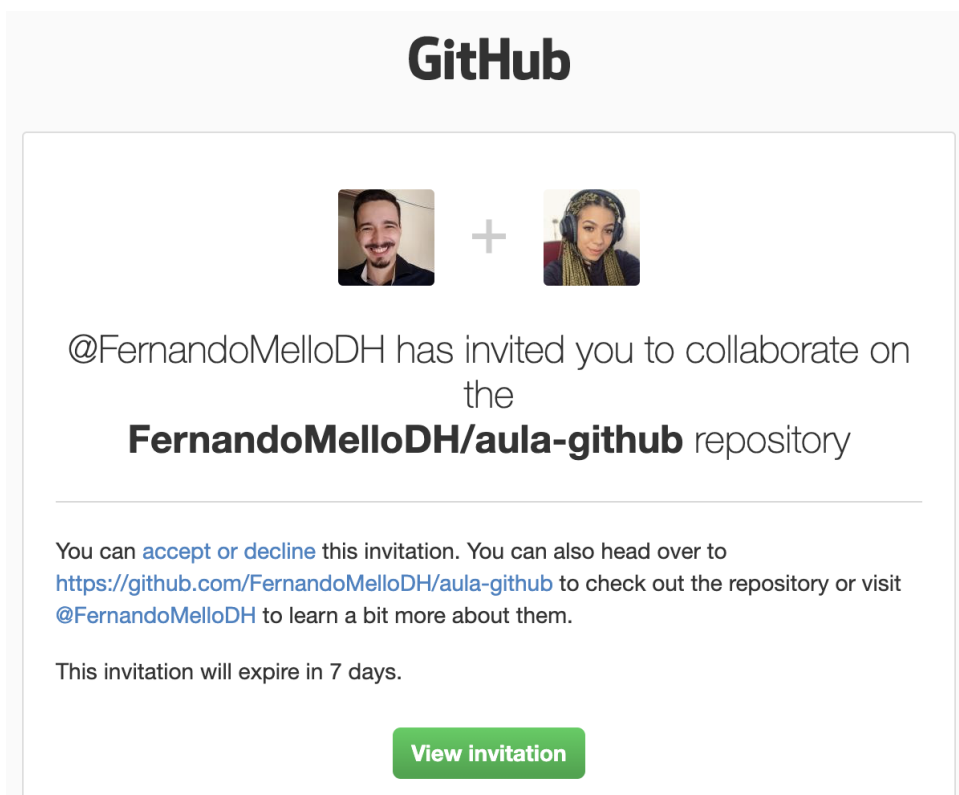


Git + GitHub

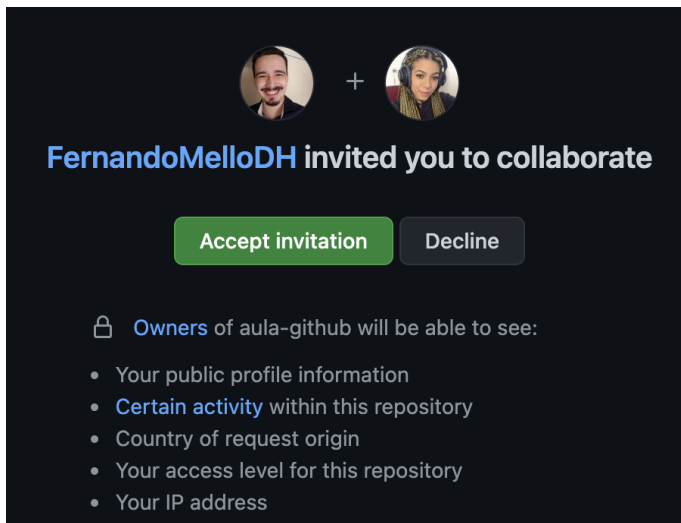
Guia passo a passo para trabalhar na Mochila do Viajante.

Primeiros passos

1. A primeira coisa que você precisa fazer é fornecer seu nome de usuário a quem for trabalhar com você em um repositório remoto, por exemplo seu professor ou um colega de sala, para que ele possa adicioná-lo como um colaborador ao repositório. O convite pode ser feito através do menu Settings do repositório.
2. Depois de ter sido adicionado como colaborador, você receberá um e-mail com um convite semelhante ao da imagem a seguir:

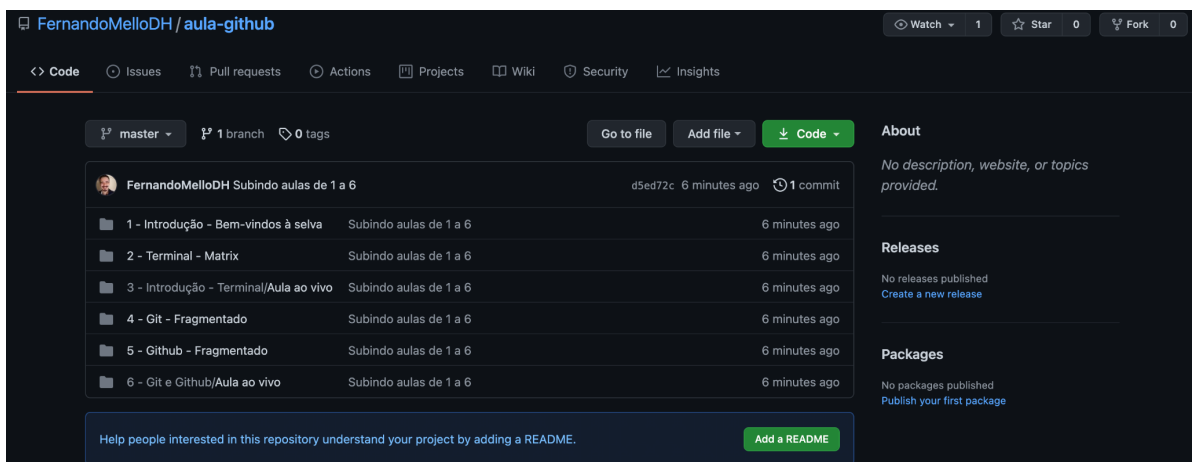


3. Clique no botão "View invitation" e você será redirecionado ao site Github para que possa aceitar o convite:



É importante que antes desta etapa você já tenha se logado no github com seu nome de usuário e senha. Caso não tenha, será necessário realizar o processo de login.

4. Depois de aceitar o convite, aparecerá o repositório correspondente:



Antes de clonar o repositório

5. Verifique se o seu nome de usuário e e-mail estão configurados no Git utilizando os seguintes comandos no seu Terminal:

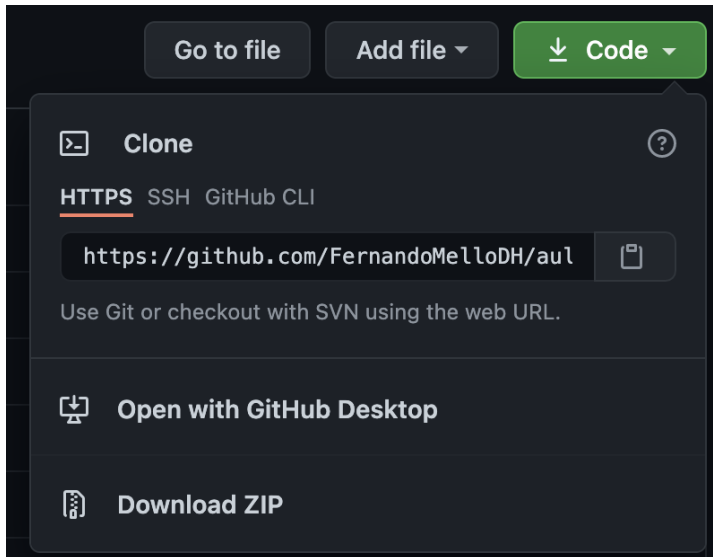
```
git config user.name
git config user.email
```


6. Se não tiver sido configurados, é hora de você fazê-lo utilizando os seguintes comandos em seu Terminal:

```
git config user.name "nome_usuario"
git config user.email "email@email.com"
```

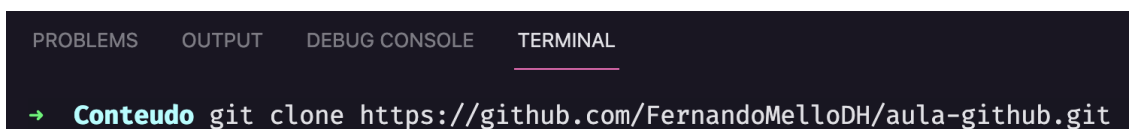
Clonando o repositório no meu computador

- Então, para clonar o repositório em seu computador, você deve clicar no botão **Code** localizado na margem superior. Lá, uma janela será exibida com o URL correspondente ao repositório. É muito importante que a opção **HTTPS** esteja sublinhada (verifique se a opção SSH não está sublinhada).



- Copie o endereço clicando no ícone da pasta: 
- Em seu computador, escolha uma pasta para clonar o repositório. Pode ser a pasta da matéria, pois uma vez executado o comando de clonagem será criada uma pasta dentro dela. Você não precisa inicializar essa pasta com git (o comando `git init` não precisa ser utilizado neste caso).
- Agora você pode clonar!

