

**Aluno que executou este roteiro:**

**Aluno que acompanhou a execução deste roteiro:**

**\*Obs.:Sempre que eu escrever “Abra o seu arquivo e verifique a situação do arquivo“, quero que você tire um print e coloque neste documento embaixo da instrução**

**Abra a pasta criada para atividade do Git**

1. git clone “o link do seu repositório no github”

Agora feche o terminal, vá até sua pasta, abra a nova pasta, crie um arquivo de texto, escreva algo nele, salve as alterações e abra o terminal dentro dessa nova pasta 2.

git add “nome do seu arquivo”.”tipo do seu arquivo”

3. git commit -m “Escreva algo”
4. git branch

```
C:\Users\jorge\OneDrive\Área de Trabalho\atividadeGit2>git branch
* master
```

**\*Obs.:Se não existir a branch “main”, faça:  
git branch -M main //Renomear a branch**

```
C:\Users\jorge\OneDrive\Área de Trabalho\atividadeGit2>git branch -M main
```

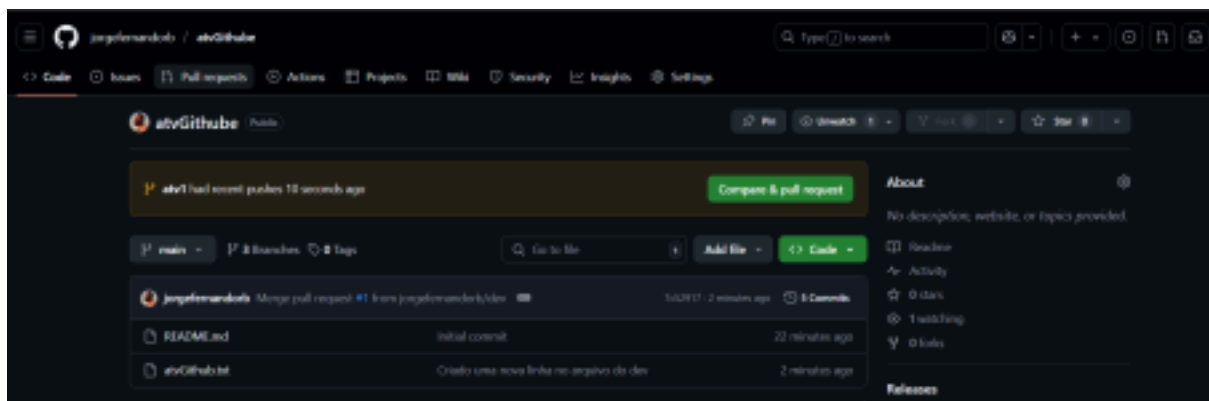
5. git branch dev
6. git switch dev

Agora vá no seu arquivo e escreva algo nele na segunda linha

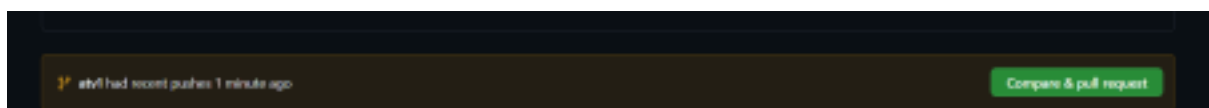
7. git add “nome do seu arquivo”.”tipo do seu arquivo”
8. git commit -m “Escreva algo”
9. git push origin dev

