



GIT/GITHUB DE UMA FORMA MAIS TRYBER DE SER

Entendendo um pouquinho mais de GIT/GITHUB

NOME DO COMANDO	DESCRIÇÃO
git clone link-do-repositório	Clona o repositório
git config --global user.name git config --global user.email seuemail@exemplo.br	Configurar o nome de usuário e email
git push -u origin nome-da-branch	O -u é necessário somente na primeira vez que for fazer o push, isso serve para poder linkar a sua branch local com a branch do remoto
git push ou git push origin nome-da-branch	Empurra suas alterações para o repositório remoto
git pull	Traz as alterações do repositório remoto para o local
git status	Verifica se algo foi alterada ou adicionado, mostra o status do repositório local
git add . ou git add nome-do-arquivo	Adiciona o arquivo ou alterações
git commit -m "comentário à ser adicionado"	Faz um comentário sobre aquela adição ou alteração

NOME DO COMANDO	DESCRIÇÃO
git init	Inicia um repositório local, importante, apenas a pasta raiz deve ter o git init e não as subpastas.
git branch	Cria branch
git checkout nome-da-branch	Altera para branch
git checkout -b nome-da-branch	Cria branch e já altera para a própria
git log	Exibe histórico
git remote -v	Exibe seu repositório remoto
git remote set-url origin git@github.com:nomeuser/nome-repositório.git	Troca a url do seu repositório remoto

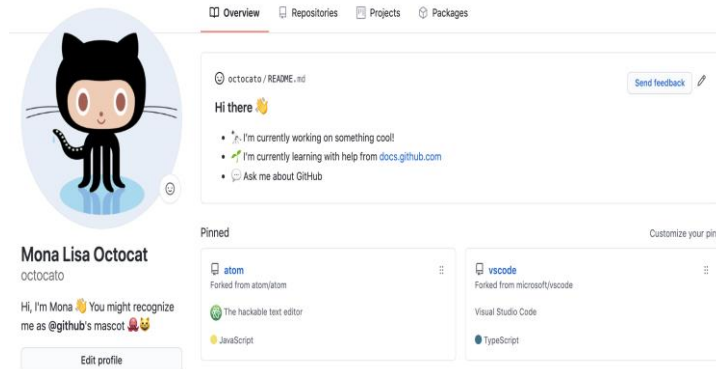


Para ficar mais nítido:



Local

Pastas e arquivos que estão no seu computador.



Remoto

Pastas e arquivos que estão no github.com
Por exemplo: [link](#)

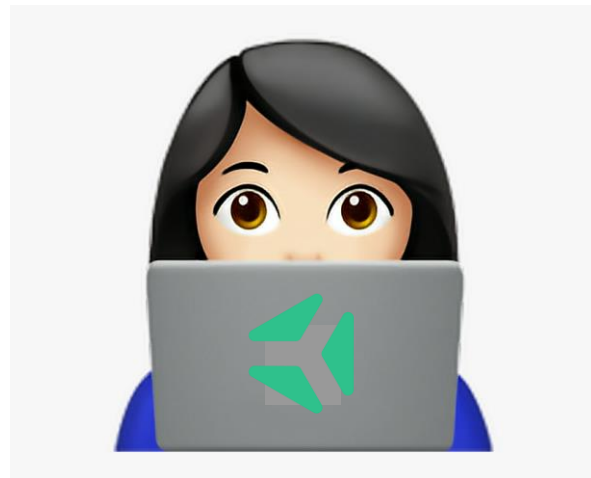




Imagine que...

Odinéia precisa criar um repositório para fazer os exercícios da semana de HTML da Trybe.

Esse é um passo a passo para ajudar Odinéia e **VOCÊ TRYBER** a entender um pouquinho mais de GIT e GITHUB.





Imagine que...

Odinéia vai entrar no repositório que ela já tem criado no git hub, conforme ela aprendeu no course, e vai clonar para sua nova máquina, já que ainda não o tinha...