Execute o comando `git clone "url_do_repositorio"` na pasta escolhida:



- Se esta é a primeira vez que você está clonando um repositório, o Github pedirá que você faça o processo de login. Nesse caso, você deve inserir seu nome de usuário e senha onde for indicado ou será redirecionado para uma página onde você deve aceitar a conexão.
- Depois de clonado, uma sequência de linhas de comandos como essa aparecerá no seu Terminal:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

→ Conteudo git clone https://github.com/FernandoMelloDH/aula-github.git
Cloning into 'aula-github'...
remote: Enumerating objects: 110, done.
remote: Counting objects: 100% (110/110), done.
remote: Compressing objects: 100% (72/72), done.
remote: Total 110 (delta 36), reused 110 (delta 36), pack-reused 0
Receiving objects: 100% (110/110), 80.85 MiB | 142.00 KiB/s, done.
Resolving deltas: 100% (36/36), done.
→ Conteudo
```

13. Agora você pode acessar a Mochila! Se você executar o comando ls, notará que haverá uma nova pasta com o nome do repositório clonado. Para acessá-lo, basta trocar de pasta e posicionar-se na nova:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

→ Conteudo cd aula-github
→ aula-github git:(master) x
```

14. Você perceberá que está na pasta correta porque ao lado do nome da mochila o branch principal “main ou master” aparecerá em uma cor de destaque diferente da usada nas demais letras dentro de parênteses.

Se você listar as pastas, aparecerão todas as pastas das aulas onde você deve, a partir de agora fazer o upload das partes práticas realizadas:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

→ aula-github git:(master) x ls
1 - Introdução - Bem-vindos à selva      4 - Git - Fragmentado
2 - Terminal - Matrix                    5 - Github - Fragmentado
3 - Introdução - Terminal                 6 - Git e Github
→ aula-github git:(master) x
```

Criando uma branch e enviando os exercícios

15. Para criar uma branch usamos o comando

```
git branch <nome-branch>
```

Para a Mochila vamos criar uma branch com nosso sobrenome e nome:

Exemplo:

```
git branch Potter_Harry
```

Em seguida, mudamos para esse ramo usando o comando

```
git checkout <nome-branch>
```

No nosso caso:

```
git checkout Potter_Harry
```

16. Podemos visualizar que estamos em nosso ramo porque em vez de (main) nosso sobrenome e nome aparecerão:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

→ aula-github git:(master) x git branch Potter_Harry
→ aula-github git:(master) x git checkout Potter_Harry
Switched to branch 'Potter_Harry'
→ aula-github git:(Potter_Harry) x
```

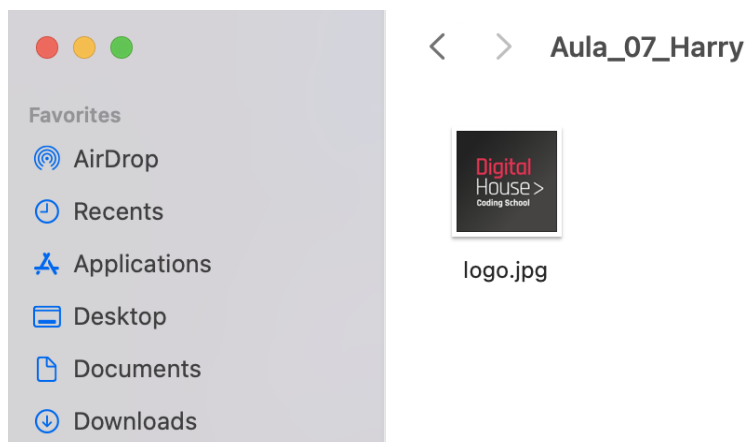
17. Entramos na aula correspondente do dia. Se um determinado nome não for especificado, criaremos a pasta seguindo o exemplo a seguir:

Aula_X_Assunto

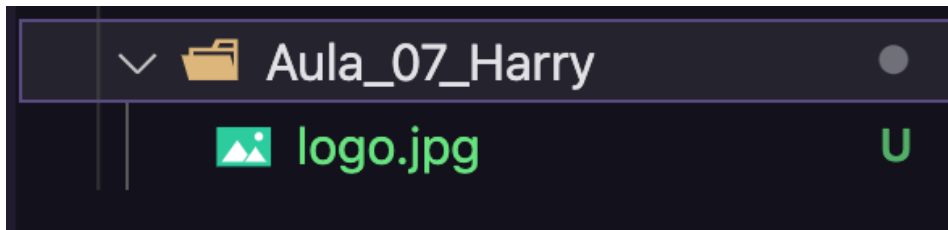
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

