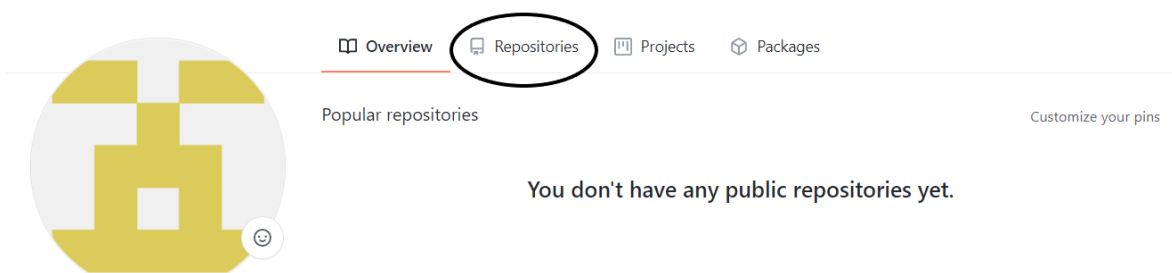


Primeiros passos para clonar um repositório - Mapa Mental

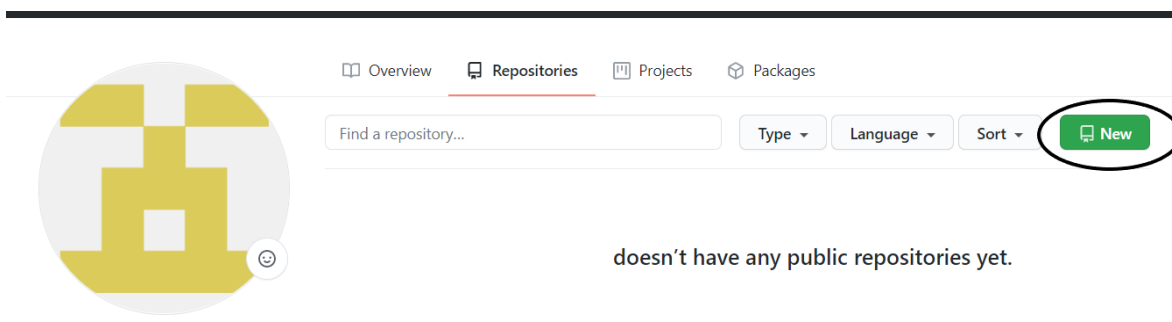


Criando um repositório

- No github lá, na sua conta, clique em “Repositories”.





- Clique no botão “New”




- Coloque o nome do repositório: **Projetos** e pode deixar como **Private**.


Owner * Repository name *

 Aqui aparece seu usuário / Projetos 


Great repository names are short and memorable. Need inspiration? How about [friendly-umbrella?](#)

Description (optional)

☐  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☒  **Private**
You choose who can see and commit to this repository.

- Abaixo tem o botão Create Repository, clique nele.

☒  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Criando dois arquivos no master

- Iremos criar dois arquivos no master, mas primeiro certifique-se que está em:
~/Projetos/MeuPrimeiroGit (master)

Com o editor nano criaremos os dois arquivos abaixo:

nano index.html

nano style.css

- Vamos visualizar os arquivos

git status

```
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html
        style.css
```

- Vamos rastrear os arquivos no git:

git add .

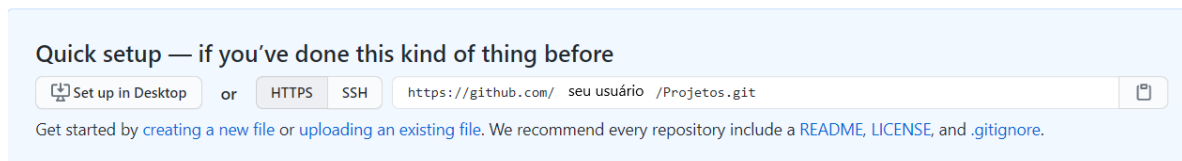
- Na sequência criar um ponto de salvamento:

```
git commit -m "Criando os arquivos index.html e style.css"
```

Enviando arquivos para o repositório Projetos

Para enviar os arquivos remotamente, precisa acessar o github remotamente.