The screenshot shows a GitHub repository interface. At the top, there are buttons for 'Go to file', 'Add file', and a green 'Code' button. Below these, a dropdown menu is open, displaying 'Clone' options: 'HTTPS', 'SSH' (which is selected and underlined), and 'GitHub CLI'. The SSH option shows the URL 'git@github.com:caabeatriz/trybe-exe' and a note to 'Use a password-protected SSH key.' Below the clone options is a 'Download ZIP' button. In the background, the repository details are visible, including the owner 'caabeatriz', a pull request, and a file tree with 'exercises/2.1' and 'README.md'.

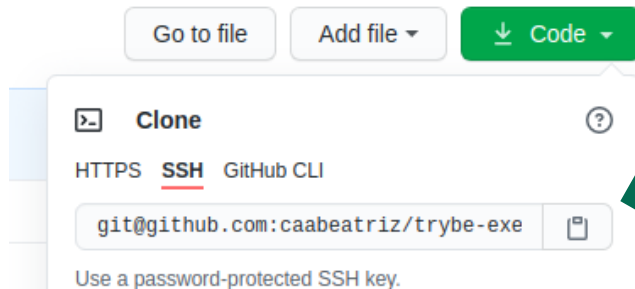




Passo a passo

1. Clicar em SSH
2. Copiar o link do SSH
3. Ir ao terminal e digitar **git clone link-do-repositório**

Importante: Não dê um git clone em uma pasta que já tenha um repositório, isso pode gerar problemas depois...



```
→ odinea git clone https://github.com/caabeatriz/trybe-exercises.git
Cloning into 'trybe-exercises'...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 20 (delta 3), reused 16 (delta 0), pack-reused 0
Unpacking objects: 100% (20/20), 1.98 KiB | 224.00 KiB/s, done.
→ odinea ls
trybe-exercises
```





Dica

Para saber se a pasta onde você está clonando já tem um repositório:

No seu **terminal**, vá até a pasta onde você quer verificar e digite **ls -la**, esse comando mostra as pastas ocultas, se aparecer **.git**, essa pasta já é um repositório e **você não deve** clonar outro repositório



Com **.git** não deve clonar outro repositório

```
exercises/271/trybe-skills-test
t:(main) ls -la
ls -la
-rw-r--r-- carol-silva 4096 mar  8 12:08 .
-rw-r--r-- carol-silva 4096 mar  8 12:07 ..
-rw-r--r-- carol-silva 4096 mar  8 12:08 exercises
-rw-r--r-- carol-silva 4096 mar  8 12:08 .git
-rw-r--r-- carol-silva  17 mar  8 12:07 README.md
```



Sem **.git**, pode clonar outro repositório

```
ls -la
-rw-r--r-- carol-silva 4096 mar  8 12:07 .
-rw-r--r-- carol-silva 4096 mar  8 13:41 ..
-rw-r--r-- carol-silva 4096 mar  8 12:08 trybe-exercises
```





Fluxo de adição de arquivo

1. 1. GIT CLONE

Serve para “salvar” o repositório na sua máquina. **Obs: Uma vez salvo, não é preciso salvar toda vez que for realizar alterações e adições.**

3. EDIÇÃO

Odinéia começa a editar e adicionar arquivos no seu projeto, a cada edição, ela adiciona um comentário com os seguintes comandos:

5. ADICIONAR COMENTÁRIO

É necessário, depois que adicionar o arquivo, realizar um comentário com o comando **git commit -m “Algum comentário específico”**

2. CRIAR BRANCH

A partir da branch principal (main ou master), crie uma branch e já altere para ela com o comando **git checkout -b nome-da-nova-branch**

4. ADICIONAR EDIÇÃO

Com o **git add nome-do-arquivo**, é possível adicionar os arquivos em staging.

6. ENVIO DAS ALTERAÇÕES

Antes de enviar, vamos verificar se tudo está certo, com o comando **git status** podemos verificar se ainda existe algo a ser alterado. Se ao digitar **git status** aparecer a mensagem de **not commit**, pode continuar com o fluxo, se aparecer outra mensagem, volte ao passo 4. Qualquer dúvida, procure as pessoas instrutoras da sua turma.