→ aula-github git:(Potter_Harry) x mkdir Aula_07_Harry
→ aula-github git:(Potter_Harry) x
```

Lá iremos criar ou fazer upload do conteúdo trabalhado nas aulas. Se for um arquivo .jpg, .png ou .pdf, você pode copiá-lo diretamente do explorador de arquivos:



18. No código do Visual Studio, parecerá que temos um novo arquivo não rastreado:



19. Se executarmos o comando `git status`, a nova pasta criada aparecerá pronta para adicionar, além de outras alterações:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
→ aula-github git:(Potter_Harry) * git status
On branch Potter_Harry
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .DS_Store
    "1 - Introdu\303\247\303\243o - Bem-vindos \303\240 selva/Aula virtual/1. Defini\303\247\303\26
5es e Objetivos da Disciplina/Semana-t\303\255pica.jpg"
    "1 - Introdu\303\247\303\243o - Bem-vindos \303\240 selva/Aula virtual/1. Defini\303\247\303\26
5es e Objetivos da Disciplina/Semana-t\303\255pica.psd"
    Aula_07_Harry/

nothing added to commit but untracked files present (use "git add" to track)
→ aula-github git:(Potter_Harry) *
```

20. É hora de fazer o `git add .` e o `commit`

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
→ aula-github git:(Potter_Harry) * git add .
→ aula-github git:(Potter_Harry) * git commit -m "adicionando alterações"
[Potter_Harry 0242f8e] adicionando alterações
4 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 .DS_Store
create mode 100644 "1 - Introdu\303\247\303\243o - Bem-vindos \303\240 selva/Aula virtual/1. Defini\30
3\247\303\265es e Objetivos da Disciplina/Semana-t\303\255pica.jpg"
create mode 100644 "1 - Introdu\303\247\303\243o - Bem-vindos \303\240 selva/Aula virtual/1. Defini\30
3\247\303\265es e Objetivos da Disciplina/Semana-t\303\255pica.psd"
create mode 100644 Aula_07_Harry/logo.jpg
→ aula-github git:(Potter_Harry) *
```

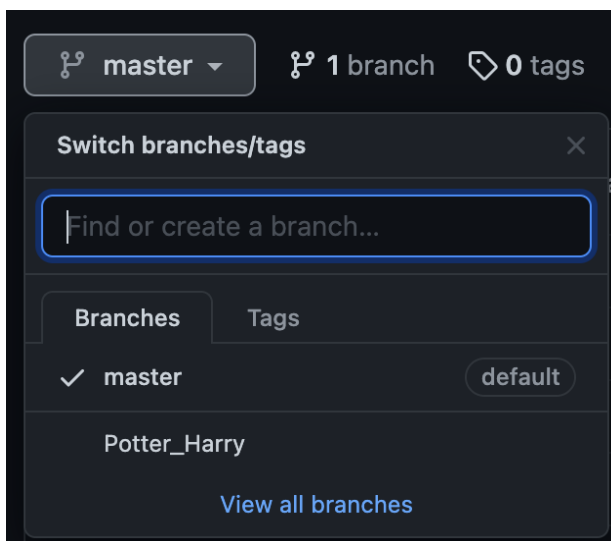
21. Depois de fazer o `commit`, podemos enviar nosso branch com o comando:

```
git push origin <nome-branch>
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

→ aula-github git:(Potter_Harry) git push origin Potter_Harry
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 53.76 KiB | 10.75 MiB/s, done.
Total 8 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
remote:
remote: Create a pull request for 'Potter_Harry' on GitHub by visiting:
remote:   https://github.com/FernandoMelloDH/aula-github/pull/new/Potter_Harry
remote:
To https://github.com/FernandoMelloDH/aula-github.git
 * [new branch]      Potter_Harry -> Potter_Harry
→ aula-github git:(Potter_Harry) █
```

22. Podemos ver no repositório remoto que nossa branch foi criada e está listada ao lado da branch dos nossos colegas:



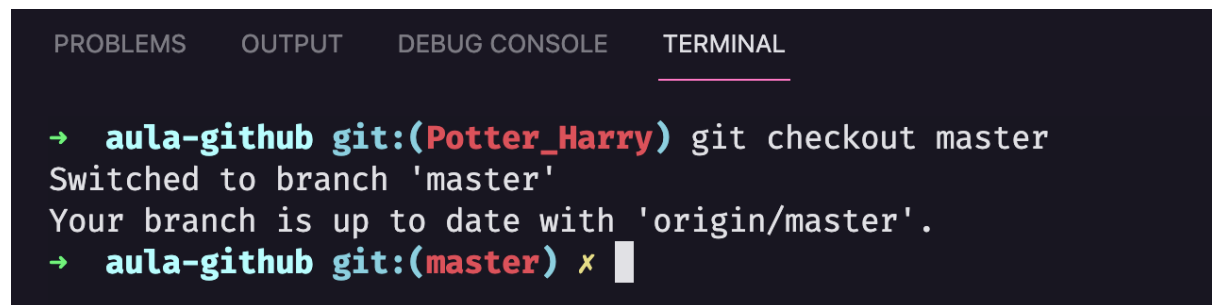
Se eu selecionar o branch, posso ver o último commit que fiz e o arquivo na pasta correspondente à classe:



Fazendo um merge local

23. Depois de fazer o push para meu branch, iremos unir com o branch principal, para isso mudamos de branch com o comando:

```
git checkout main
ou
git checkout master
```



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

→ aula-github git:(Potter_Harry) git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
→ aula-github git:(master) x
```

24. Será conveniente trazermos as alterações do repositório remoto primeiro, antes de fazer um push. Certamente nossos colegas carregaram muitas alterações desde que clonamos:

(Certamente muitas pastas serão carregadas, não se preocupe, é normal)



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

→ aula-github git:(master) x git pull origin master
warning: Pulling without specifying how to reconcile divergent branches is
discouraged. You can squelch this message by running one of the following
commands sometime before your next pull:

    git config pull.rebase false  # merge (the default strategy)
    git config pull.rebase true   # rebase
    git config pull.ff only       # fast-forward only

You can replace "git config" with "git config --global" to set a default
preference for all repositories. You can also pass --rebase, --no-rebase,
or --ff-only on the command line to override the configured default per
invocation.

From https://github.com/FernandoMelloDH/aula-github
* branch          master      -> FETCH_HEAD
Already up to date.
→ aula-github git:(master) x
```

25. Este comando só me trará as mudanças no branch principal, se eu utilizar o comando git status:


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

→ aula-github git:(master) git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
→ aula-github git:(master) █
```

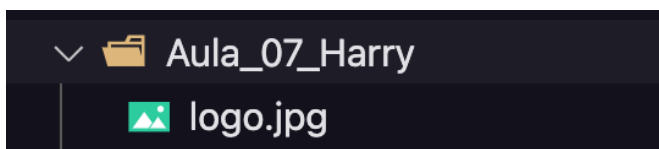
26. É hora do merge! Para isso usamos o comando:

```
git merge <nome-branch>
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

→ aula-github git:(master) git merge Potter_Harry
Already up to date.
→ aula-github git:(master) █
```

27. Agora minha pasta e seu conteúdo estarão na branch main:



28. Se eu executar um git status após fazer o merge, ele só deve me avisar que a branch remota está desatualizada e eu preciso fazer o push do git para as alterações:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

→ aula-github git:(master) git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

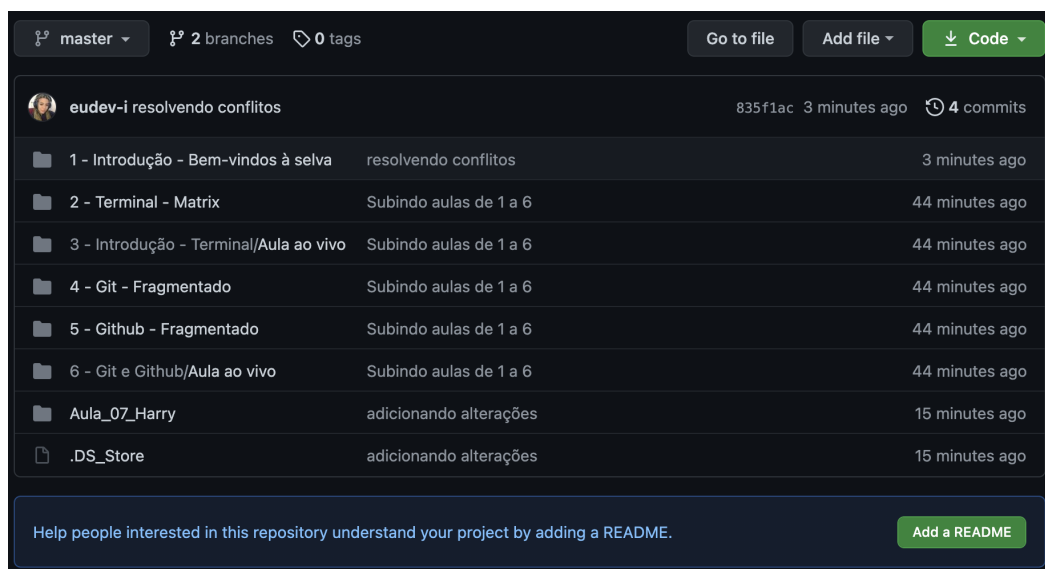
nothing to commit, working tree clean
→ aula-github git:(master) █
```

29. Agora posso atualizar o repositório remoto com minhas alterações já na branch main ou master:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

→ aula-github git:(master) git push origin master
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 479 bytes | 479.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/FernandoMelloDH/aula-github.git
07b690f..835f1ac master -> master
→ aula-github git:(master) █
```

