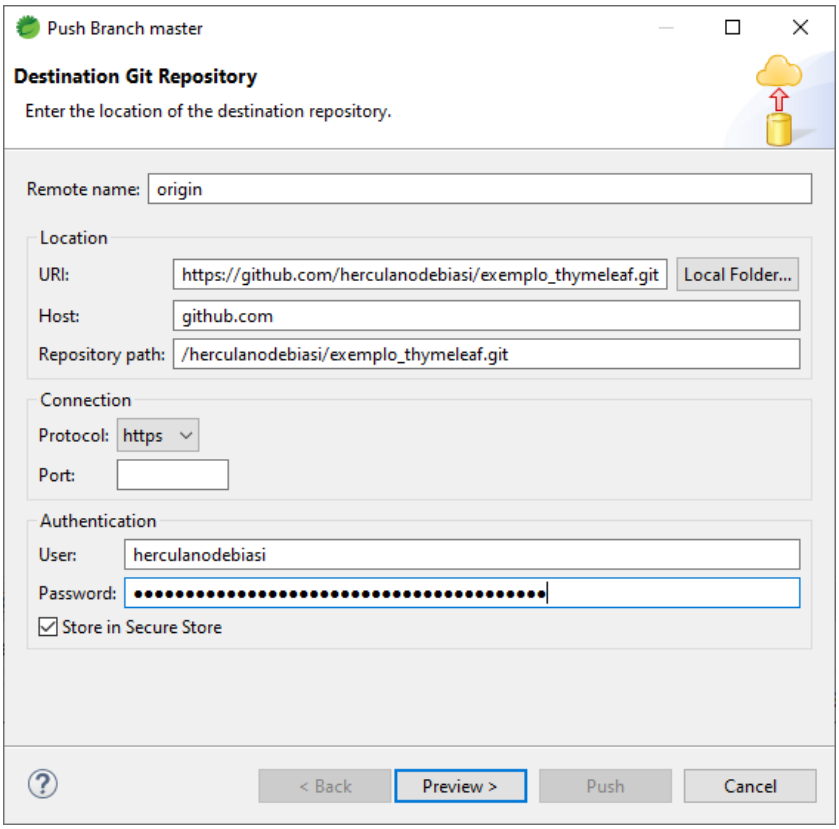
Clique agora sobre o projeto e escolha a opção $Push\ Branch\ 'master'$ ...



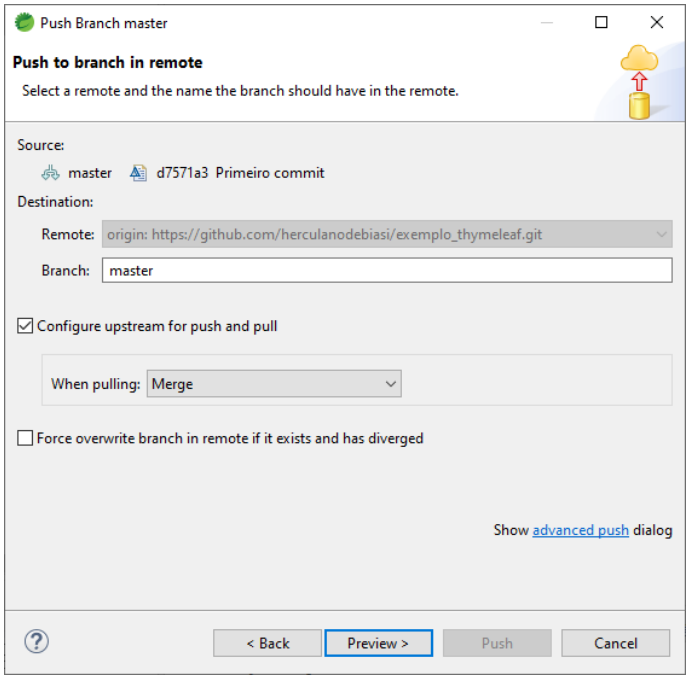
Os seguintes dados deverão aparecer na janela:

- URI: https://github.com/herculanodebiasi/exemplo_thymeleaf.git
- *Host*: github.com
- *Repository path*: /herculanodebiasi/exemplo_thymeleaf.git

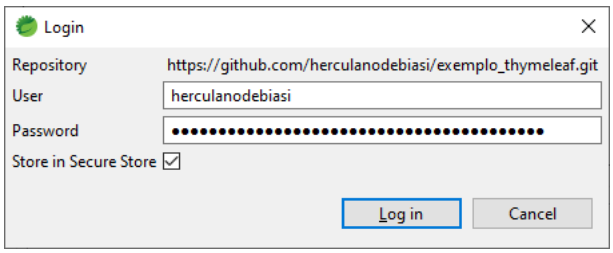
No campo $Password$ insira o *token* gerado no início deste roteiro.



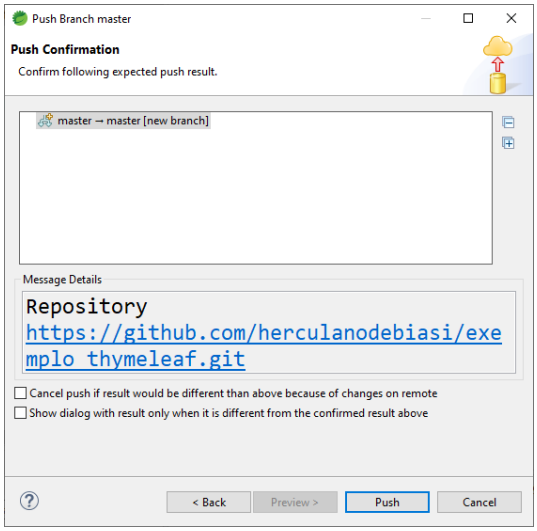
Clique em $Preview$.



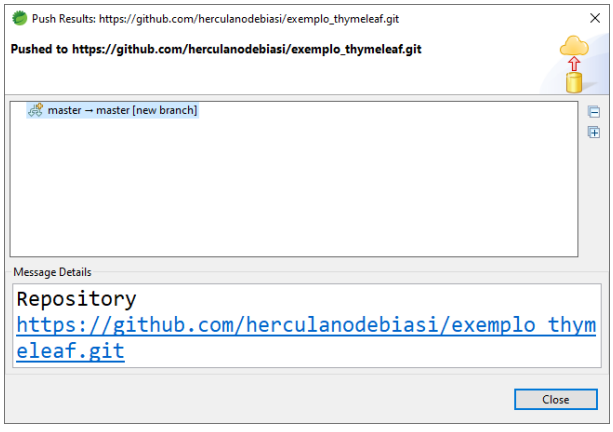
Clique novamente em *Preview*. Caso solicitado entre novamente com o usuário e *token.*



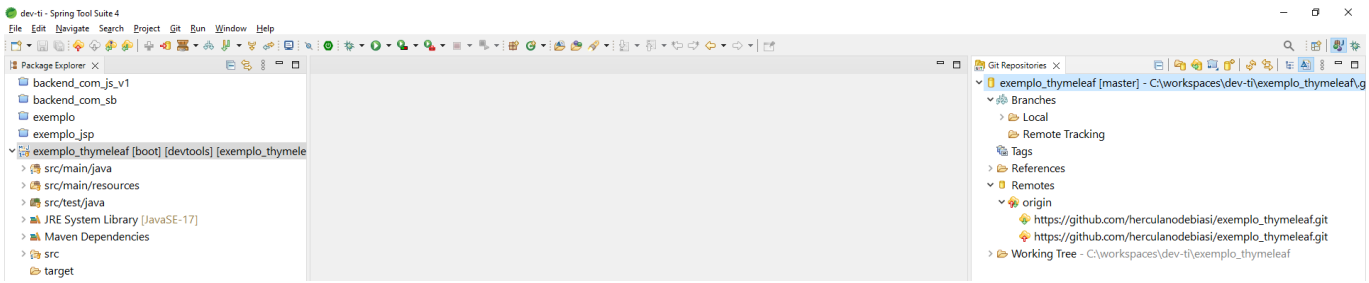
Clique em *Push*.



