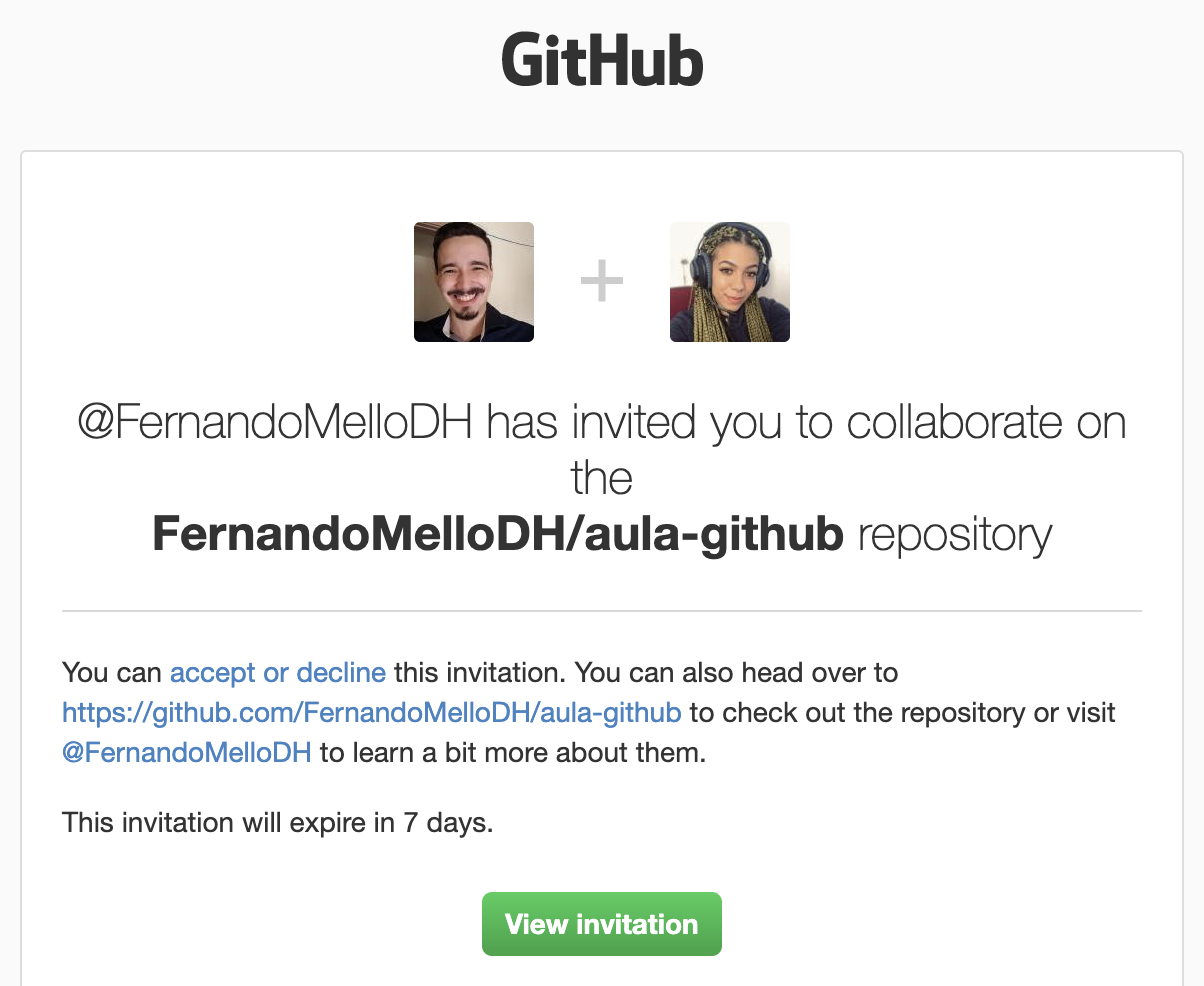
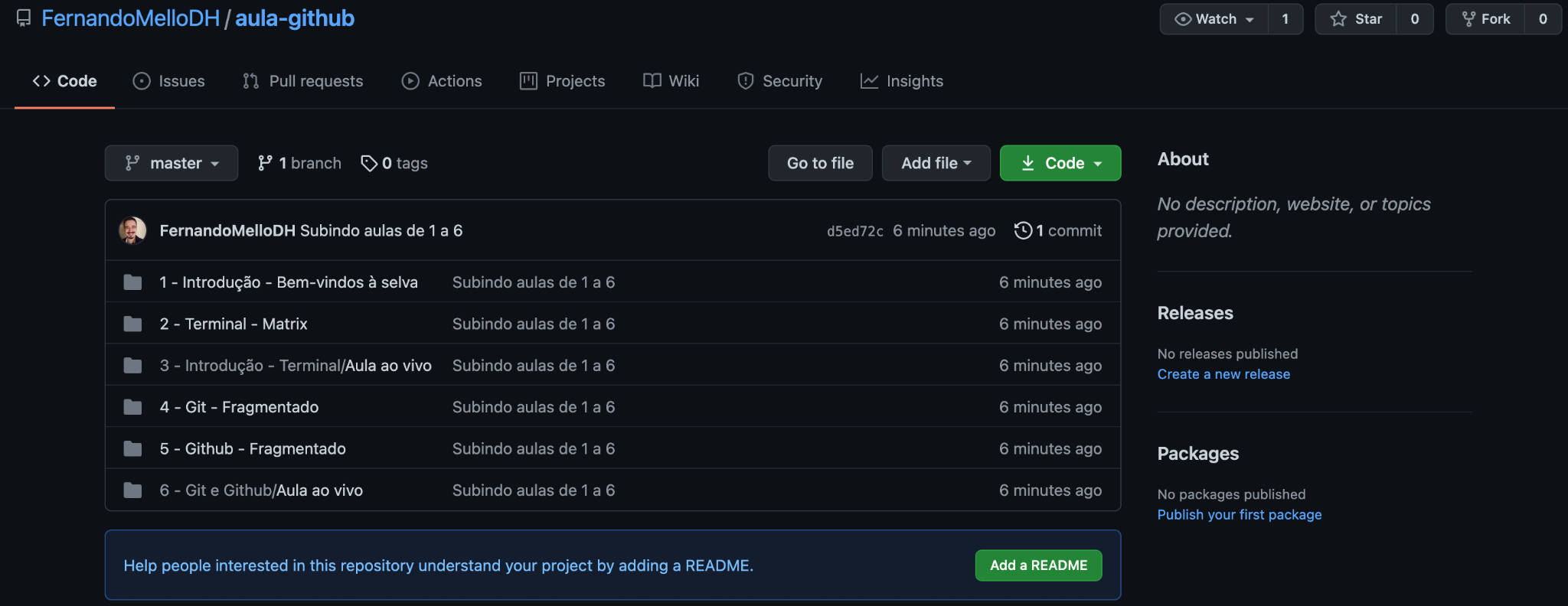
Introdução a Informática

**Git + GitHub**

Guia passo a passo para trabalhar na Mochila do Viajante.

