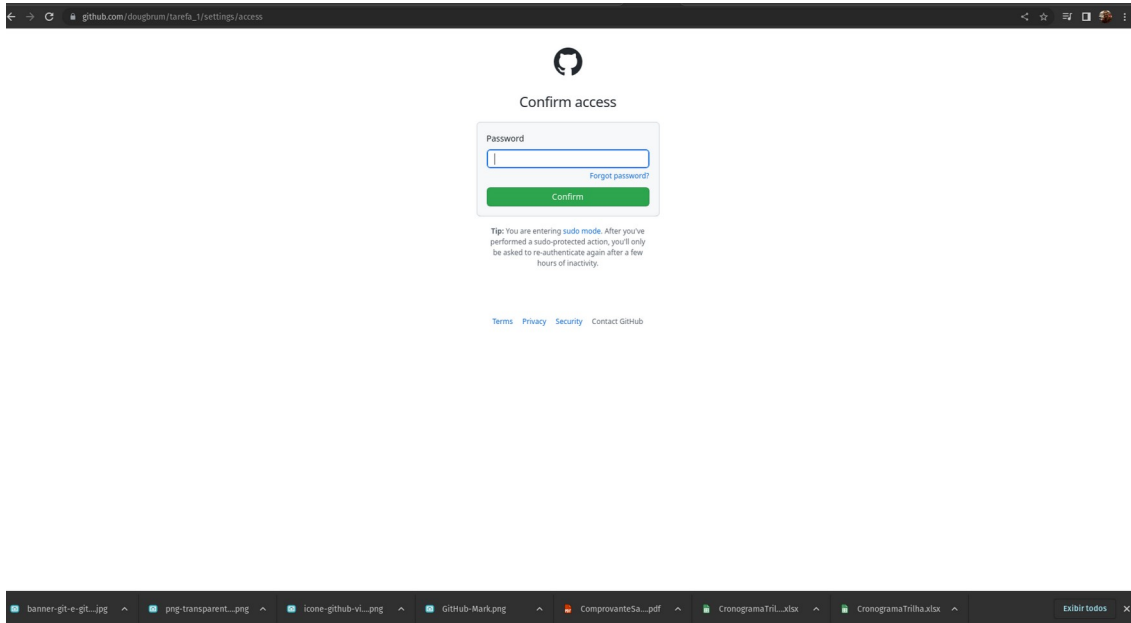


Registro para a resolução da tarefa proposta

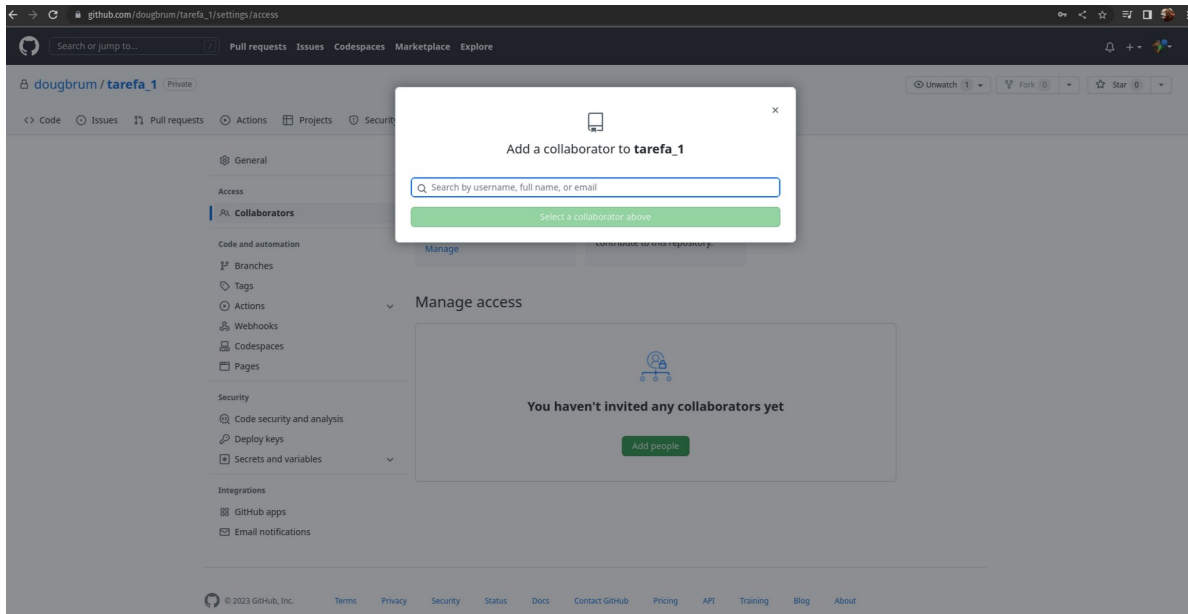
1) Criar uma conta privada no GitHub e conceder acesso ao seu instrutor da Sprint.