Agora vá no seu repositório no github e atualize a página

Cliquem em Pull Request



Cliquem em Compare & pull request

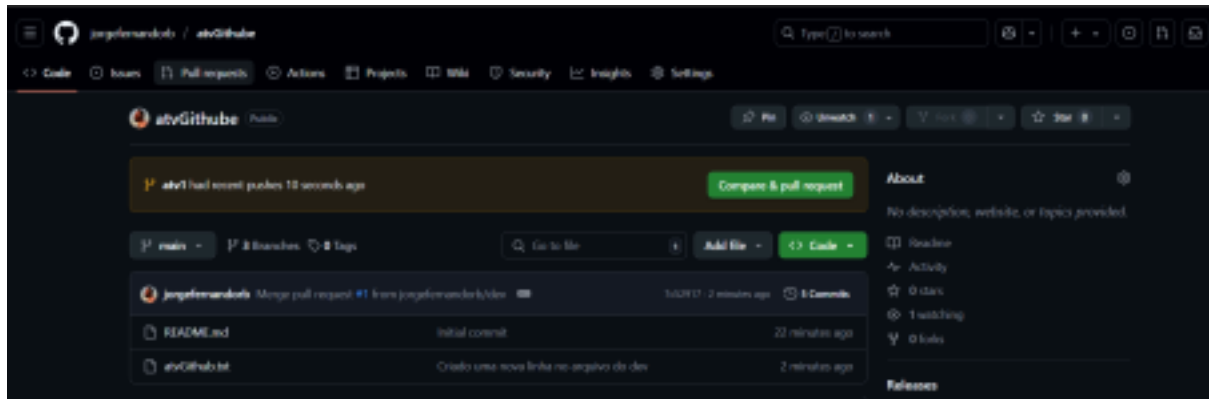


10. git branch atv1
11. git branch atv2
12. git switch atv1

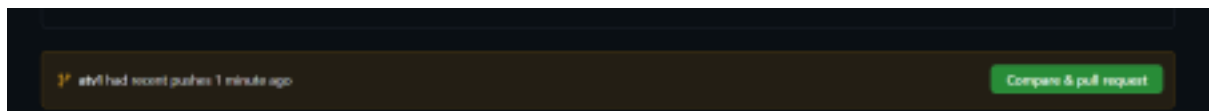
Agora vá no arquivo e altere

13. git add “nome do seu arquivo”.”tipo do seu arquivo”
14. git commit -m “Escreva algo”

15. git push origin dev  
Agora vá no seu repositório no github e atualize a página  
Cliquem em Pull Request

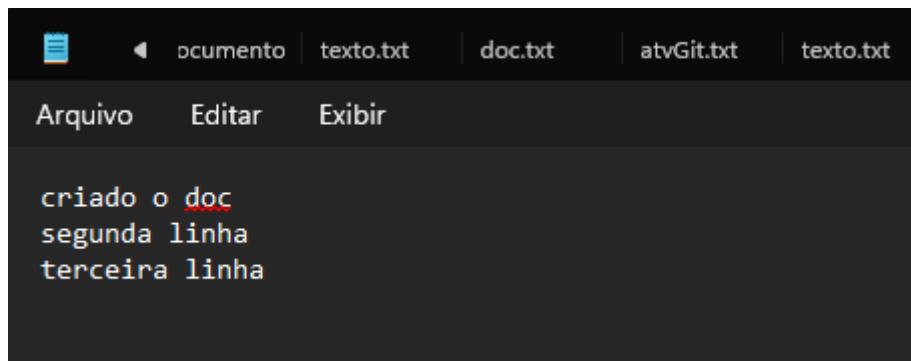


Cliquem em Compare & pull request



- 16. git switch main
- 17. git pull origin main
- Abra o seu arquivo e verifique a situação do arquivo
- 18. git branch **atv1**
- 19. git branch **atv2**
- 20. git branch **atv3**
- 21. git switch **atv1**

