

Versionamento com Git e GitHub

Miguel Miranda Morandini

Versionamento no GitHub

O objetivo deste trabalho é aprender e praticar recursos básicos de versionamento de software por meio do GitHub, compreendendo como funciona a linha de versionamento (histórico de commits) e adquirindo fluência na configuração e primeiros passos com Git e GitHub, na criação de commits, bem como na execução de push, a criação de Pull Requests, resolução de conflitos, criação e gerenciamento de forks, criação de branch e realização de merge.

Comando Básicos:

- **git init**: inicia o git na pasta especificada no endereço do terminal.
- **touch nome-do-arquivo**: cria um novo arquivo. No exemplo, da imagem 1, criou-se o arquivo 'index.html'.
- **git add .**: envia todos os arquivos da pasta para o Git, saindo da fase de Modified e indo para a fase de Staging (ainda não está no repositório remoto/github)
- **git status**: mostra em que estagio estão os arquivos da pasta.
- **git commit -m "exemplo-de-mensagem"**: envia um comentário sobre a nova adição.
- **git push**: envia todas as alterações para o GitHub.

```
migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1
$ git init
Initialized empty Git repository in C:/Users/migue/Documents/curso-git-e-github/code/projeto1/.git/

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ touch index.html

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git add .

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git commit -m "Adicionado arquivo index"
[master (root-commit) 7133808] Adicionado arquivo index
1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Imagem 1. Criando e fazendo o commit do arquivo 'index.html'

```

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ touch estilo.css

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        estilo.css

nothing added to commit but untracked files present (use "git add" to track)

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
    add

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git add .

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git commit -m "Adicionado arquivo de estilo css"
[master 688f972] Adicionado arquivo de estilo css
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 estilo.css

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git status
On branch master
nothing to commit, working tree clean

```

Imagem 1. Criando e fazendo o commit do arquivo 'estilo.css'

Branch é uma ramificação do projeto, que continua a partir de um ponto específico da linha do tempo. Depois é possível adicionar as novas features criadas nela a branch principal, main, através de um merge.

- **git branch:** exibe todas as branches existentes.
- **git branch nome-da-nova-branch:** cria uma branch nova.

```

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git branch
* master

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git branch teste

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git branch
* master
  teste

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git checkout teste
Switched to branch 'teste'

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (teste)
$ sit status
bash: sit: command not found

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (teste)
$ git status
On branch teste
nothing to commit, working tree clean

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (teste)
$ git branch
  master
* teste

```

Imagem 2. Criando a branch 'teste'

- **git checkout nome-da-branch-a-entrar**: sai da branch atual e entra em outra.
- **git checkout -b nome-da-nova-branch**: sai da branch atual, cria uma nova e já entra nela.
- **git merge nome-da-branch-que-será-a-principal**: une os conteúdos da branch atual à branch especificada no comando. No exemplo a seguir, existe a branch master e a branch teste. A branch teste possui o arquivo 'mercado-pago.js' que não está na branch master. Então, através do comando merge, as duas branches serão fundidas, fazendo com que o arquivo 'mercado-pago.js' faça parte da branch master.

```
migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (teste)
$ git checkout master
Switched to branch 'master'

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git status
On branch master
nothing to commit, working tree clean

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git merge teste
Updating e541198..49a0dcd
Fast-forward
 mercado-pago.js | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 mercado-pago.js
```

Imagem 3. Comando merge para juntar features de teste à branch master

- **git log --oneline**: exibe todos os commits realizados e seu id.

```
migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git log --oneline
49a0dcd (HEAD -> master, teste) Adicionado arquivo js mercado pago
e541198 Adicionado codigo html
688f972 Adicionado arquivo de estilo css
7133808 Adicionado arquivo index
```

Imagem 4. Comando git log --oneline

É comum acontecer conflitos em repositórios quem varias pessoas trabalham. Como no exemplo abaixo, onde fui alterado o arquivo 'estilo.css' na branch teste tornando a cor do background green. Porém, na branch master fizeram a alteração para blue. Quando a tentativa de merge ocorre, a mensagem de conflito é disparada:

```

migue@DESKTOP-F68GVQT MDNGW54 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git checkout teste
Switched to branch 'teste'

migue@DESKTOP-F68GVQT MDNGW54 ~/Documents/curso-git-e-github/code/projeto1 (teste)
$ git status
On branch teste
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   estilo.css

no changes added to commit (use "git add" and/or "git commit -a")

migue@DESKTOP-F68GVQT MDNGW54 ~/Documents/curso-git-e-github/code/projeto1 (teste)
$ git add .