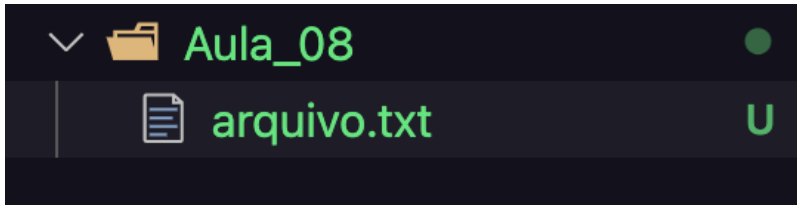
30. Posteriormente, podemos verificar o repositório remoto, minha pasta e seu conteúdo já estará na branch principal:



Combinando com repositório remoto

Existe outra opção de fazer uma fusão diretamente do Github, para isso devemos seguir os seguintes passos:

1. Mudamos o ramo, atualizamos com uma fusão e adicionamos um novo arquivo a uma pasta.



2. Fazemos um git add e um git commit como fizemos anteriormente:

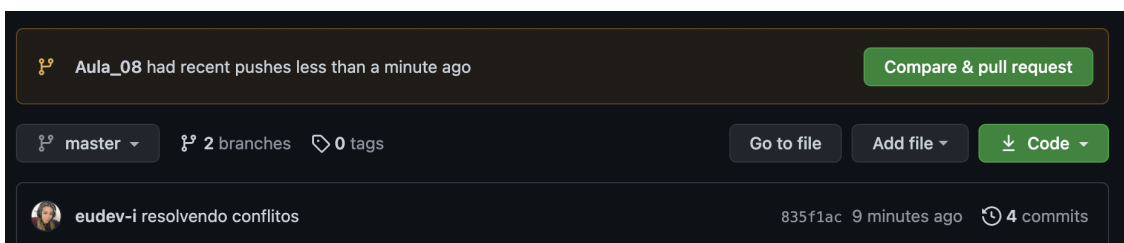
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

→ aula-github git:(Aula_08) git add .
→ aula-github git:(Aula_08) x git commit -m "Adicionando arquivo"
[Aula_08 1841626] Adicionando arquivo
1 file changed, 1 insertion(+)
create mode 100644 Aula_08/arquivo.txt
→ aula-github git:(Aula_08) █
```

3. Agora enviamos nossas alterações através do comando git push a partir de nossa branch:

```
→ aula-github git:(Aula_08) git push origin Aula_08
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 357 bytes | 357.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'Aula_08' on GitHub by visiting:
remote:   https://github.com/FernandoMelloDH/aula-github/pull/new/Aula_08
remote:
To https://github.com/FernandoMelloDH/aula-github.git
* [new branch]      Aula_08 -> Aula_08
→ aula-github git:(Aula_08) █
```

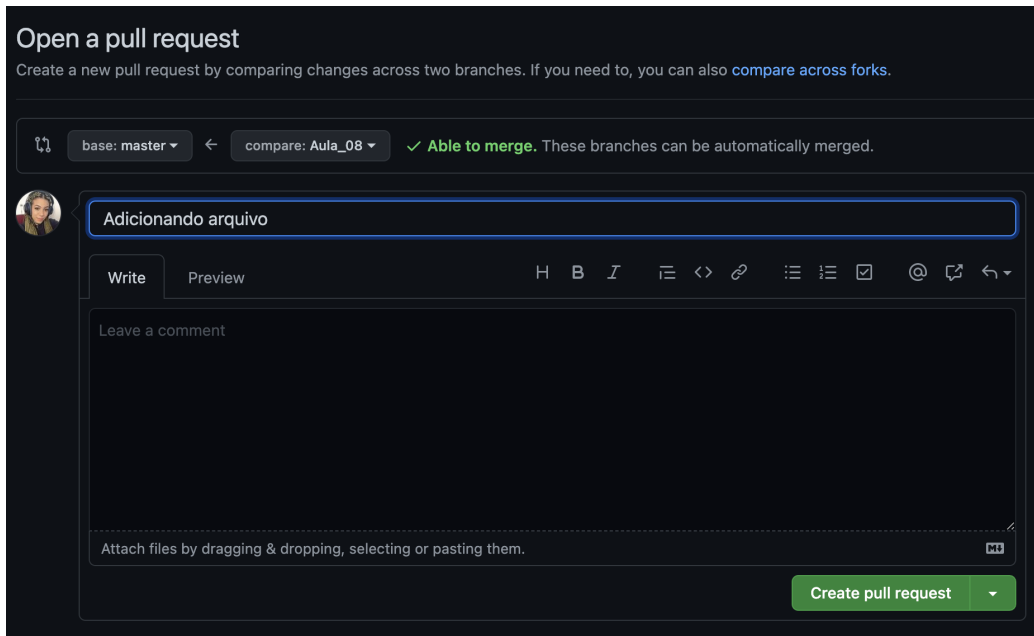
4. Se formos ao Github, a seguinte alteração deve aparecer no repositório:



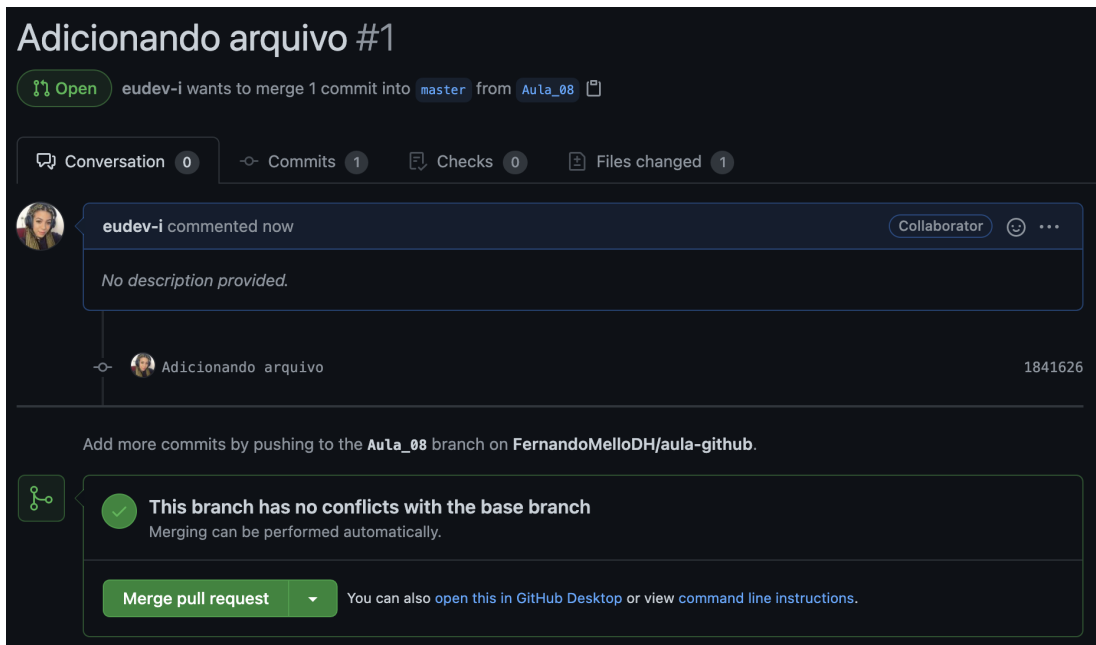
5. Isso significa que minha alteração foi carregada na minha branch e posso fazer o merge. Anteriormente, tínhamos feito esse merge localmente a partir de nosso

terminal no código do Visual Studio, agora vamos tentar fazer o merge a partir do repositório remoto.

Para fazer isso, clicamos no botão **Compare & Pull request**. Lá, entraremos em uma nova janela onde indicará que faremos uma solicitação de pull. A solicitação pull é a ação de validar um código que será unido de uma branch para outra.

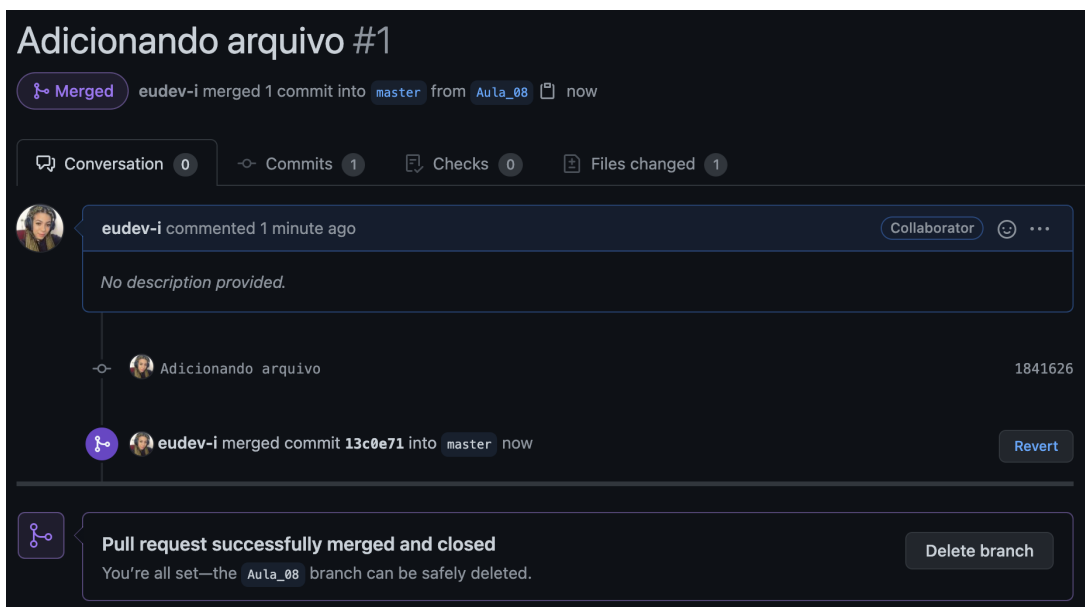


6. Na janela, vemos que nos dá a opção de escolher de qual branch a qual branch eu vou unir, neste caso, deixo as branches padrão como estão. Em seguida, pressiono o botão Create pull Request
7. Uma nova janela aparecerá, que nos permitirá unir a branch:



Se houver conflitos, eles aparecerão e teremos que corrigi-los. Em geral, isso não deveria acontecer, pois estamos trabalhando em nossas próprias pastas.