Como eu já possuía uma conta no GitHub, vou mostrar o processo que utilizei para dar acesso ao meu instrutor.

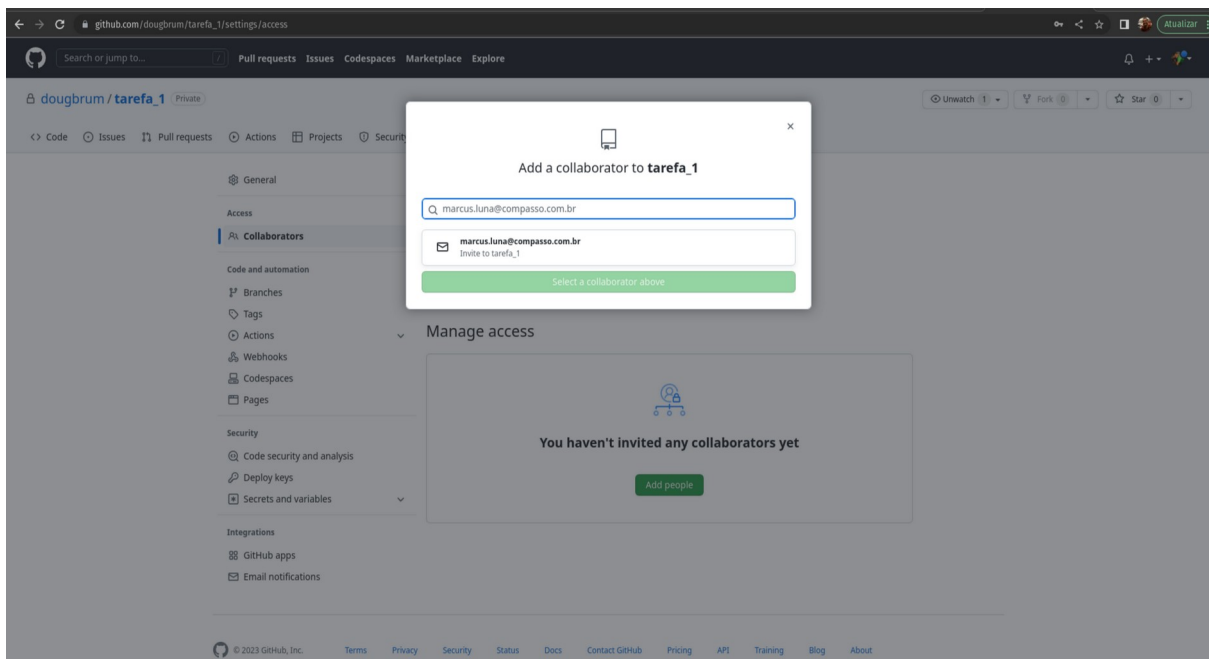
Em meu repositório tarefa_1 cliquei na aba “Settings”. Em seguida na coluna à esquerda cliquei em “Collaborators”. Nesse momento o GitHub pede a minha senha pessoal, por segurança.