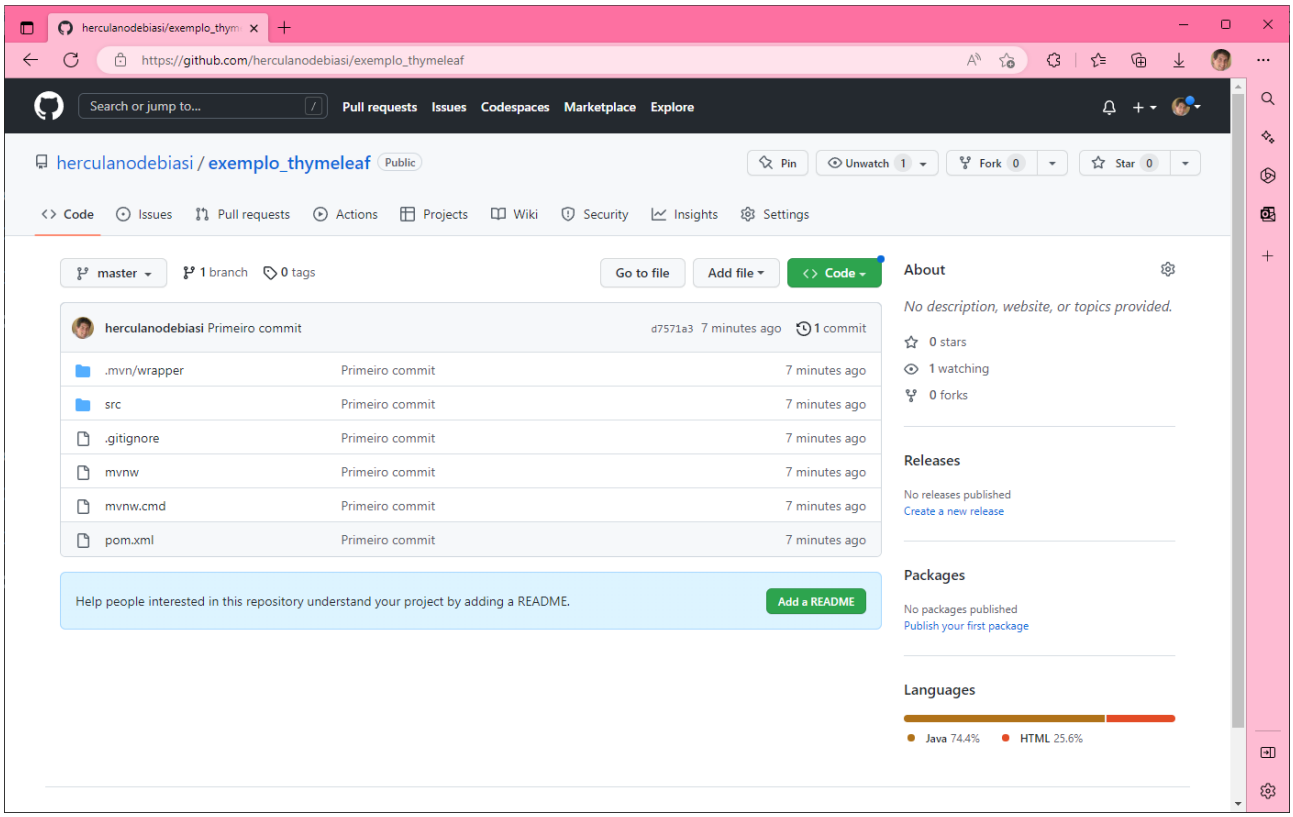
Clique em *Close*.