- Copie o código do repositório



Quando você cria o repositório, são dados os passos para chegar ao push.

```
echo "# Projeto1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/IvanaBernardo/Projeto1.git
git push -u origin main
remote add origin https://github.com/seuusuario/Projetos.git
```

No nosso caso só iremos executar os **passos em negritos**, pois já iniciamos e comitamos.

```
git branch -M main
```

Troca de master para main (devido o token)

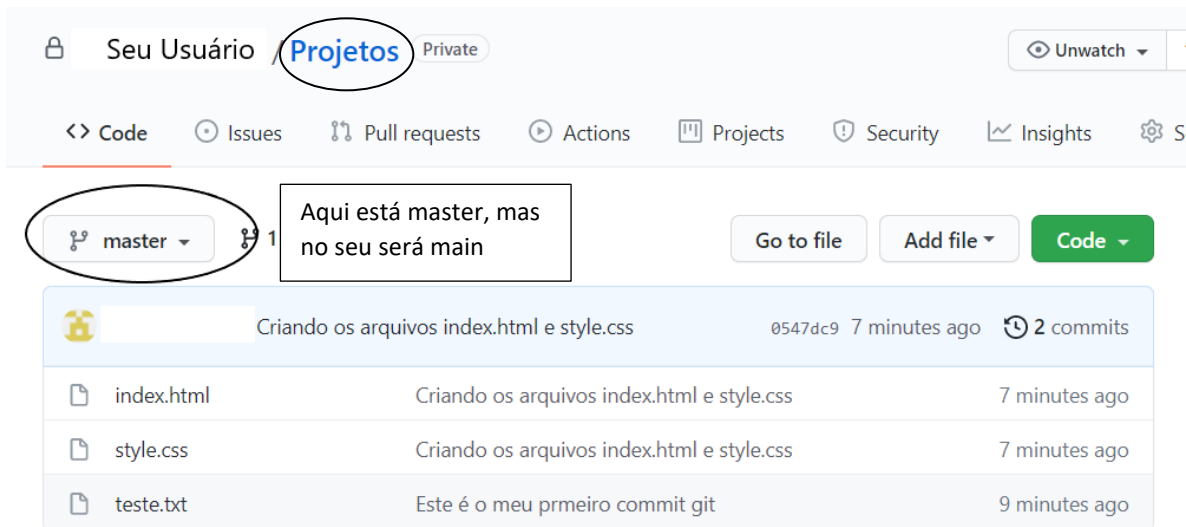
```
git remote add origin https://github.com/IvanaBernardo/Projeto1.git
```

Acessar remotamente o repositório do github.

```
git push -u origin main
```

Envia todos os commits do branch local para o repositório remoto identificado pelo nome origin em seu branch remoto nomeado main.

Se clicar no link “Projetos” verá o branch “main” com os 3 arquivos.



Obs.: Quando estiver no master, use git push para as próximas vezes.

Criando branches

A branch basicamente são versões diferentes do sistema, a branch principal se chama master.

Para trabalhar em linha de produção cria-se outro branch que servirão de ambiente de teste ou linha de produção.

- Podem existir dois branches:

1º Trabalhando na versão atual do sistema (master)

2º Trabalhando na versão 2.0 do sistema (produção)

- Iremos criar um novo branch

git branch producao (será uma cópia do master)

- Com o comando: git branch, visualiza todas as branches.

git branch

* master (mostra qual branch está trabalhando)
producao

- Agora mudaremos de branch master para branch producao.

git checkout producao

Switched to branch 'producao'
~/Projetos/MeuPrimeiroGit (producao)

- Pode executar o comando git branch para visualizar a alteração.

git branch

```
master
* producao
```

- Vamos enviar todos os commit no branch local para o repositório remoto identificado pelo nome origin em seu branch remoto nomeado producao:

```
git push -u origin producao
```

```
To https://github.com/IvanaBernardo/Projetos.git
* [new branch]      producao -> producao
Branch 'producao' set up to track remote branch 'producao' from 'origin'.
```

- Com o comando “ls” você visualiza os arquivos
ls

```
index.html  style.css  teste.txt
```

- Agora vamos visualizá-los no github

Mude o DropDown do branch para producao, e veja que ele fez uma cópia dos arquivos que estão na branch master. Note também que mostram 2 branches.
Agora a equipe de desenvolvimento pode trabalhar com uma versão do sistema ou site enquanto a versão atual está em uso.