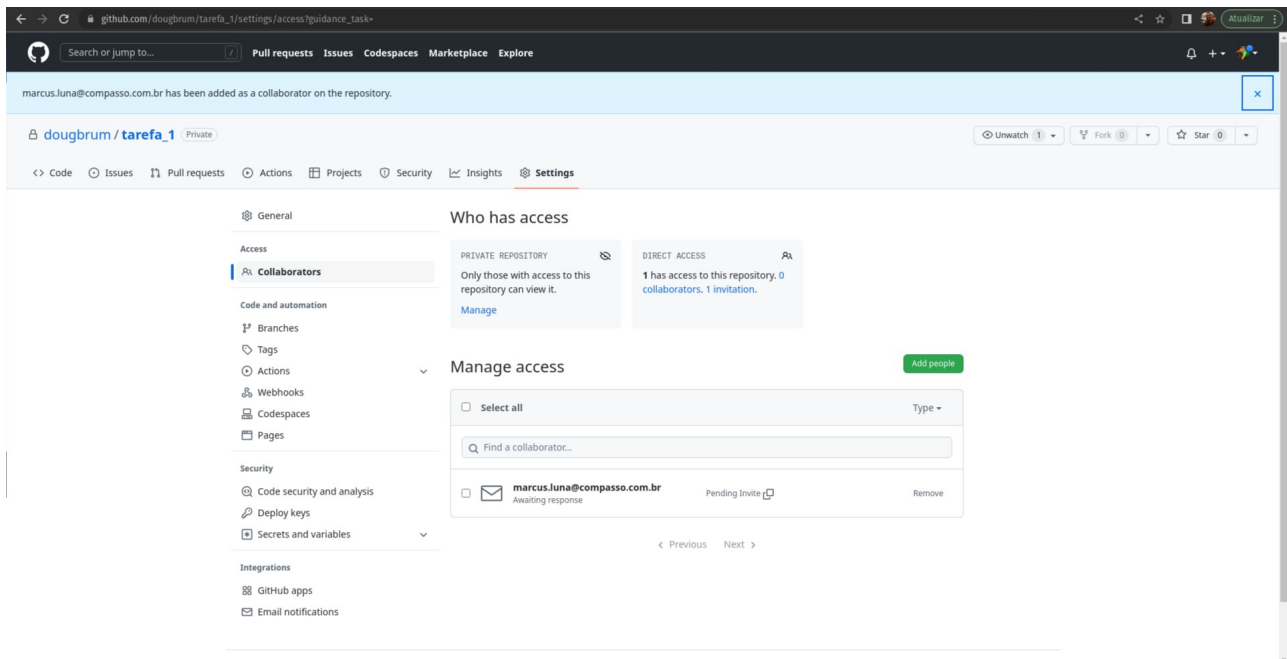
Clicando em “Add people” podemos colocar o e-mail do instrutor da Sprint, marcus.luna@compasso.com.br, e em seguida confirma em “Select a collaborator above”.



Basta preencher a lacuna com o e-mail e clicar no retângulo verde. Como abaixo mostrado:

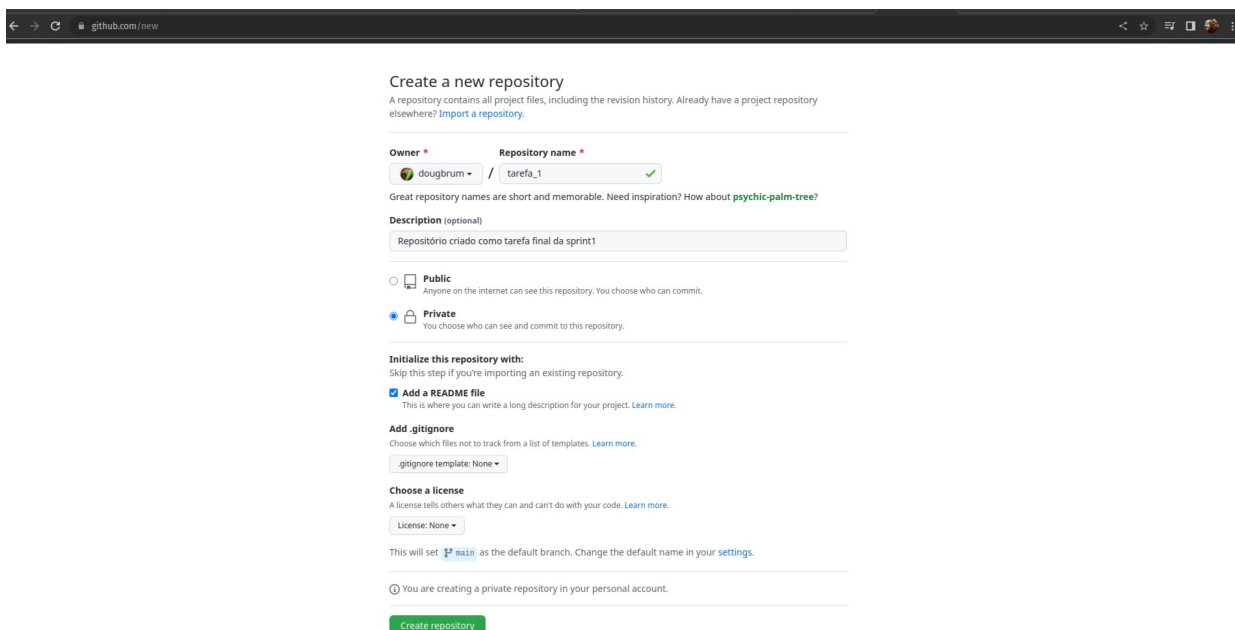


Agora que o convite foi realizado, basta aguardar para que o colaborador aceite o convite.



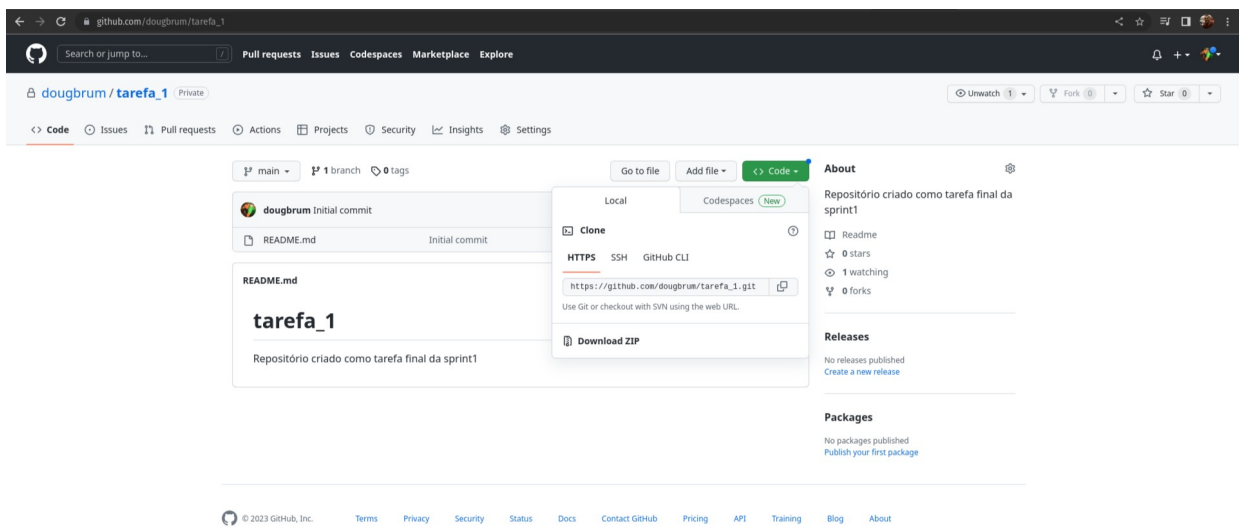
2. Criar um repositório(privado) para armazenar anotações e códigos criados ao longo do Programa de Bolsas.

Para Criar um novo repo privado, basta clicar em New repository que a página abaixo será exibida.