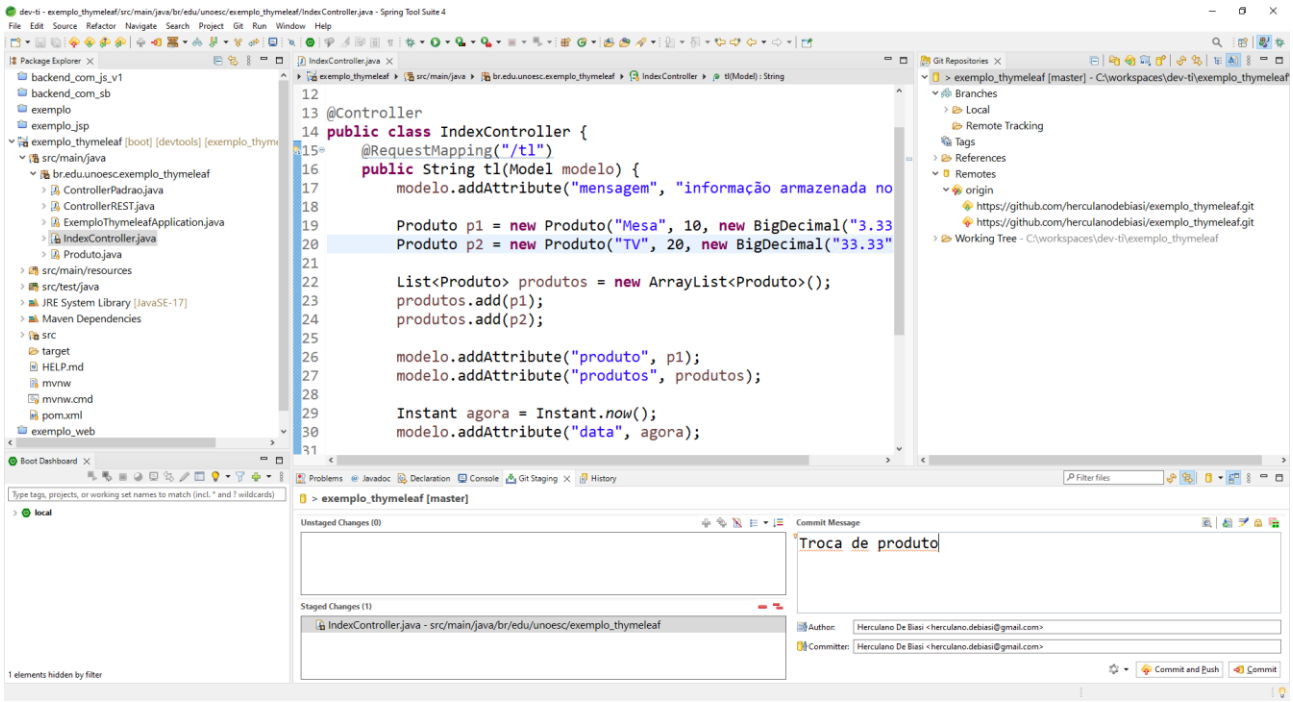
Isso deve ter configurado o repositório remoto.



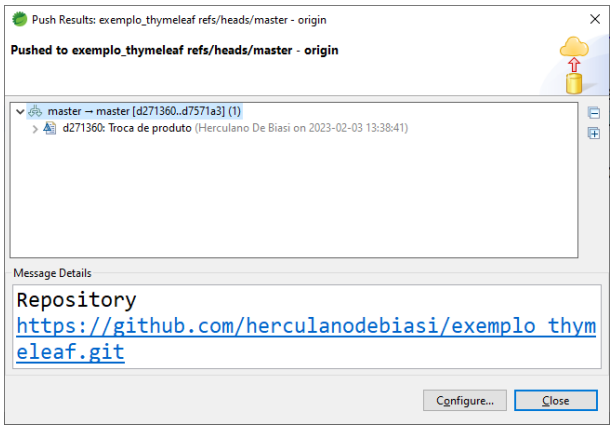
Ao atualizar a página do GitHub, a página abaixo deverá aparecer.