The screenshot shows the GitHub interface for a repository named 'Projetos'. The branch 'producao' is selected and circled in red. Below the branch selector, it says '2 branches' and '0 tags'. A message states 'This branch is even with master.' Below this, a commit history table is visible:

Commit Hash	Commit Message	Time Ago
0547dc9	Criando os arquivos index.html e style.css	21 minutes ago
	index.html	Criando os arquivos index.html e style.css
	style.css	Criando os arquivos index.html e style.css
	teste.txt	Este é o meu primeiro commit git

Desfazendo commit

- Estando no **branch producao**, iremos alterar o index.html, rastrear e cria um ponto de commit.

```
nano index.html (altere o arquivo)
git add index.html ou git add .
git commit -m "Alterando o arquivo index.html"
```

```
[producao e70c79a] Alterando o arquivo index.html
1 file changed, 1 insertion(+)
```

Observe que trouxe os 7 primeiros números do registro de log do commit, eles são suficientes para desfazer o commit, mas nada impede de usar todos os números do registro de log.

Vamos supor que algo deu errado na alteração do index.html ou o analista falou que não era necessário realizar essa alteração, você precisará voltar o arquivo de origem.

Primeiramente vamos visualizar o log com o git dos commits.

git log

```
commit e70c79a50919e2a45949ba76f192be584d83dea7 (HEAD -> producao,
origin/producao)
Author: Seuusuario <seu email>
Date:   Wed Aug 18 22:04:33 2021 -0400

    Alterando o arquivo index.html

commit 0547dc910c4470976ea6debd5b5cfc492128b85 (origin/master, master,
checkout)
Author: Seuusuario <seu email>
Date:   Wed Aug 18 21:13:42 2021 -0400

    Criando os arquivos index.html e style.css

commit 5e5852e1c41d066e19665e6e636e3779aafaf35c
Author: Seuusuario <seu email>
Date:   Wed Aug 18 21:11:32 2021 -0400

    Este é o meu primeiro commit git
```

Para voltar tem-se 2 comandos: **soft** e **reset**.

- O **soft** desfaz o commit sem apagar o que foi feito, podem ser alterados e novamente comitado, ideal para um ambiente de produção.

```
git reset --soft 0547dc9
```

ou pode ser usado o commit completo

```
git reset --soft 0547dc910c4470976ea6debd5b5cfc492128b85
```

- O **hard** apaga tudo o que foi feito.

```
git reset --hard 0547dc9
```

ou pode ser usado o commit completo

```
git reset --hard 0547dc910c4470976ea6debd5b5cfc492128b85
```

- **Iremos usar o hard**

```
git reset --hard 0547dc910c4470976ea6debd5b5cfc492128b85
```

Vamos ver o log, e observe que agora existem apenas 2 commit.

git log

```
commit 0547dc910c4470976ea6debd5b5cfcb492128b85 (origin/master, master,
checkout)
```

```
Author: Seuusuario <seu email>
```

```
Date:   Wed Aug 18 21:13:42 2021 -0400
```

```
    Criando os arquivos index.html e style.css
```

```
commit 5e5852e1c41d066e19665e6e636e3779aafaf35c
```

```
Author: Seuusuario <seu email>
```

```
Date:   Wed Aug 18 21:11:32 2021 -0400
```

```
    Este é o meu primeiro commit git
```

Se abrir o index.html notará que ele voltou para o que estava antes do comando hard.

Aqui em um tutorial como excluir um repositório no github

<https://docs.github.com/pt/github/administering-a-repository/managing-repository-settings/deleting-a-repository>

Você pode excluir o repositório do github e a pasta Projetos do seu computador para repetir as etapas novamente. Lembre-se de começar pelo mapa mental.