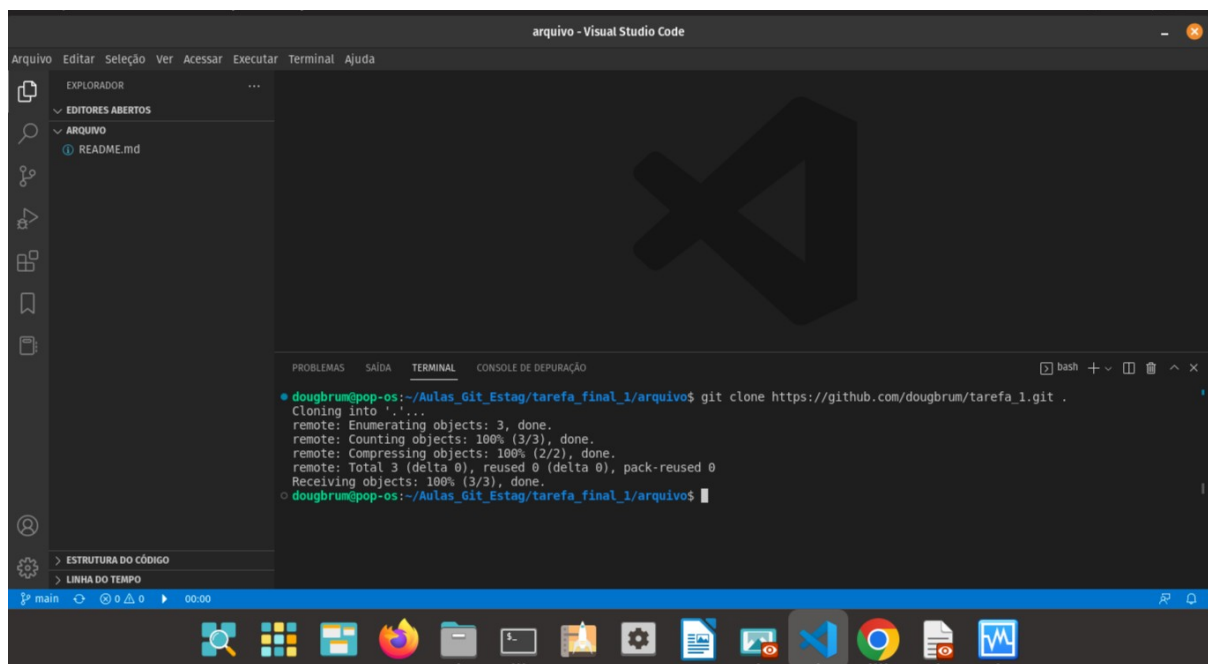


Basta preencher essa página com o nome do repositório, colocar como privado e marcar a opção “Add a README file”. Depois de seguir esses passos basta clicar em create repository.

Repositório Criado, basta ir no botão “Code” e copiar o endereço https disponível. Usaremos esse endereço para clonarmos o repositório para nosso repositório local usando o VSCODE.



Abrindo o VSCODE e com o caminho de onde queremos clonar o repositório remoto, basta dar um git clone e colar o endereço previamente copiado.