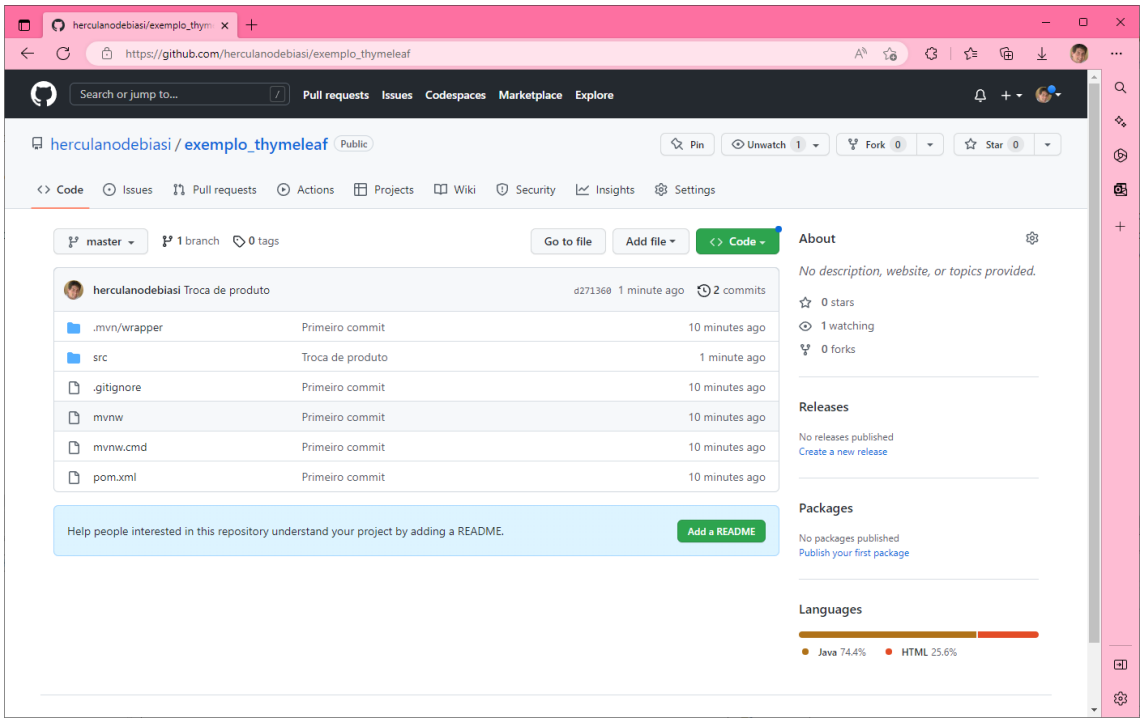
Faça agora a seguinte alteração e adicione o arquivo para ser comitado. Clique em *Commit and Push* ...