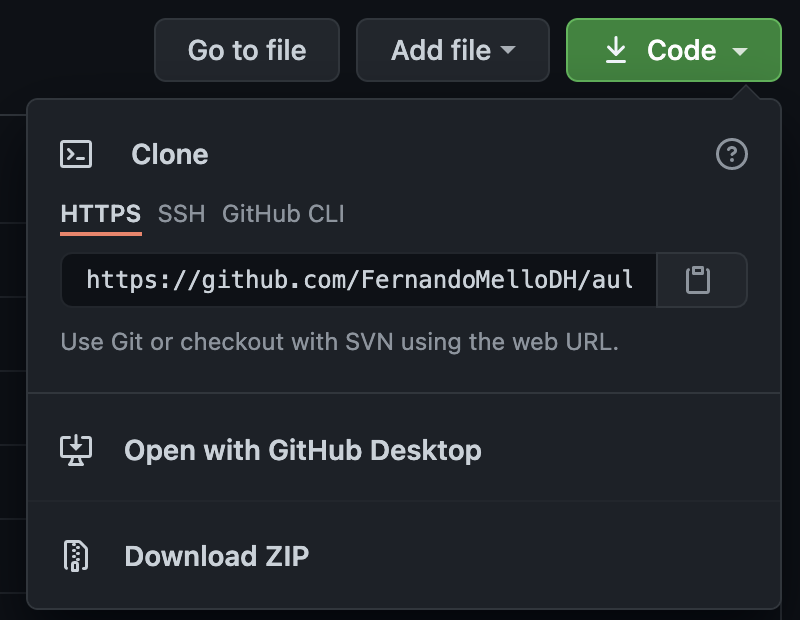
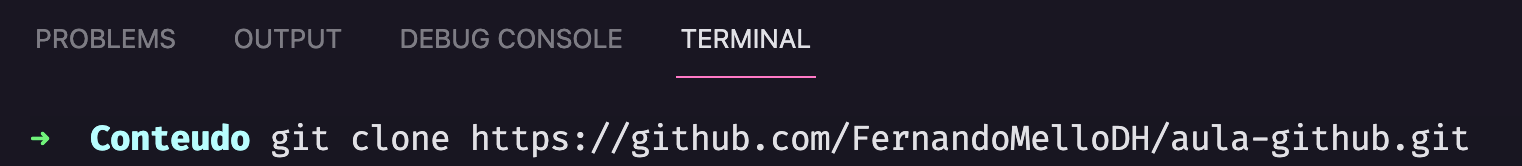
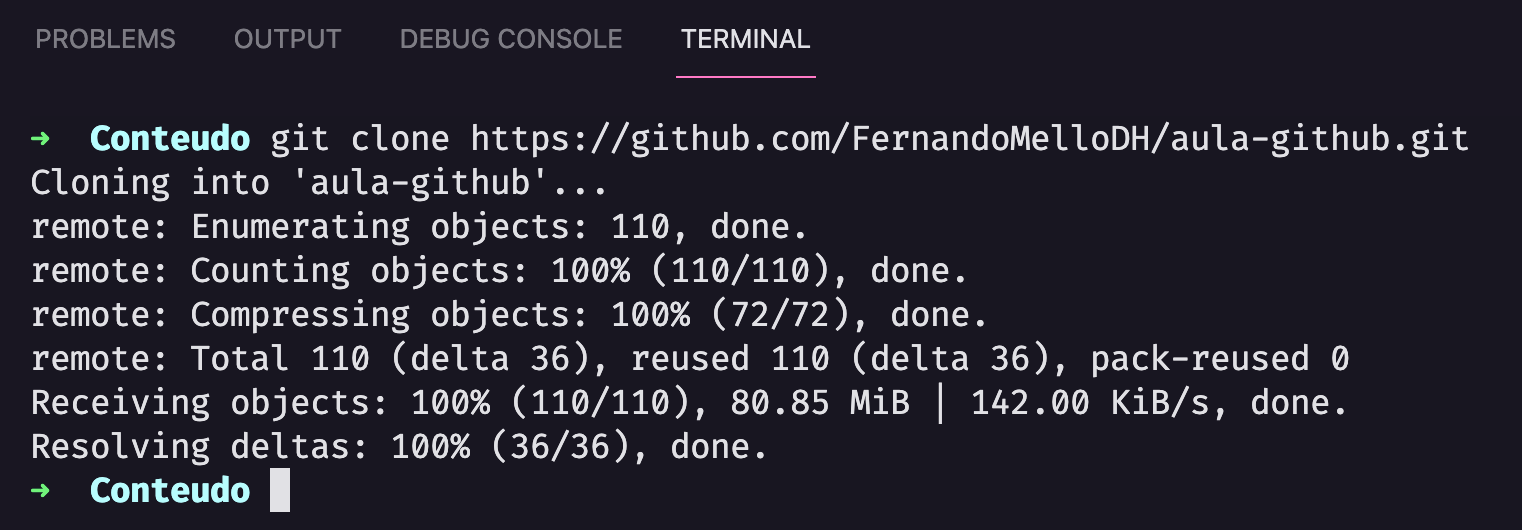
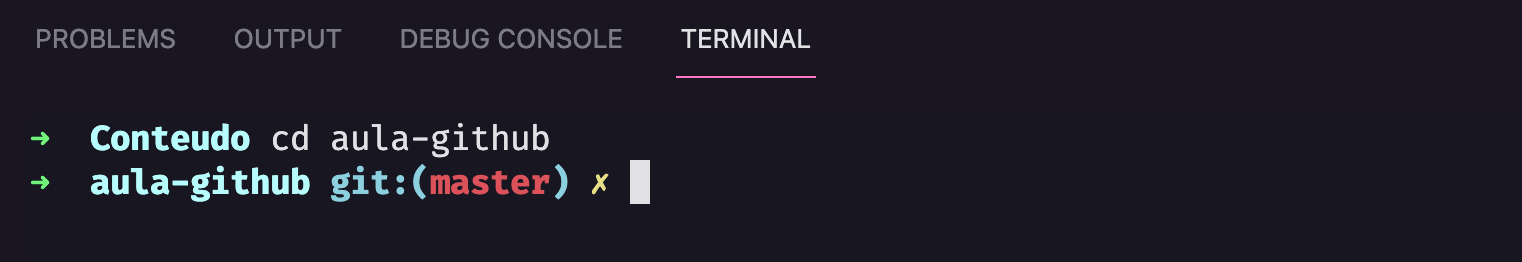
# Primeiros passos

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

1. 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
2. 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

1. 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).  
   
2. Copie o endereço clicando no ícone da pasta: 
3. 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).
4. Agora você pode clonar!  
   Execute o comando git clone "url\_do\_repositorio" na pasta escolhida:  
   
5. 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.
6. Depois de clonado, uma sequência de linhas de comandos como essa aparecerá no seu Terminal:  
   
7. 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:
8. 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:

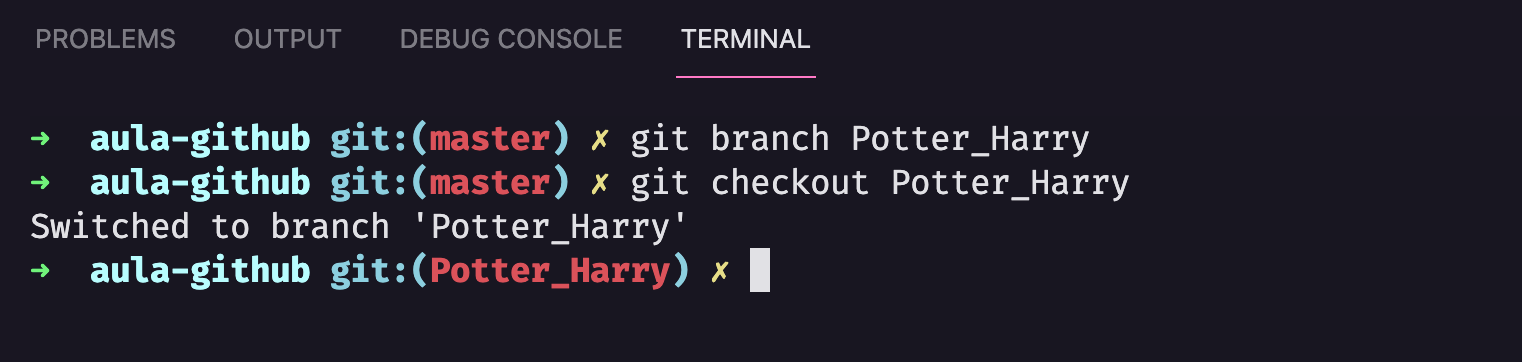
# Criando uma branch e enviando os exercícios