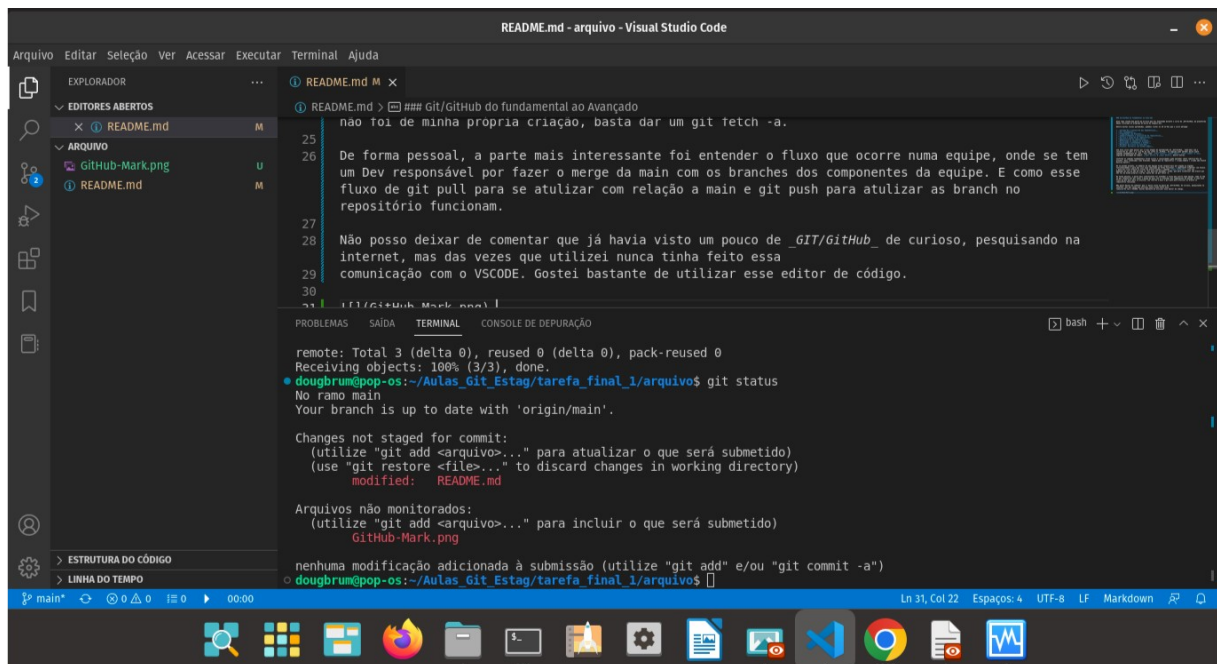
3. Escolher um software editor de código de sua preferência e integrar ao GitHub. Alguns exemplos são VSCODE, Atom e sublime.

Conforme mostrado no print anterior utilizei o VSCode como editor de códigos.

4. Criar um Markdown chamado README.md e adicione aos seu repositório. Em seu conteúdo, descrever o que estiver aprendendo, de maneira estruturada, utilizando o editor escolhido no passo anterior.

A estrutura do documento é livre. Cada aluno poderá construí-lo à sua maneira. Seja criativo, busque exemplos e/ou inspire em outro projetos.

Podemos começar a fazer mudanças no arquivo README.md e em seguida dar um git status para visualizar os arquivos untracked e os modified. Nesse momento podemos dar um git add . para tirarmos os arquivos do status untracked e em seguida commitar os arquivos com o comando git commit -a -m “juntamente com a mensagem desejada”.



The screenshot shows the Visual Studio Code interface with the README.md file open. The file content is as follows:

```
1  ## GitHub do fundamental ao Avançado
2  não foi de minha própria criação, basta dar um git fetch -a.
3
4  De forma pessoal, a parte mais interessante foi entender o fluxo que ocorre numa equipe, onde se tem
5  um Dev responsável por fazer o merge da main com os branches dos componentes da equipe. E como esse
6  fluxo de git pull para se atualizar com relação a main e git push para atualizar as branch no
7  repositório funcionam.
8
9  Não posso deixar de comentar que já havia visto um pouco de _GIT/GitHub_ de curioso, pesquisando na
10 internet, mas das vezes que utilizei nunca tinha feito essa
11 comunicação com o VSCODE. Gostei bastante de utilizar esse editor de código.
```