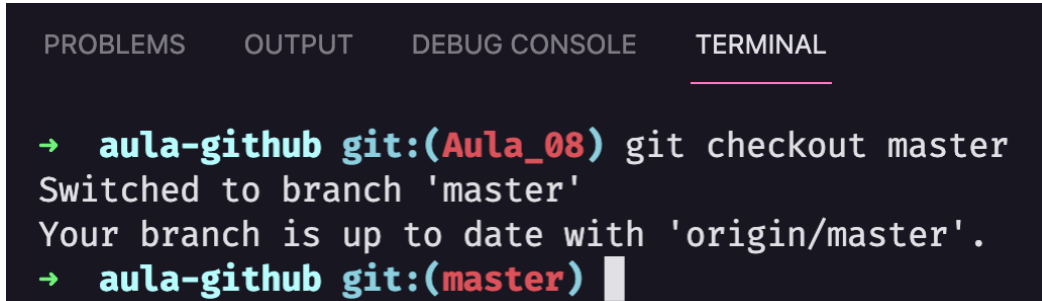
8. Em seguida, pressiono o botão Merge pull request e o botão Merge pull request muda para “confirm merge”.



9. A janela que aparece deverá confirmar que está tudo bem e que unimos nossa branch.

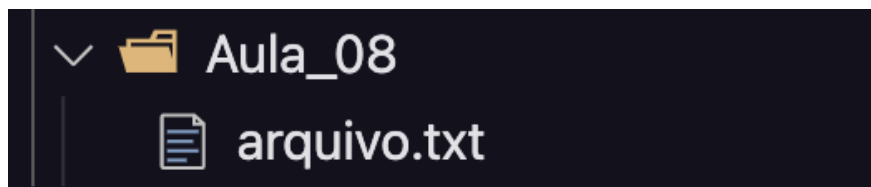
10. Resta mais uma etapa que é fazer o download desse merge para o meu repositório local. No código do Visual Studio, se ainda estiver em meu branch, faço o checkout para a main:

```
git checkout main ou git checkout master
```

A screenshot of the Visual Studio terminal window. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, with TERMINAL selected. The prompt is 'aula-github git:(Aula_08)'. The command 'git checkout master' has been entered, and the output is 'Switched to branch 'master'' and 'Your branch is up to date with 'origin/master''. The prompt is now 'aula-github git:(master)' with a cursor.

```
→ aula-github git:(Aula_08) git checkout master  
Switched to branch 'master'  
Your branch is up to date with 'origin/master'.  
→ aula-github git:(master) █
```

11. Posteriormente, fazemos um git pull das mudanças utilizando git pull origin main.
12. O merge que fizemos no repositório remoto deve aparecer no ramo principal
13. Podemos verificar isso na estrutura visual do arquivo:



14. Muito bem! Agora podemos retornar ao nosso branch e continuar adicionando arquivos seguindo desde a etapa 1 da seção: Adicionando mais arquivos ao meu branch.

Esperamos que este guia o ajude a trabalhar adequadamente com sua mochila!