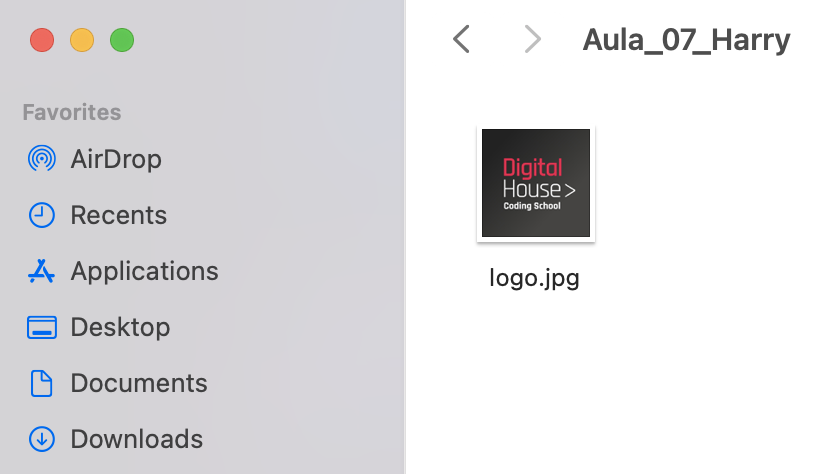
1. Para criar uma branch usamos o comando   
   git branch <nome-branch>

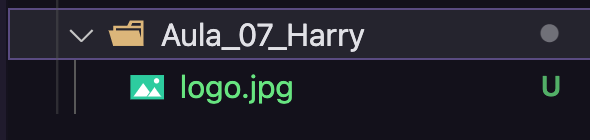
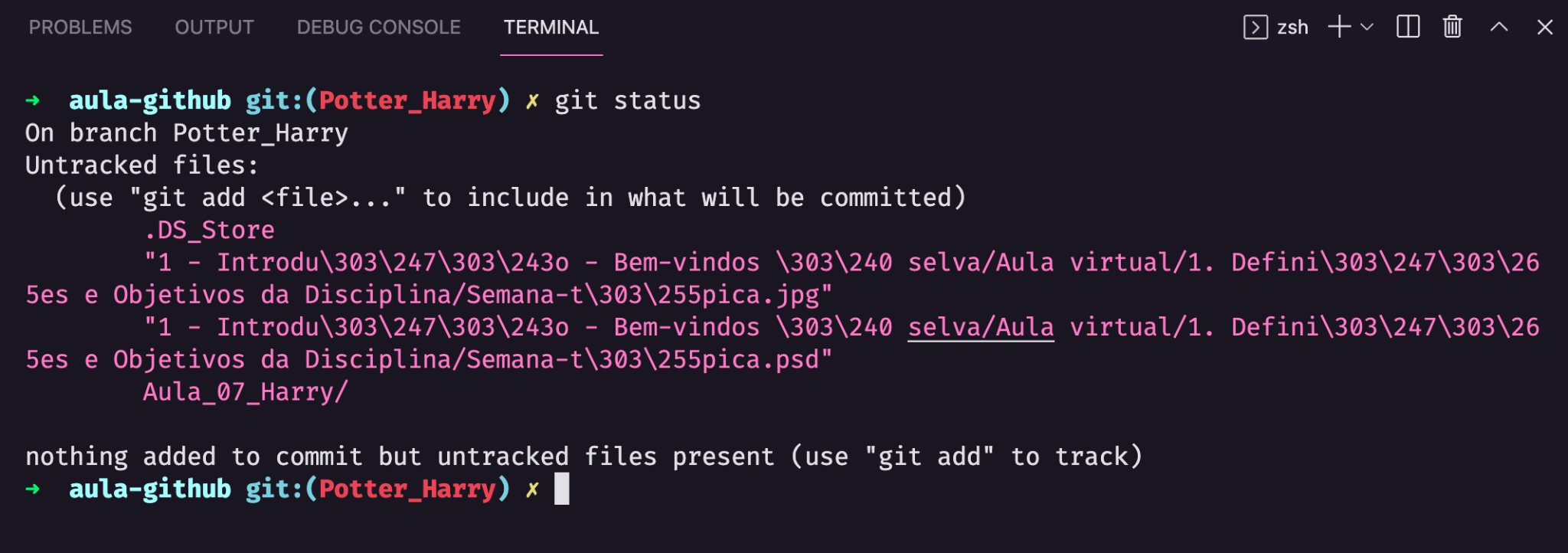