**Agora abra seu arquivo, verifique a situação do arquivo e tire um print.**



Altere seu arquivo, escreva algo na terceira linha

- 22. git add "nome do seu arquivo".formato do arquivo" //Ex.: atvGit.txt
- 23. git commit -m "Escreva algo sobre o que você fez"
- 24. git log

```

C:\Users\jorge\OneDrive\Área de Trabalho\atvGit>git switch atv1
Switched to branch 'atv1'

C:\Users\jorge\OneDrive\Área de Trabalho\atvGit>git add .

C:\Users\jorge\OneDrive\Área de Trabalho\atvGit>git commit -m "Altereí a linha 3 na branch atv1"
[atv1 6881edc] Alterei a linha 3 na branch atv1
1 file changed, 2 insertions(+), 1 deletion(-)

C:\Users\jorge\OneDrive\Área de Trabalho\atvGit>git log
commit 6881edc467f6fbbadab822a20b5b64279f02537 (HEAD -> atv1)
Author: Jorge Fernando <jorgefernandorb@gmail.com>
Date: Tue Mar 25 18:09:14 2025 -0300

    Alterei a linha 3 na branch atv1

commit 9606d6e3e407b3dbc7726a832f984c22cb45cc9b (dev, atv2)
Author: Jorge Fernando <jorgefernandorb@gmail.com>
Date: Tue Mar 25 17:43:02 2025 -0300

    Criei a branch dev, fui para ela e escrevi na segunda linha do arquivo de texto

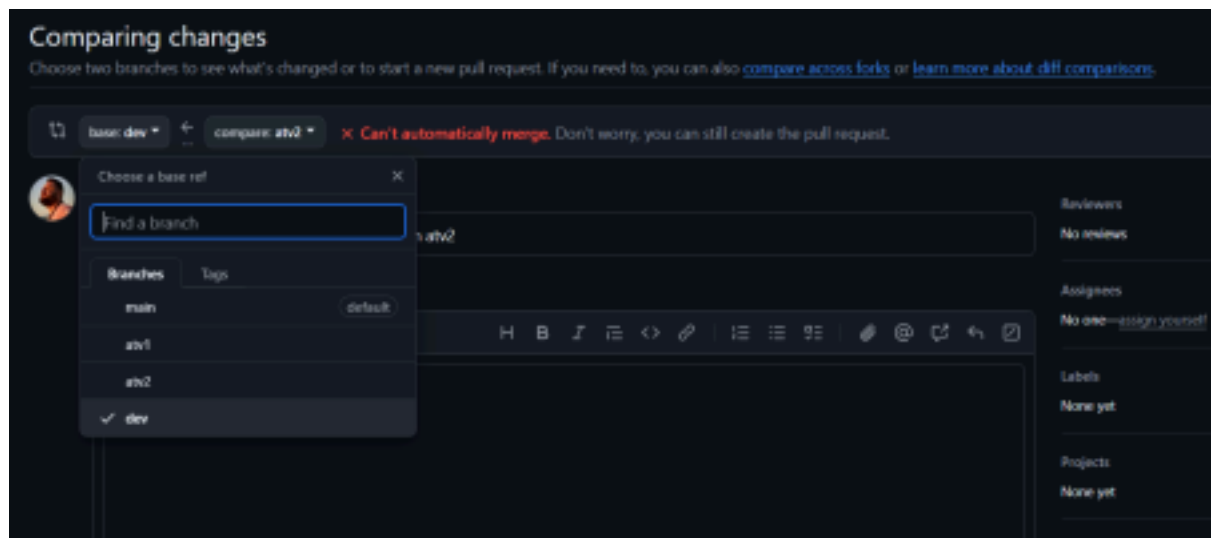
commit adcea9115ad6f656a2b034509d142fde4c8248912 (main)
Author: Jorge Fernando <jorgefernandorb@gmail.com>
Date: Tue Mar 25 17:40:39 2025 -0300

    Criado arquivo atvGit e escrito na primeira linha apenas
C:\Users\jorge\OneDrive\Área de Trabalho\atvGit>

```

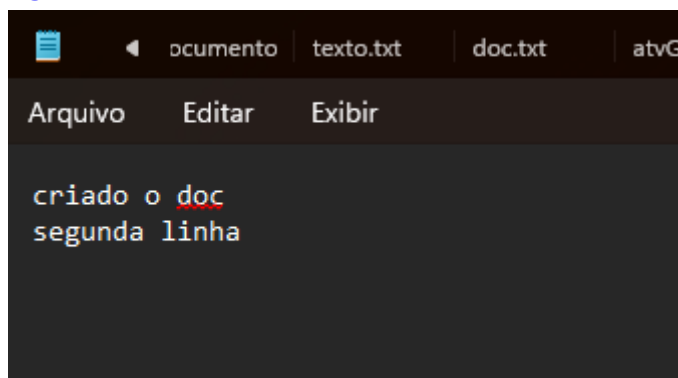
25. git push origin atv1

Vá no seu repositório, aceite a **pr**, faça o **merge** com a **dev**, para isso vá em base e escolha **dev**.



26. git switch atv2

Agora abra seu arquivo, verifique a situação do arquivo e tire um print.