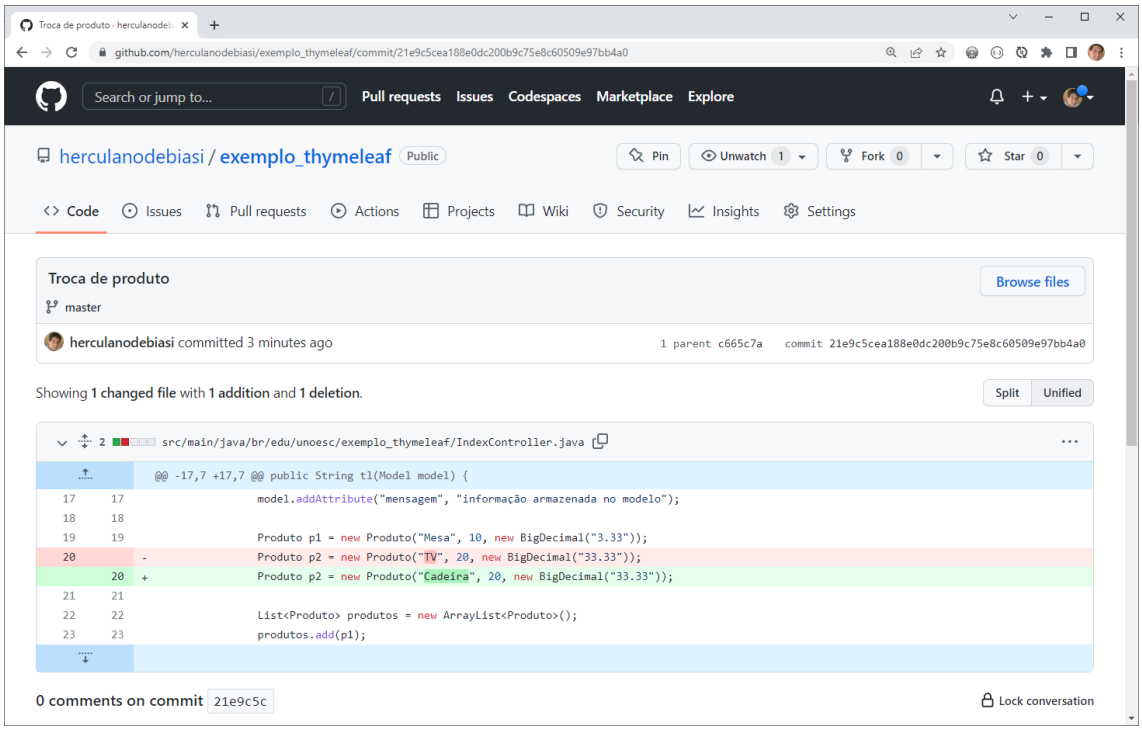
Clique em *Close*:



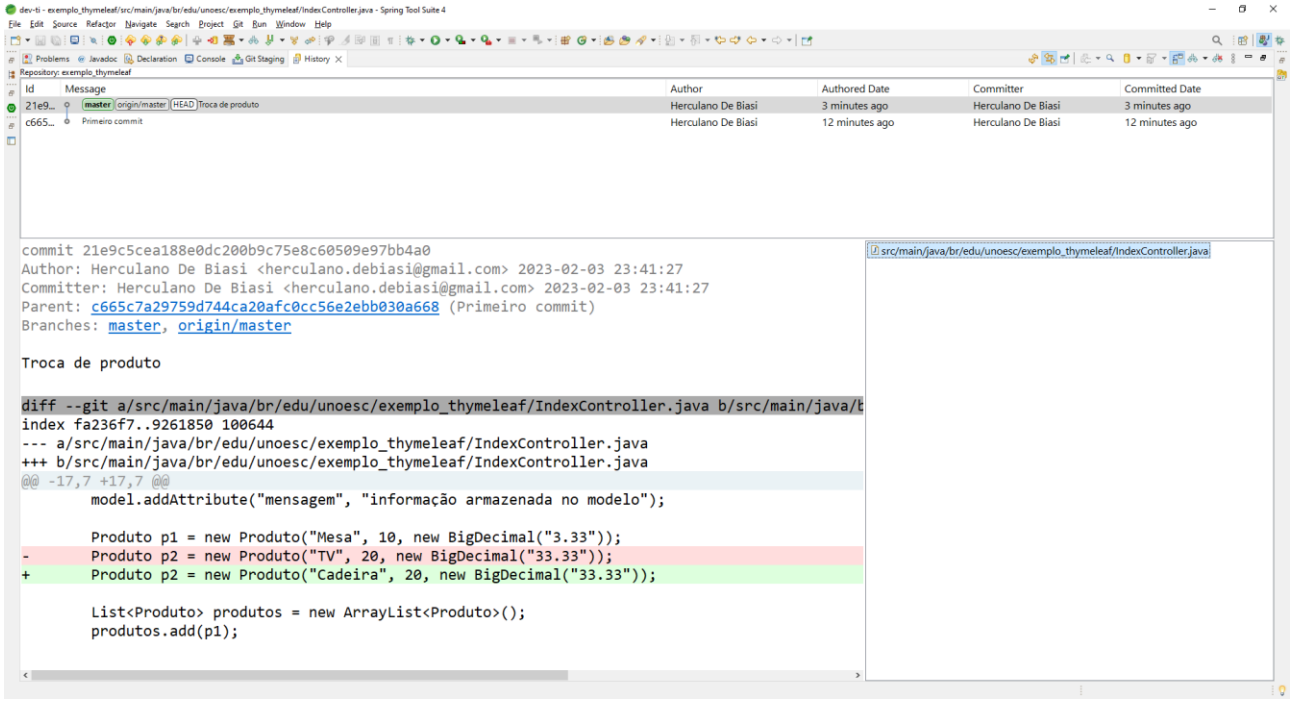
Atualize a página no GitHub para ver a atualização finalizada:



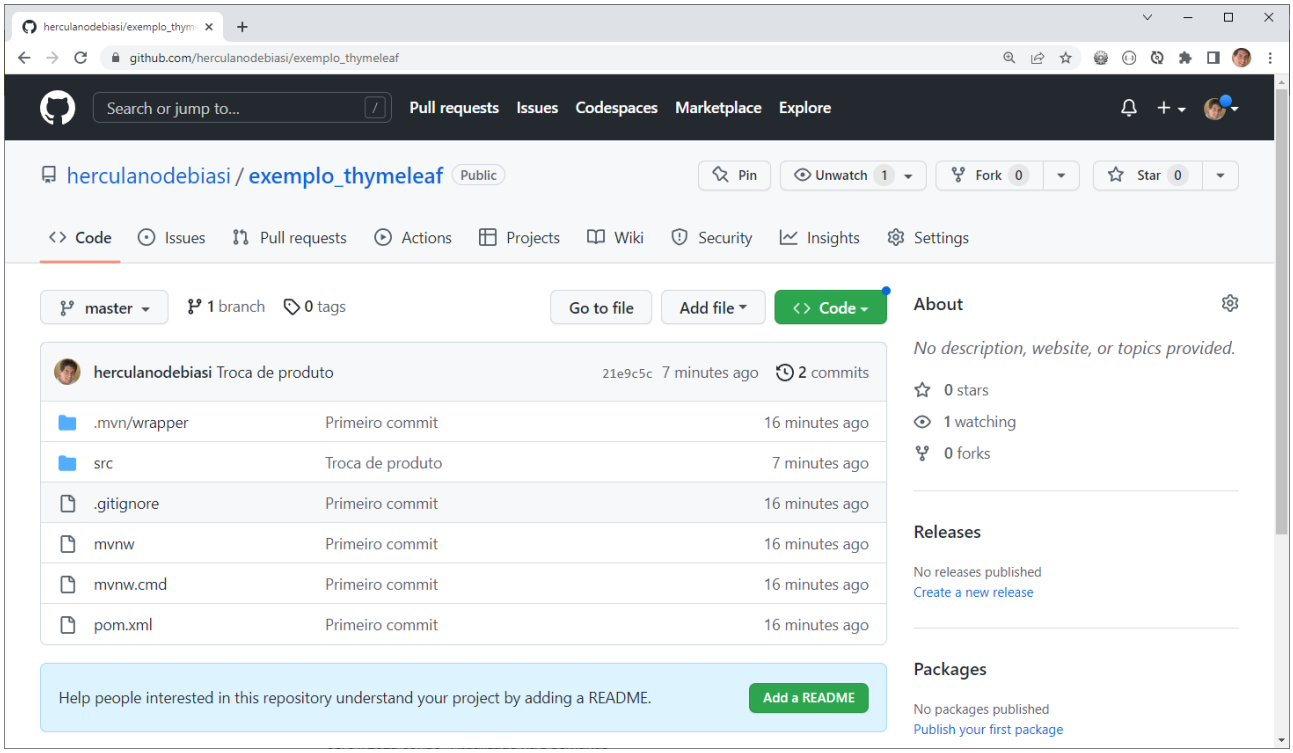
Ao clicar na mensagem 'troca de produto' é possível ver a alteração realizada neste *commit* específico.