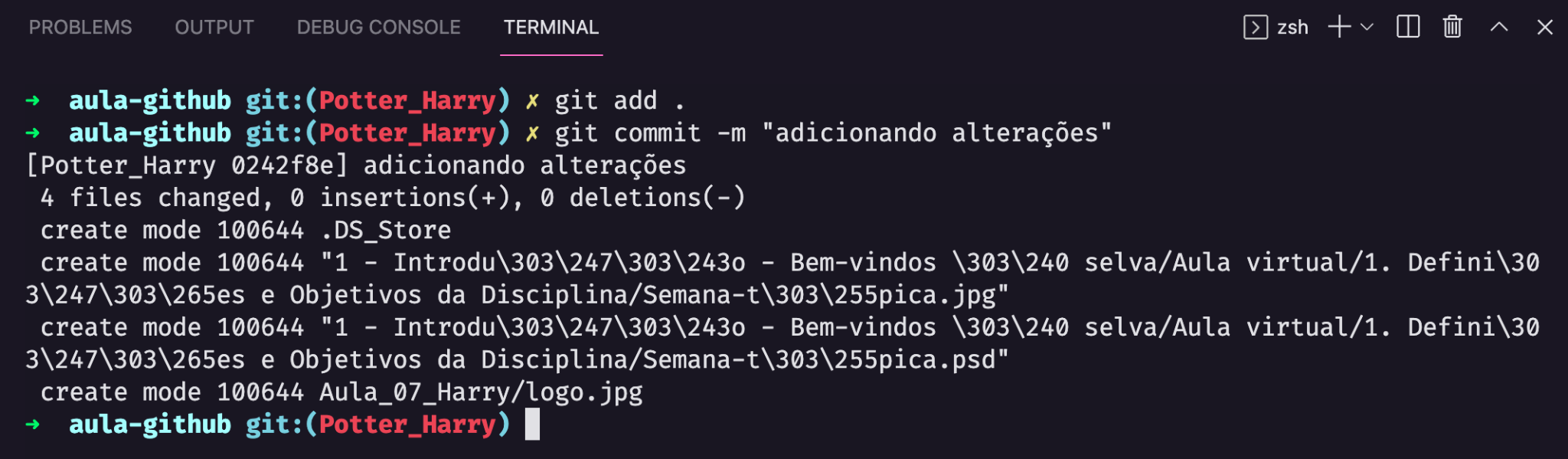
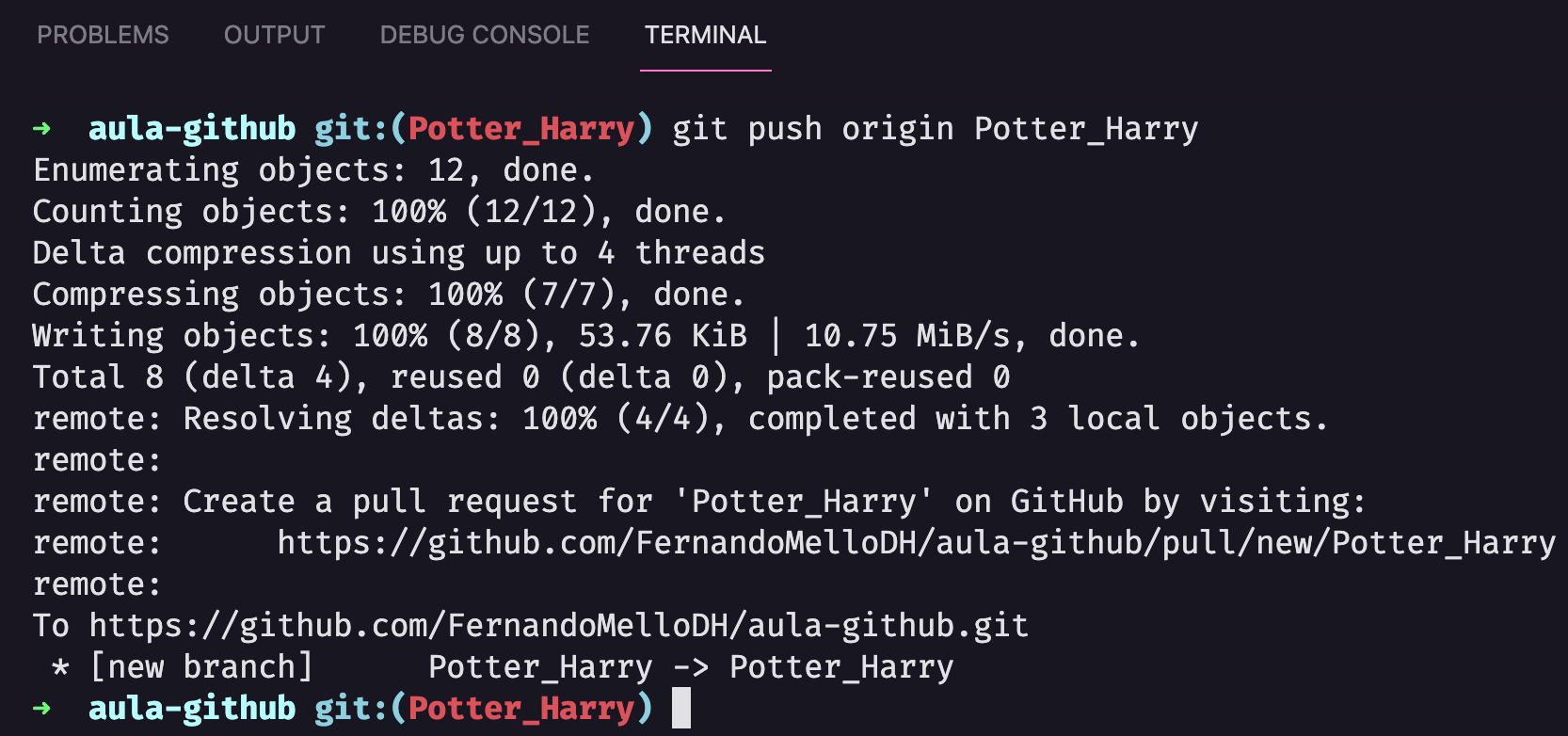
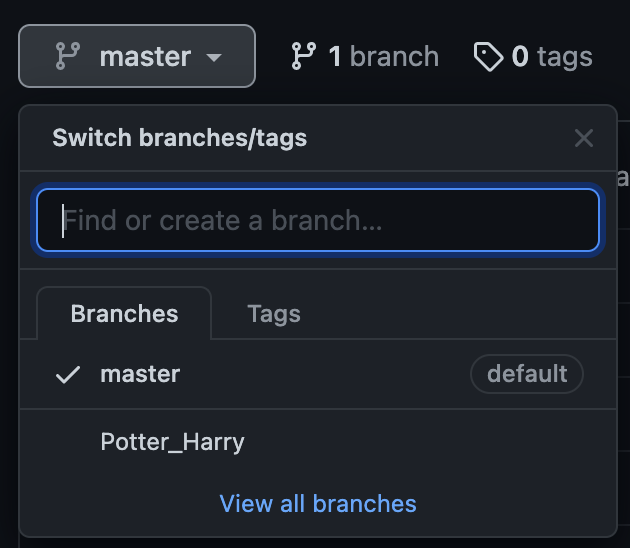
Para a Mochila vamos criar uma branch com nosso sobrenome e nome: Exemplo:  
git branch Potter\_Harry