Altere seu arquivo, escreva algo na terceira linha

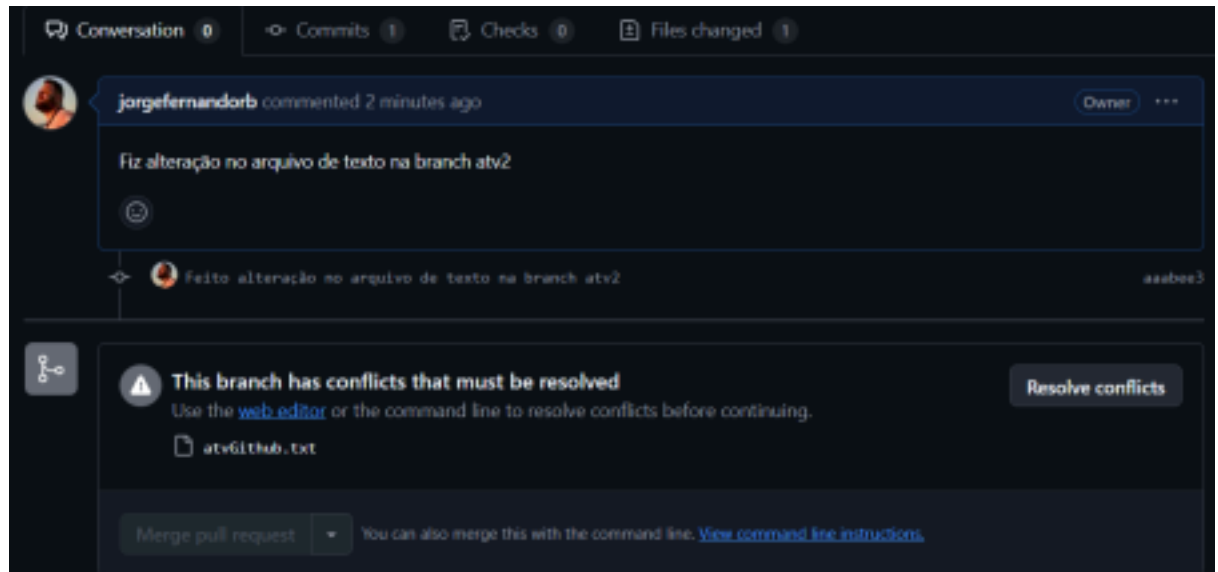
27. git add "nome do seu arquivo".formato do arquivo" //Ex.: atvGit.txt

28. git commit -m "Escreva algo sobre o que vocÊ fez"

29. git push origin atv2

Vá no seu repositório, aceite a **pr**, faça o **merge** com a **dev**, para isso vá em base e escolha **dev**, deu certo? O que aconteceu?

**Resposta:** Sim, deu certo. Comparamos a base dev com a atv2 e os conflitos surgiram, então tivemos que consertar manualmente e, em seguida, clicar em 'mark has resolved'.



Clique em "Resolve conflicts"

Resolva os conflitos

```
<<<<<<< HEAD
```

(conteúdo da branch atual)

```
=====
```

(conteúdo da branch dev)

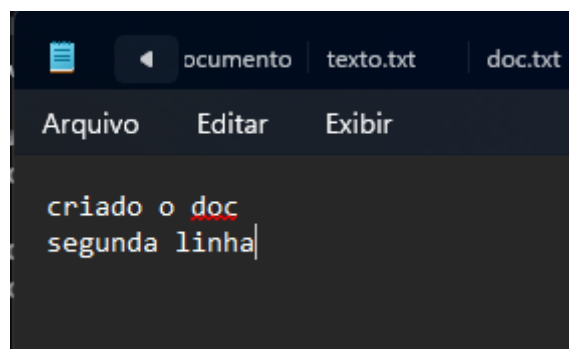
```
>>>>>>> dev
```

Após resolvido, apague: "<<<<<<< HEAD", "======" e ">>>>>>> dev"

Selecione "mark has resolved" ou algo assim (esqueci de tirar o print ^^) e "commit merge" Depois vá no pr e faça o merge

30. git switch dev

**Agora abra seu arquivo, verifique a situação do arquivo e tire um print.**

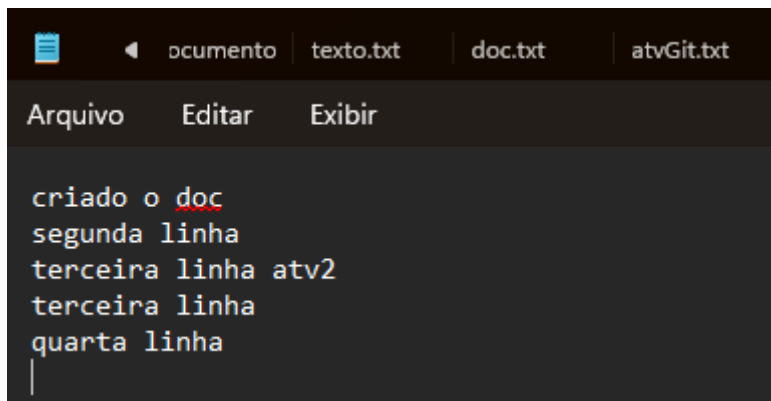


O que aconteceu?