Fluxo GITHUB

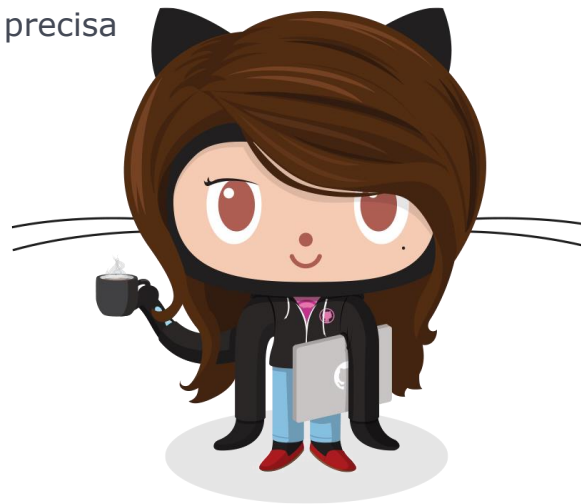


Como é a **primeira vez** que Odíneia esta fazendo seu **git push** com outra branch, é necessário utilizar **git push -u origin nome-da-branch**. Assim, ela envia sua branch local para o remoto

Observação: Se **não** fosse a primeira vez e ela usou o prefixo **-u**, era só colocar **git push**, do contrário seria necessário **git push origin nome-da-branch**

Mas ainda não acaba por aí! Como Odinéia criou uma branch, agora ela precisa juntar as modificações que estão nessa branch com a branch principal (main ou master).

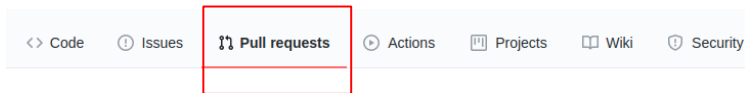
E agora começamos o fluxo do PULL REQUEST.



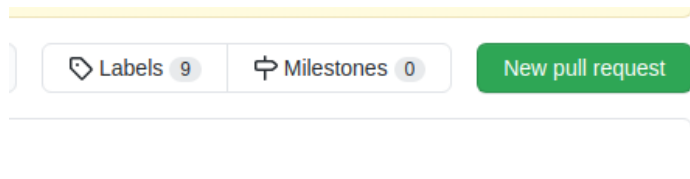


Fluxo pull request

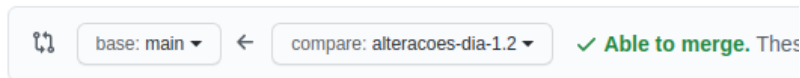
1. No seu repositório no git hub, clique em **Pull requests**



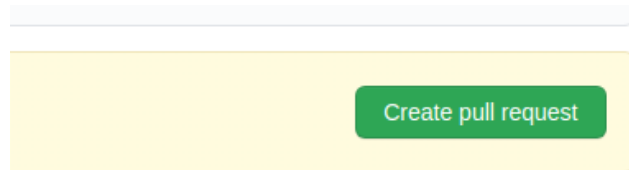
2. No lado direito, clique em **New pull request**



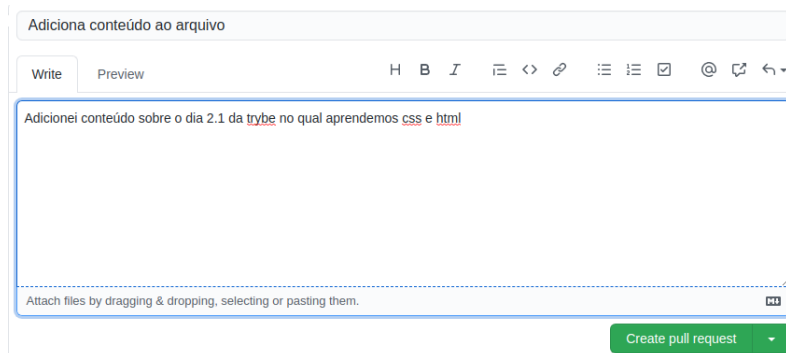
3. Clique em **compare** e escolha sua branch