Em seguida, mudamos para esse ramo usando o comando   
git checkout <nome-branch>

No nosso caso:  
git checkout Potter\_Harry

1. Podemos visualizar que estamos em nosso ramo porque em vez de (main) nosso sobrenome e nome aparecerão:  
   
2. Entramos na aula correspondente do dia. Se um determinado nome não for especificado, criaremos a pasta seguindo o exemplo a seguir:  
   **Aula\_X\_Assunto**  
   

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:

1. No código do Visual Studio, parecerá que temos um novo arquivo não rastreado:  
   
2. Se executarmos o comando git status, a nova pasta criada aparecerá pronta para adicionar, além de outras alterações:  
   
3. É hora de fazer o git add . e o commit  
   
4. Depois de fazer o commit, podemos enviar nosso branch com o comando:  
   git push origin <nombre-branch>  
   
5. 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:  
   

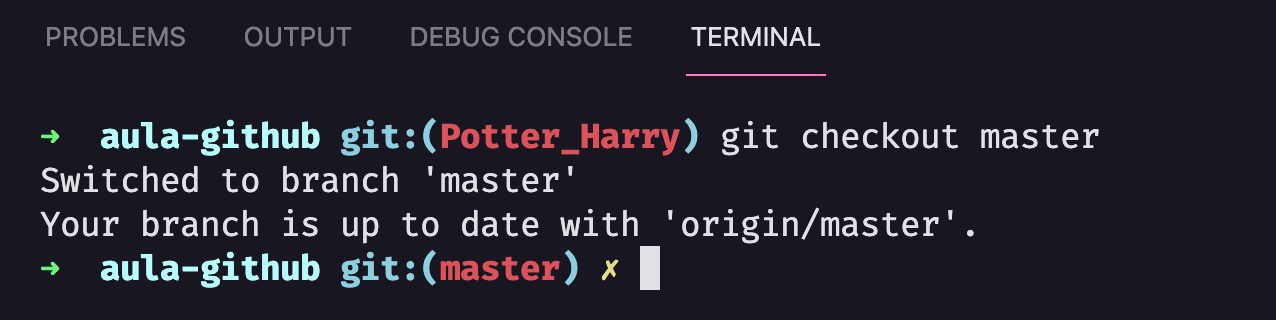
# Fazendo um merge local

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

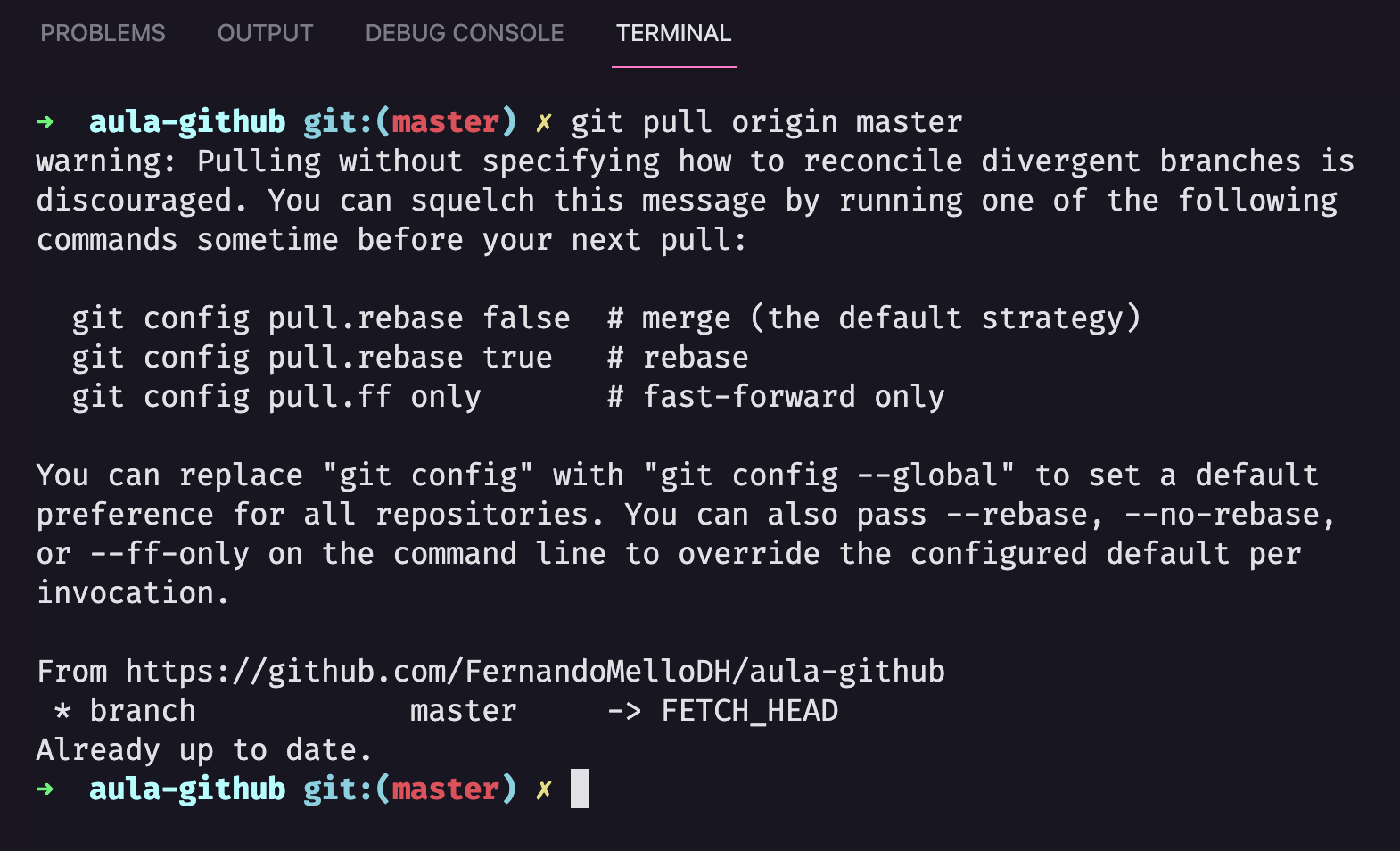
git checkout main

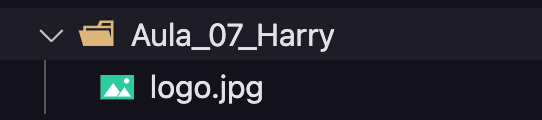
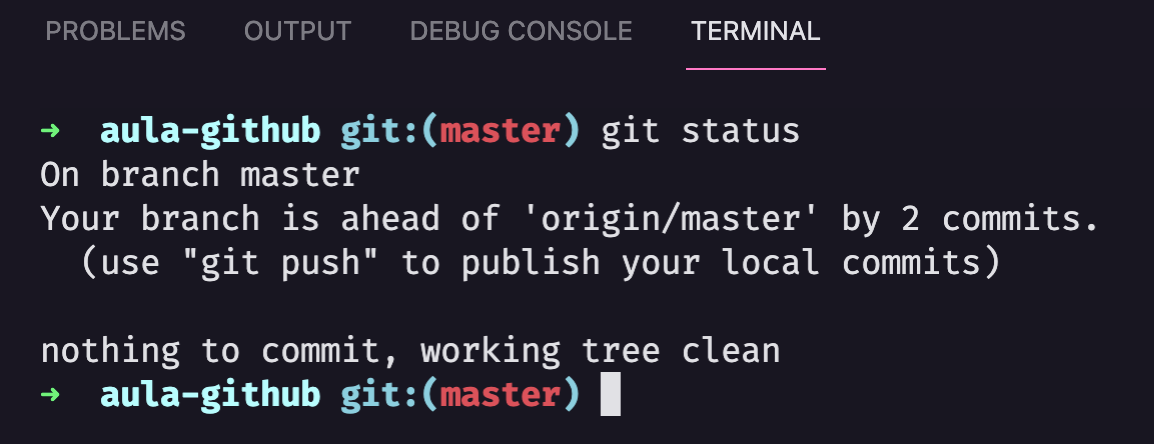
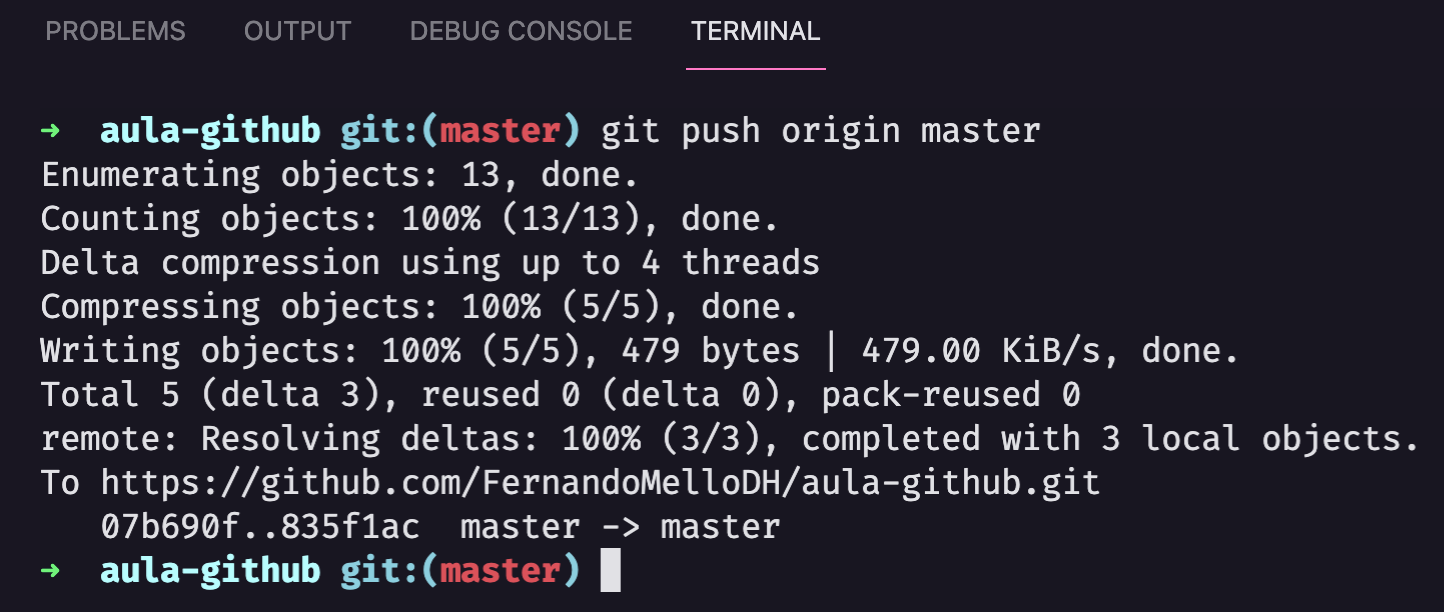
ou