**Resposta:** A branch dev foi aberta na pasta local (git), sem levar em conta o merge realizado na pasta remota (github). Assim, ela ainda mantém apenas a segunda linha que foi adicionada no momento da criação da branch (dev).

31. git pull origin dev

**Agora abra seu arquivo, verifique a situação do arquivo e tire um print.**



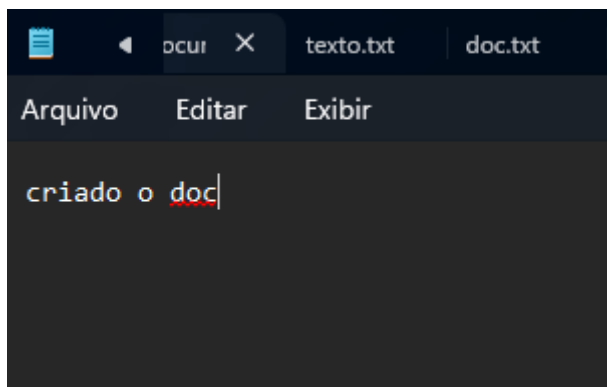
```
criado o doc
segunda linha
terceira linha atv2
terceira linha
quarta linha
|
```

O que aconteceu?

**Resposta:** Quando foi feito o pull (puxar), o merge realizado na pasta remota (github) foi sincronizado com a pasta local (git).

32. git switch atv3

**Agora abra seu arquivo, verifique a situação do arquivo e tire um print.**



```
criado o doc|
```

Por que ele está assim?

**Resposta:** Isso ocorre porque a branch atv3 (localizada na pasta local) foi criada a partir da branch main, mas ainda não foi enviada para a pasta remota.

**Obs.: Sempre que for iniciar o trabalho em uma nova branch, atualize a branch principal de desenvolvimento vá para sua branch e faça o merge**

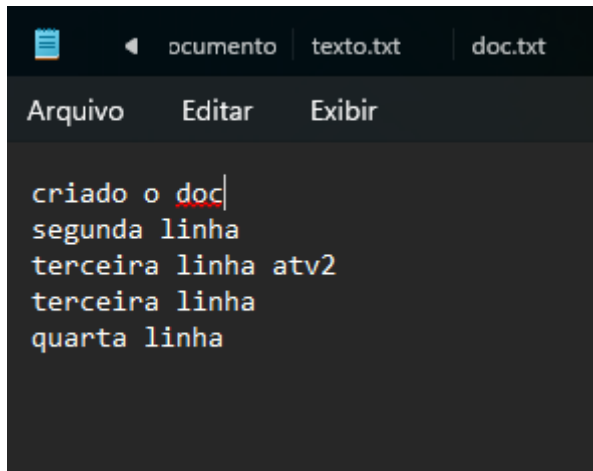
33. git switch dev

34. git pull origin dev

35. git switch atv3

36. git merge dev

Agora abra seu arquivo, verifique a situação do arquivo e tire um print.



```
criado o doc
segunda linha
terceira linha atv2
terceira linha
quarta linha
```

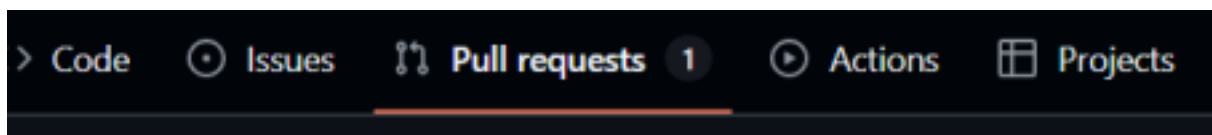
Crie um novo arquivo de texto na pasta

37. git add "nome do seu arquivo".formato do arquivo" //Ex.:  
atvGit.txt
38. git commit -m "Escreva algo sobre o que você fez"
39. git log
40. git push origin atv3