Clonando um repositório

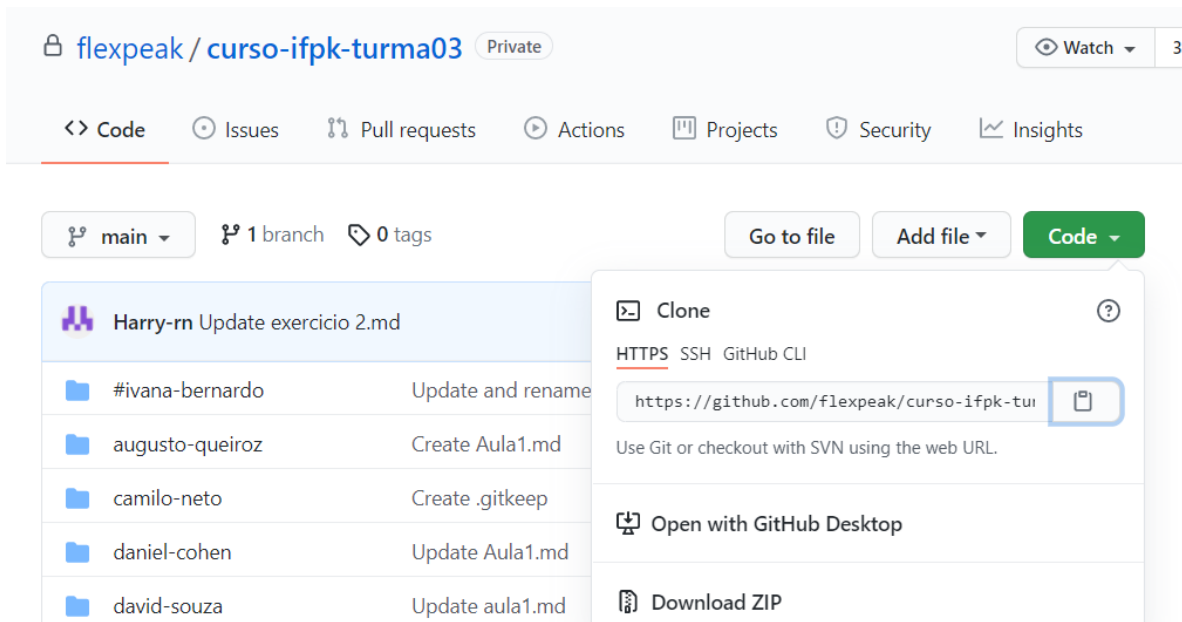
Para finalizar iremos aprender a clonar do repositório para a máquina local, mas iremos fazer na pasta Projetos que foi criado na máquina. Execute o comando “cd ..” para sair da subpasta.

```
cd ..
```

```
~/Projetos
```

Para exemplificar iremos clonar o repositório do curso.

Entre no repositório e copie o código.



O comando “git clone” é usado para copiar um repositório existente para um diretório local

```
git clone https://github.com/flexpeak/curso-ifpk-turma03.git
```

```
Cloning into 'curso-ifpk-turma03'...
remote: Enumerating objects: 1219, done.
remote: Counting objects: 100% (714/714), done.
remote: Compressing objects: 100% (538/538), done.
remote: Total 1219 (delta 516), reused 168 (delta 168), pack-reused 505
Receiving objects: 100% (1219/1219), 9.15 MiB | 661.00 KiB/s, done.
Resolving deltas: 100% (664/664), done.
```

Prontinho, se listar a pasta Projetos, verá a pasta “curso-ifpk-turma03” e dentro dela todos as pastas e arquivos que estão no github.

```
cd curso-ifpk-turma03
```

```
~/Projetos/curso-ifpk-turma03 (main)
```

```
ls
```

```
'#ivana-bernardo'/  fabio-cantuario/  marla-guedes/  rebecca-souza/
augusto-queiroz/  flavio-silva/    matheus-lima/  robson-lima/
camilo-neto/      francisco-junior/ mauricio-mathias/ thiago-farias/
daniel-cohen/     isabela-monteiro/ peterson-albuquerque/ thiago-santos/
david-souza/      joao-rocha/      rafael-oliveira/ wolfgang-aly/
elton-oliveira/   lorrان-farias/    raimundo-neto/
emilly-araujo/    lucas-lobes/     raissa-lima/
erike-martins/    marcos-costa/    readme.md
```