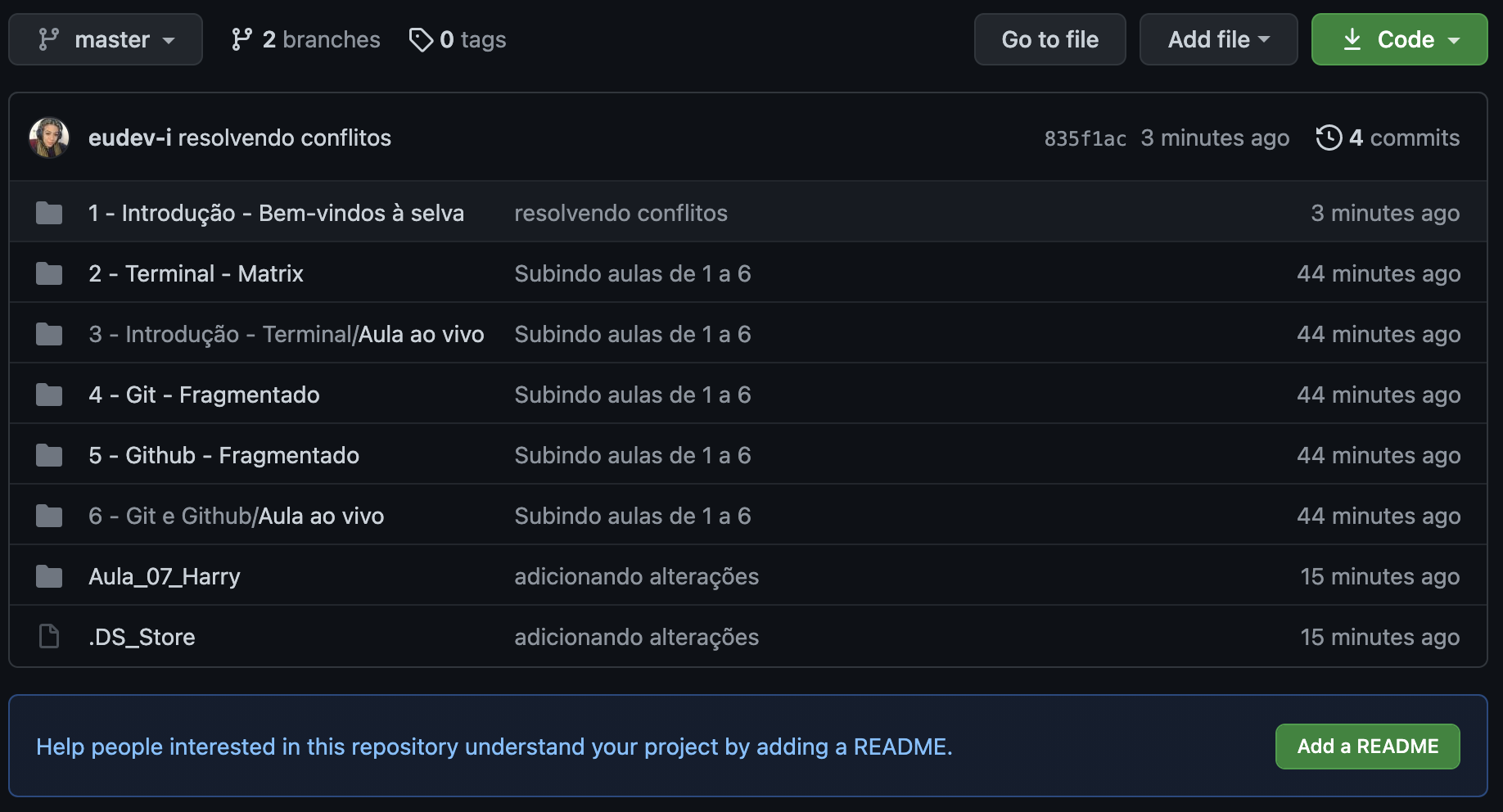
git checkout master



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

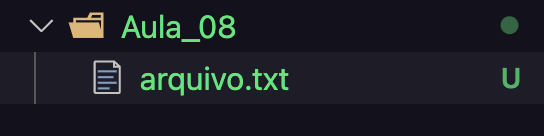
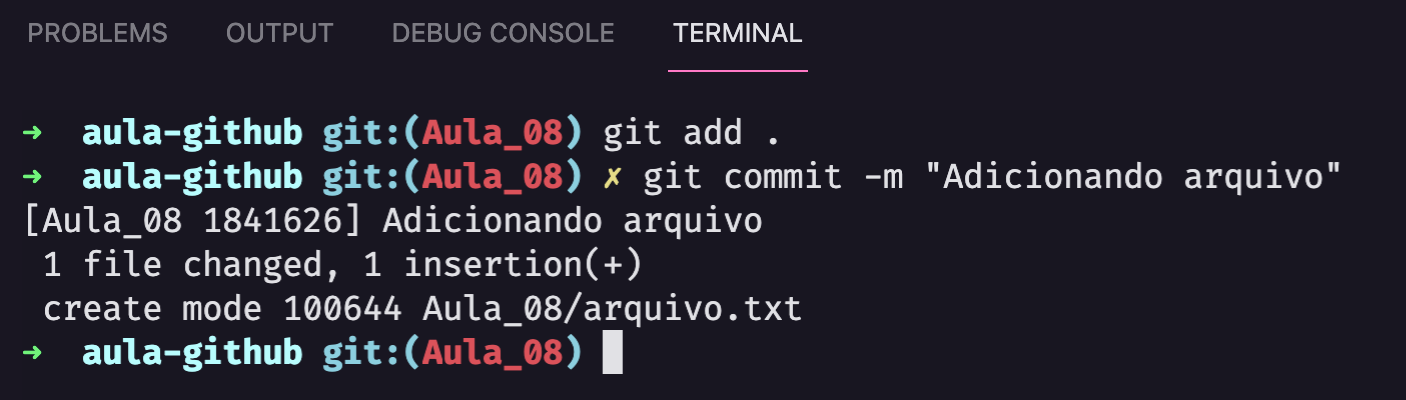
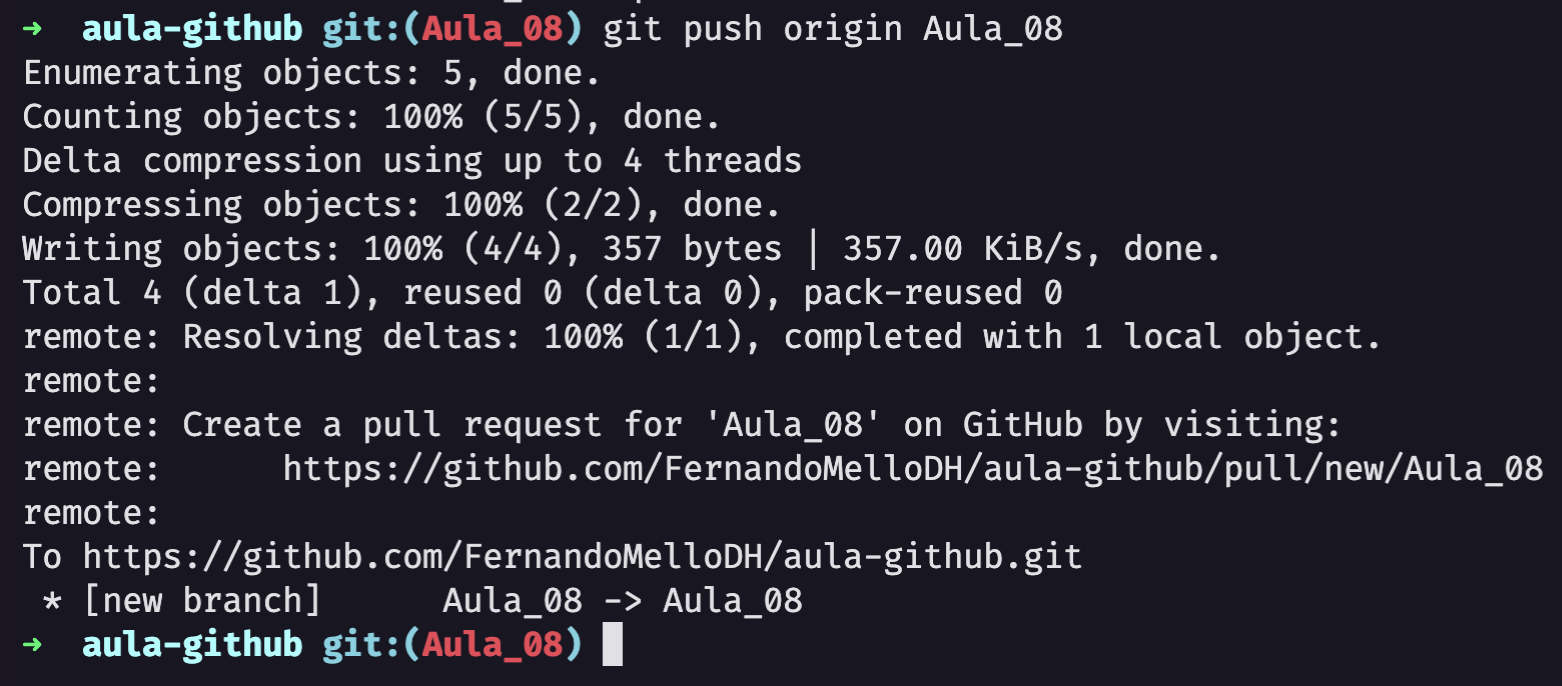
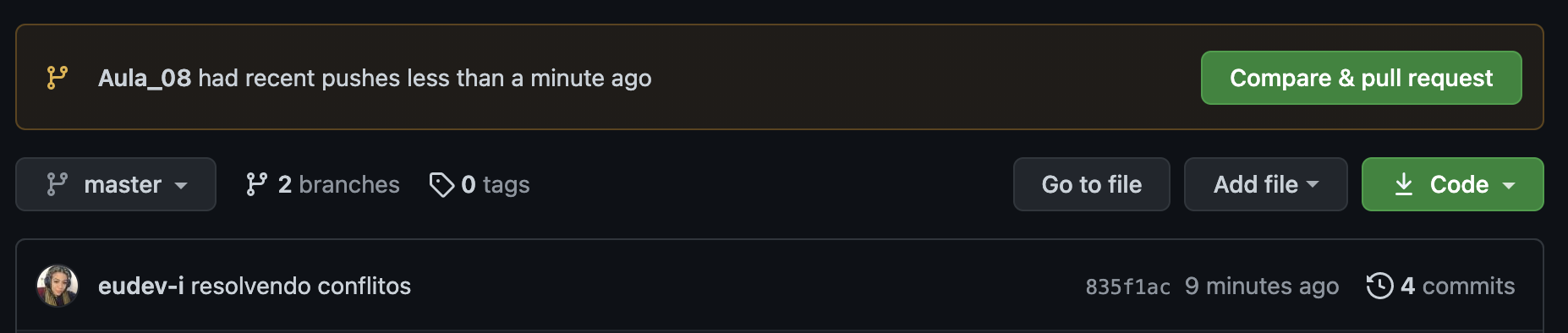
(Certamente muitas pastas serão carregadas, não se preocupe, é normal)  