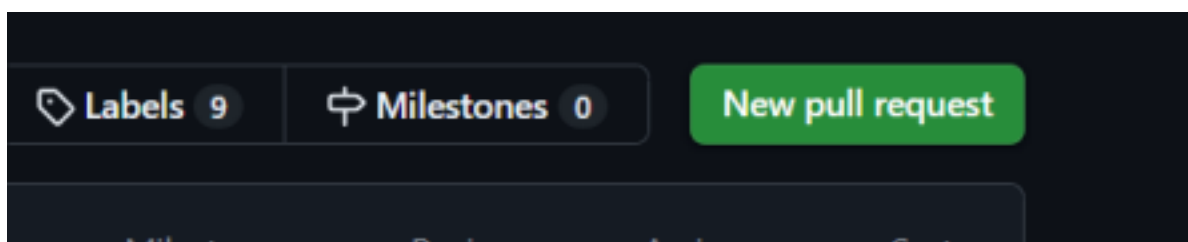
Vá no seu repositório, aceite a **pr**, faça o **merge** com a **dev**, para isso vá em base e escolha **dev**.

Agora abra sua branch main no seu repositório no git hub ?

Selecione "pull request" na barra do seu repositório

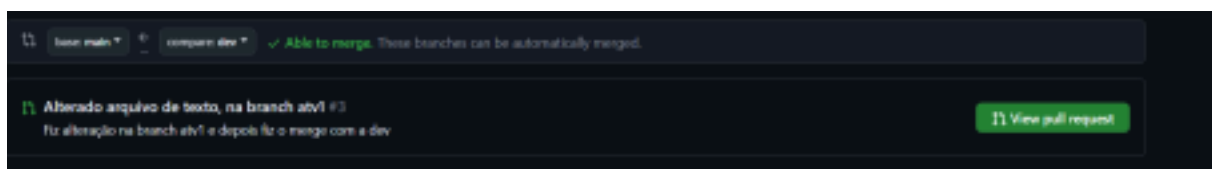


Depois "new pull request"

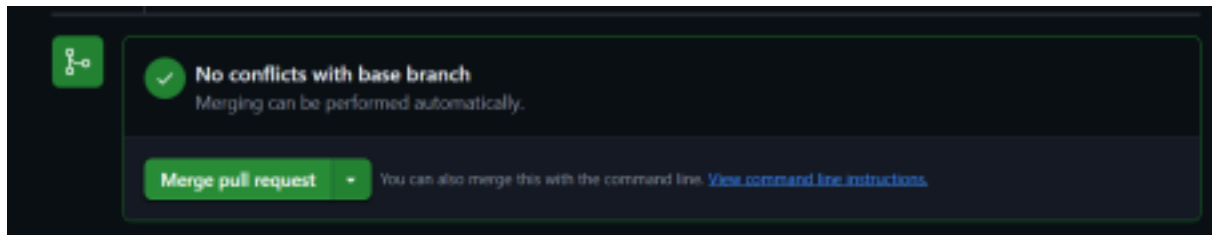


Selecione a base "main" e compare "dev"

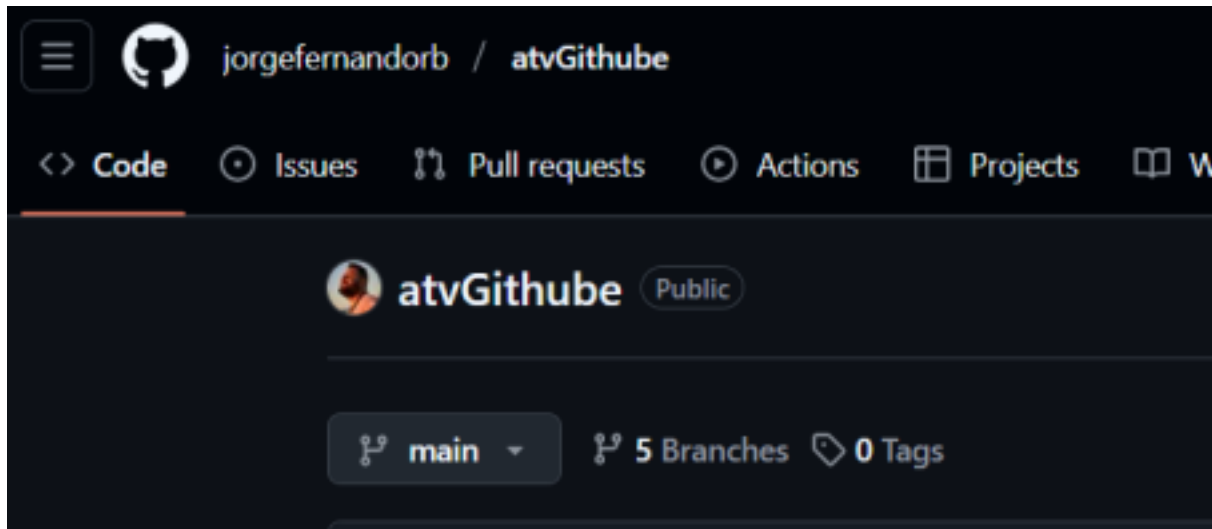
Depois "view pull request"