The terminal output shows the result of the `git status` command:

```
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
doughbrum@pop-os:~/Aulas_Git_Estag/tarefa_final_1/arquivo$ git status
No ramo main
Your branch is up to date with 'origin/main'.

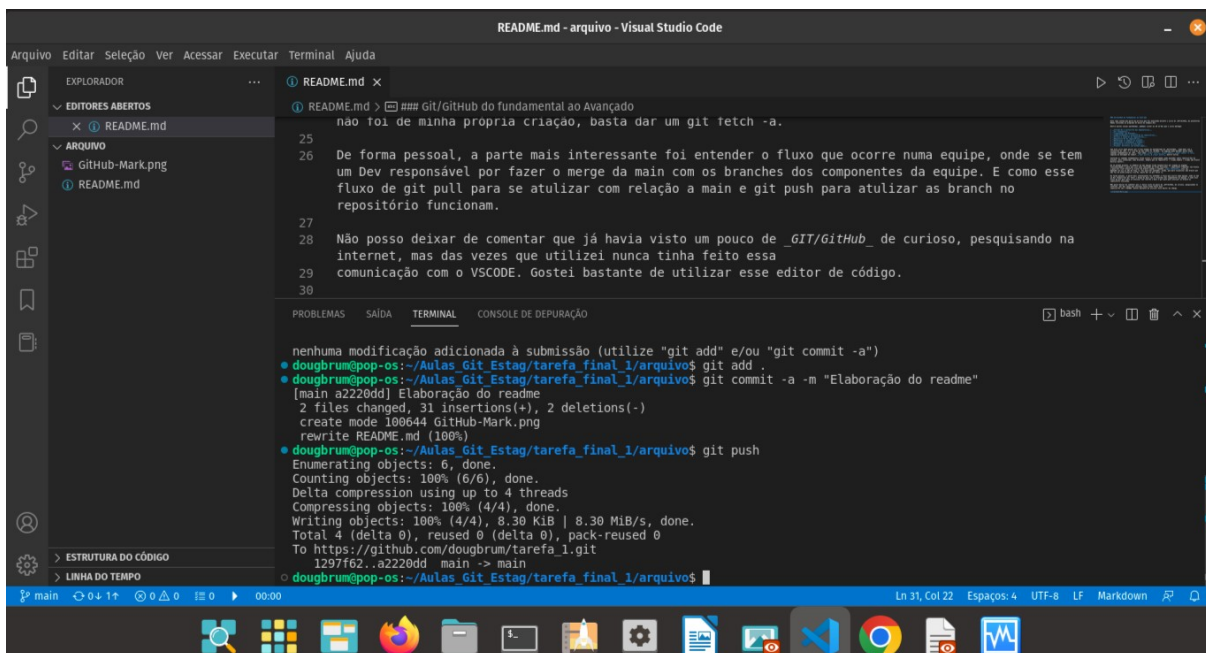
Changes not staged for commit:
  (utilize "git add <arquivo>..." para atualizar o que será submetido)
  (use "git restore <file>..." to discard changes in working directory)
   modified:   README.md

Arquivos não monitorados:
  (utilize "git add <arquivo>..." para incluir o que será submetido)
   GitHub-Mark.png

nenhuma modificação adicionada à submissão (utilize "git add" e/ou "git commit -a")
doughbrum@pop-os:~/Aulas_Git_Estag/tarefa_final_1/arquivo$
```

5. Realizar as operações de commit e push para o GitHub das estruturas de pasta e arquivos criados nos exercícios anteriores.

Tudo feito corretamente precisamos dar um git push para enviar as alterações para o GitHub.



The screenshot shows the Visual Studio Code interface with the README.md file open. The file content is the same as in the previous screenshot. The terminal output shows the result of the `git add`, `git commit`, and `git push` commands:

```
nenhuma modificação adicionada à submissão (utilize "git add" e/ou "git commit -a")
doughbrum@pop-os:~/Aulas_Git_Estag/tarefa_final_1/arquivo$ git add .
doughbrum@pop-os:~/Aulas_Git_Estag/tarefa_final_1/arquivo$ git commit -a -m "Elaboração do readme"
[main a2220dd] Elaboração do readme
 2 files changed, 31 insertions(+), 2 deletions(-)
 create mode 100644 GitHub-Mark.png
 rewrite README.md (100%)
doughbrum@pop-os:~/Aulas_Git_Estag/tarefa_final_1/arquivo$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 8.30 KiB | 8.30 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/doughbrum/tarefa_1.git
 1297f62..a2220dd  main -> main
doughbrum@pop-os:~/Aulas_Git_Estag/tarefa_final_1/arquivo$
```