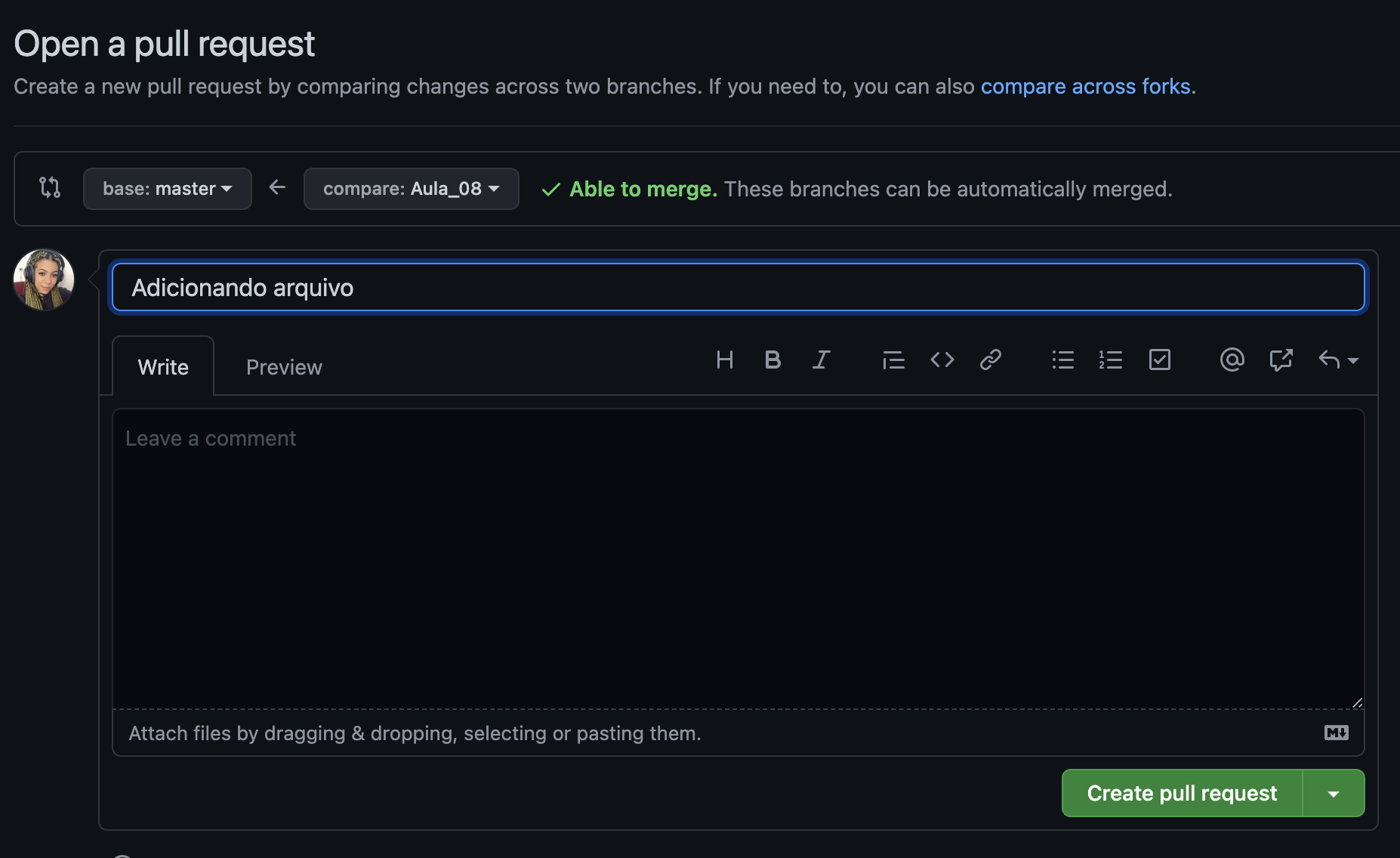
1. Este comando só me trará as mudanças no branch principal, se eu utilizar o comando git status:  
   
2. É hora do merge! Para isso usamos o comando:  
   git merge <nombre-branch>  
   
3. Agora minha pasta e seu conteúdo estarão na branch main:  
   
4. Se eu executar um git status após fazer o merge, ele só deve me avisar que a branch remota está desatualizada e eu preciso fazer o push do git para as alterações:  
   