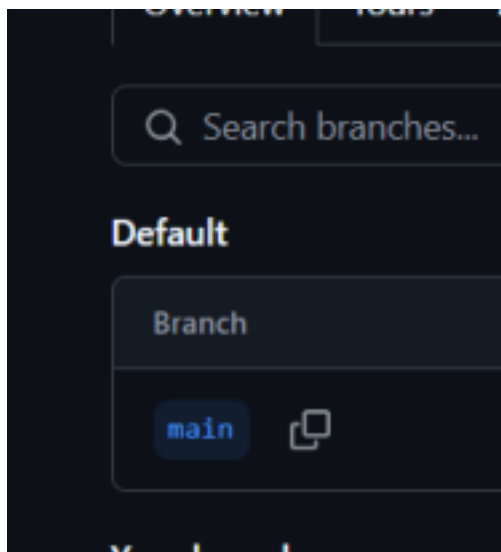
Depois "merge pull request"



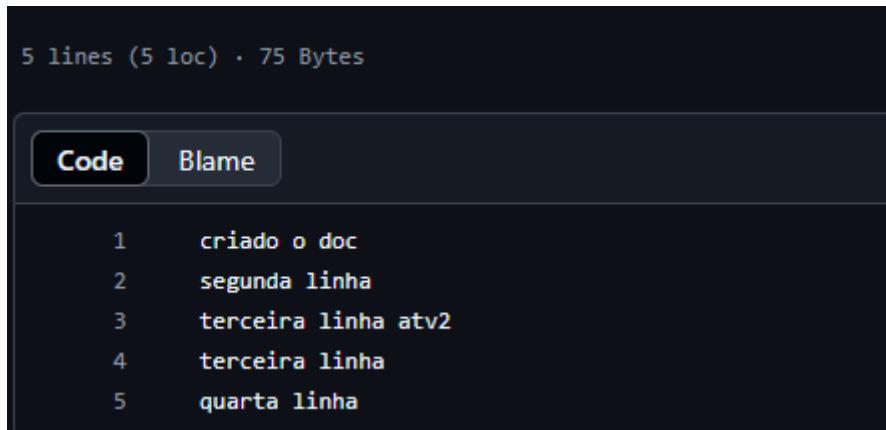
Agora code e depois branch



Selecione a branch main



Agora abra seu arquivo, verifique a situação do arquivo e tire um print.



The screenshot shows a code editor interface with a dark background. At the top, it says "5 lines (5 loc) · 75 Bytes". Below this, there are two tabs: "Code" (selected) and "Blame". The code content is as follows:

```
1   criado o doc
2   segunda linha
3   terceira linha atv2
4   terceira linha
5   quarta linha
```

Volte para o terminal

41. git switch main

Qual a situação da pasta?

**Resposta:** A pasta contém apenas o arquivo doc criado originalmente na pasta local, sem refletir o merge realizado na pasta remota, que inclui tanto o arquivo doc quanto o novo arquivo de texto.

Qual a situação do arquivo?

**Resposta:** O arquivo doc permanece inalterado, pois a branch main na pasta local ainda não foi sincronizada com a pasta remota.

42. git pull origin main

Qual a situação da pasta?

**Resposta:** Após o pull, a pasta local contém tanto o arquivo criado anteriormente quanto o novo, pois houve a fusão com a pasta remota.

Qual a situação do arquivo?

**Resposta:** O arquivo local agora contém as linhas que foram incluídas no merge realizado no arquivo remoto.