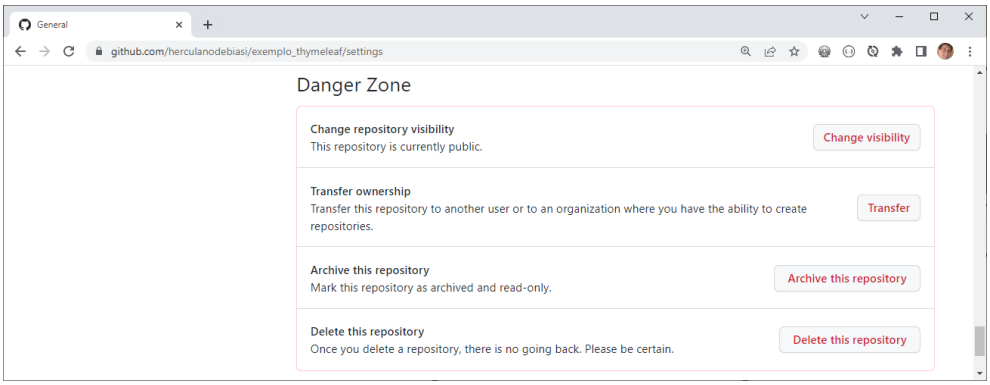
No STS é possível ver o histórico (*log*) na janela $History$. Ao selecionar um *commit* específico na parte superior da janela e um arquivo específico no lado direito, pode-se ver, na janela inferior, as diferenças entre as versões.



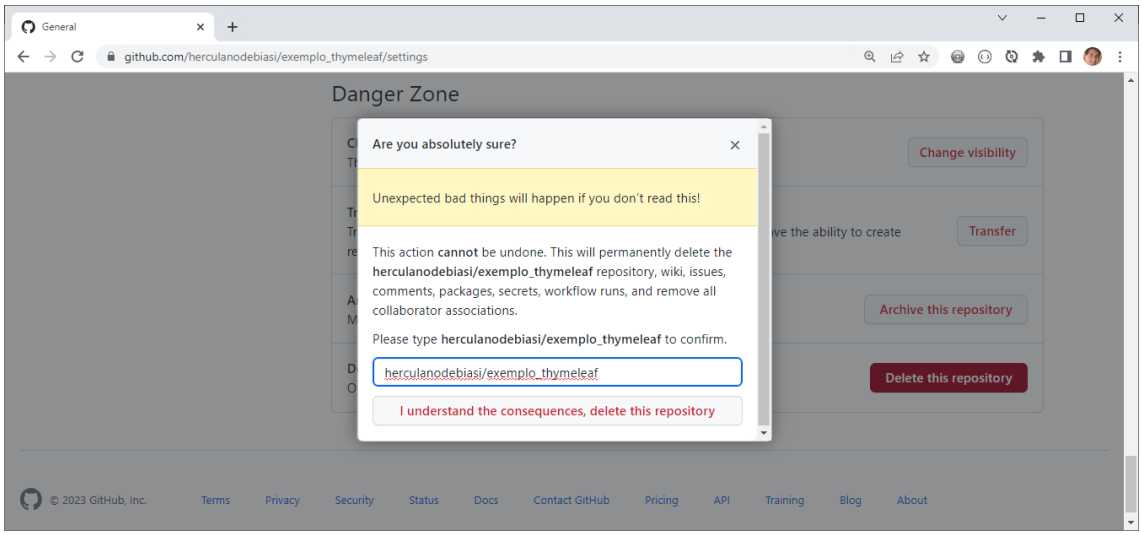
Para remover um repositório do GitHub, clique no botão $Settings$ do repositório.



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.