5. Agora posso atualizar o repositório remoto com minhas alterações já na branch main ou master:  
   

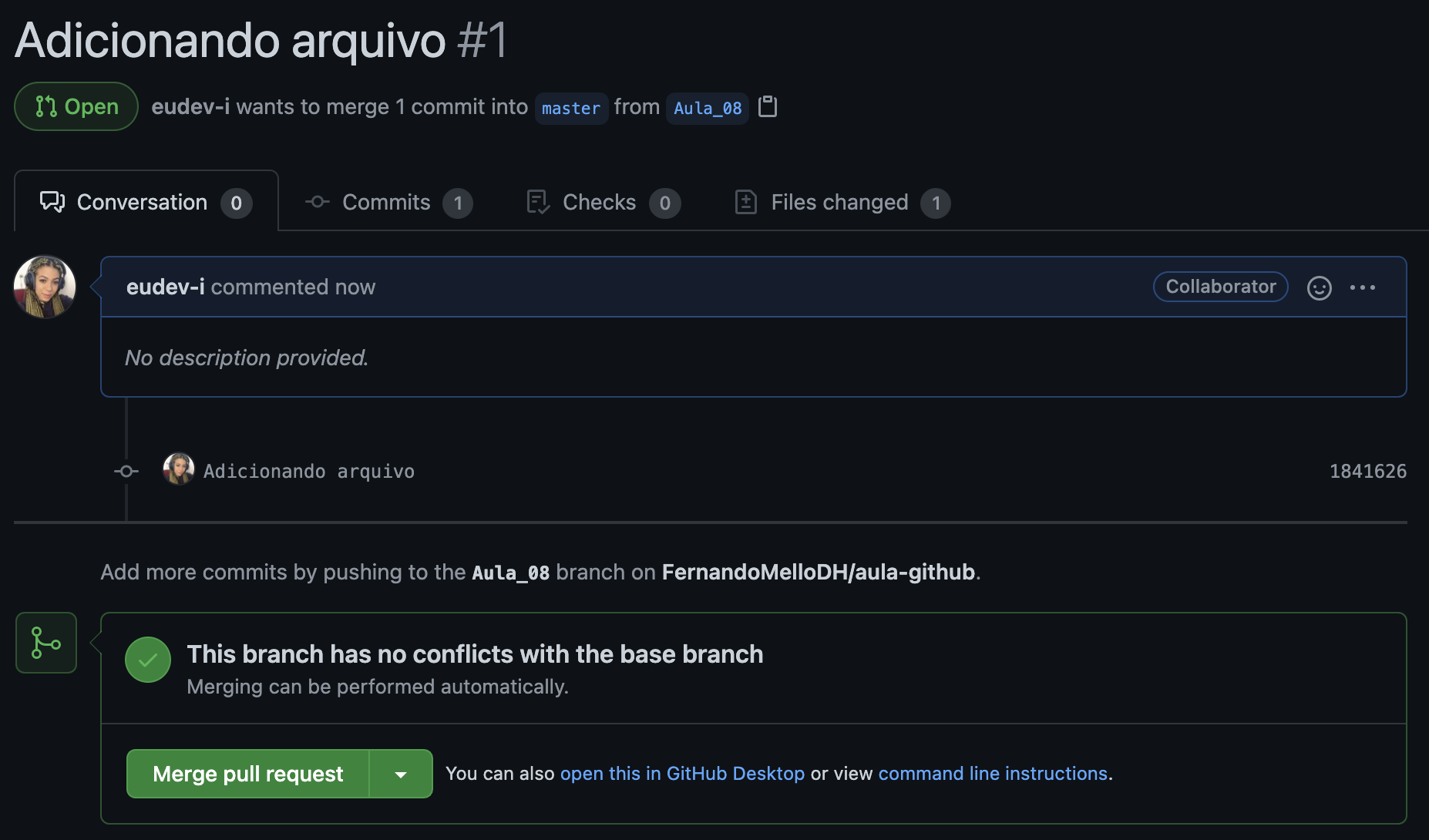
1. Posteriormente, podemos verificar o repositório remoto, minha pasta e seu conteúdo já estará na branch principal:

Combinando com repositório remoto

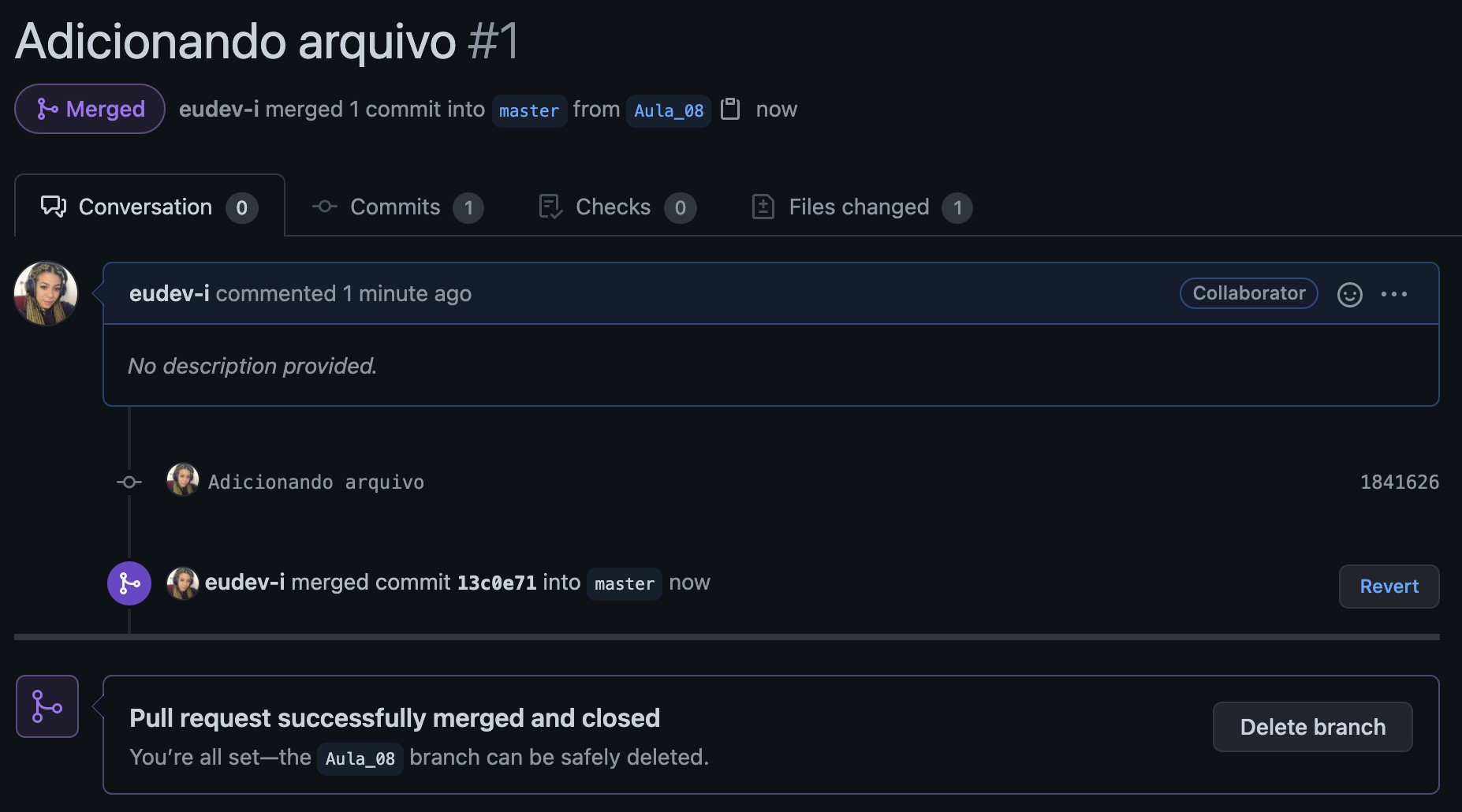
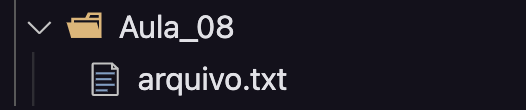
Existe outra opção de fazer uma fusão diretamente do Github, para isso devemos seguir os seguintes passos:

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


1. 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
2. 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.

1. Em seguida, pressiono o botão Merge pull request e o botão Merge pull request muda para “confirm merge”.
2. A janela que aparece deverá confirmar que está tudo bem e que unimos nossa branch.
3. 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  
   
4. Posteriormente, fazemos um git pull das mudanças utilizando git pull origin main.
5. O merge que fizemos no repositório remoto deve aparecer no ramo principal
6. Podemos verificar isso na estrutura visual do arquivo:  
   
7. 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!**