4. Ao lado direito, clique em **Create Pull request**



5. Escreva alguma descrição sobre essa nova alteração e clique em **Create pull request**



Continua



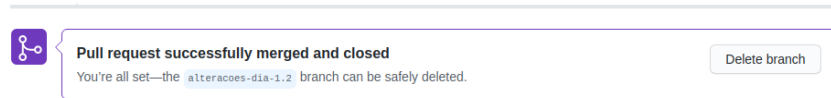


Fluxo pull request

6. Como não temos conflitos, clique em **Merge pull request** e depois **Confirm merge**



7. Após mergeado, caso você não vá utilizar mais essa branch, clique em **Delete branch**



Prontinho agora suas alterações já estão na branch principal :D





Fluxo de github

De local para remoto



Imagine que...

Gertrudes já tem uma pasta na qual está fazendo alguns exercícios da aula e agora quer criar um repositório, pois quer mostrar para as amigas suas resoluções...

Vamos acompanhar Gertrudes e o que ela deve fazer ?





Passos necessários

Gertrudes já tem uma pasta onde salvou seus exercícios

```
→ trybe-exercises ls  
Bloco-1
```

Agora ela precisa fazer essa pasta virar um repositório local, ela consegue criar ao digitar **git init** na pasta principal, observe:

```
→ trybe-exercises git init  
Initialized empty Git repository in  
ta/trybe-exercises/.git/  
→ trybe-exercises git:(master)
```





Passos necessários

Para verificar se um repositório Git foi de fato iniciado, você pode executar o comando **git status**

```
→ trybe-exercises git:(master) git status
No ramo master

No commits yet

nada para enviar (crie/copie arquivos e use "git
```



Como ela já tem alguns arquivos, é preciso fazer o fluxo de adicionar, comentar e enviar para um repositório remoto.





Passos necessários para adição de arquivos

`git checkout -b nome-da-nova-branch`

`git add nome-do-arquivo-a-ser-adicionado`

`git commit -m "comentario"`

Atenção

Antes de dar **git push**, é necessário linkar seu repositório local com o repositório no github.

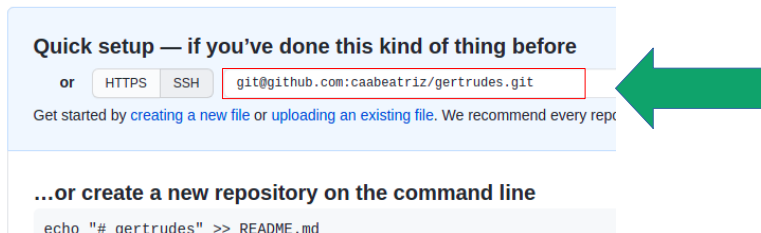
É necessário criar um repositório no github, caso não saiba como fazer, [clique aqui](#) e siga até o **passo 6**.





Passos necessários para adição de repositório

Agora criado o repositório remoto, copie esse link



e volte ao seu terminal e digite **git remote add nomeRepositório linkRepositório**

Depois digite **git remote -v**

Veja se tudo deu certo, o retorno vai ser algo parecido com:

```
cs@cs-Vostro-5490:~/Área de Trabalho/Trybe/gertrudes$ git remote -v
origin  git@github.com:caabeatriz/gertrudes.git (fetch)
origin  git@github.com:caabeatriz/gertrudes.git (push)
```

nomeRepositório - o ideal é colocar **origin**, mas pode ser um de sua escolha, esse nome é o que será referência na sua máquina na hora de dar push.

linkRepositório - é o link que você copiou após a criação do repositório.





Passos necessários para enviar os arquivos de local para remoto

Agora, vamos enviar os arquivos local para o novo repositório remoto

git push -u nomeRepositório nome-da-branch

nomeRepositório - o nome que você colocou anteriormente, em nosso exemplo é **origin**.

Nome-da-branch - o nome da sua branch.

Prontinhoooo!!! Os arquivos locais de Gertrudes agora estão no remoto para mostrar para as amigas.





Sua conquista é a
nossa conquista.



betrybe.com