migue@DESKTOP-F68GVQT MDNGW54 ~/Documents/curso-git-e-github/code/projeto1 (teste)
$ git commit -m "Adicionado cor de fundo no estilo css"
[tester 5635232] Adicionado cor de fundo no estilo css
1 file changed, 1 insertion(+), 1 deletion(-)

migue@DESKTOP-F68GVQT MDNGW54 ~/Documents/curso-git-e-github/code/projeto1 (teste)
$ git status
On branch teste
nothing to commit, working tree clean

migue@DESKTOP-F68GVQT MDNGW54 ~/Documents/curso-git-e-github/code/projeto1 (teste)
$ git checkout master
Switched to branch 'master'

migue@DESKTOP-F68GVQT MDNGW54 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   estilo.css

no changes added to commit (use "git add" and/or "git commit -a")

migue@DESKTOP-F68GVQT MDNGW54 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git add .

migue@DESKTOP-F68GVQT MDNGW54 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git commit -m "Fundo alterado de black para azul no estilo css"
[master 077f942] Fundo alterado de black para azul no estilo css
1 file changed, 2 insertions(+), 2 deletions(-)

migue@DESKTOP-F68GVQT MDNGW54 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git merge teste
Auto-merging estilo.css
CONFLICT (content): Merge conflict in estilo.css
Automatic merge failed; fix conflicts and then commit the result.

```

Imagem 5. Exemplo de conflito em merge



Imagem 6. Conflito no arquivo estilo.css

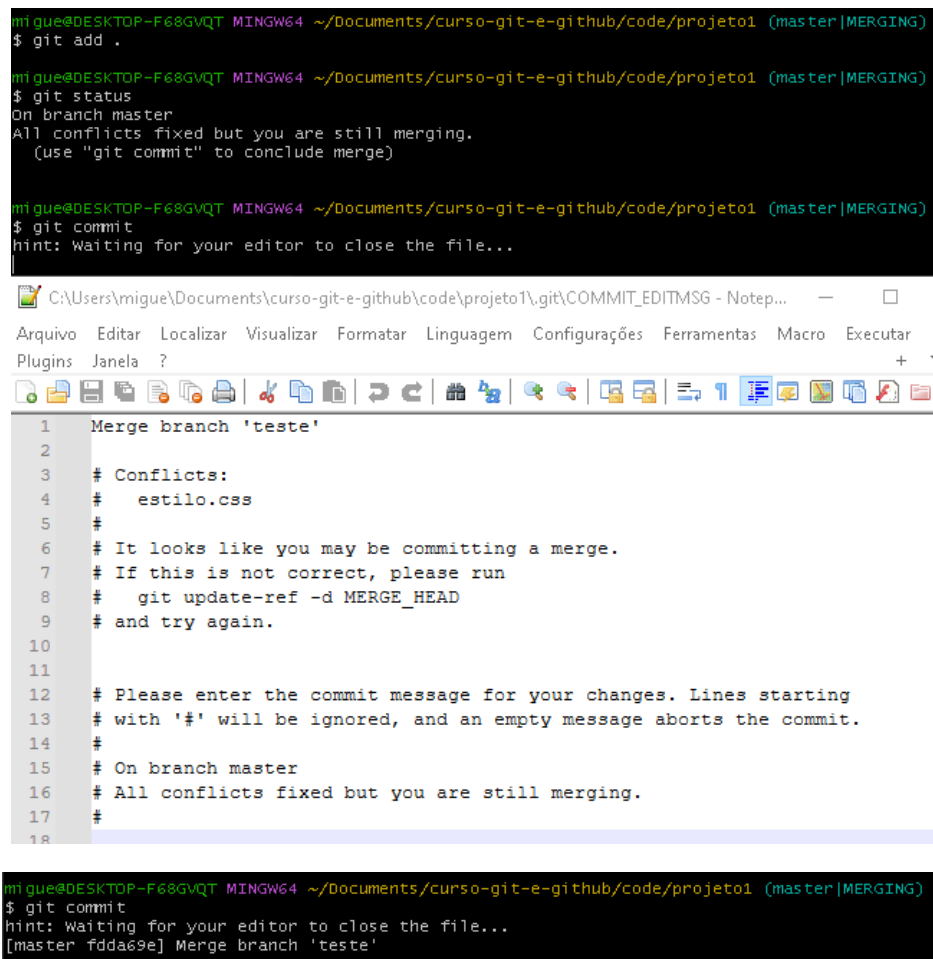
Para resolver o conflito, basta apagar o que não é desejado estar no arquivo. Como no exemplo, onde a cor final deve ser blue.



```
estilo.css ...\projeto1 ! X .env .gitignore A tes ...
code > projeto1 > estilo.css > body
1 body {
2     background: blue;
3 }
4
```

Imagem 7. Resolução de conflito em estilo.css

Por fim, adiciona o arquivo alterado no Git, através do **git add ..** E se faz o commit, sem a necessidade de uma mensagem, pois já é gerado uma mensagem de merge padrão.



```
migue@DESKTOP-F68GVQT MINGW64 ~/Documents/cursos-git-e-github/code/projeto1 (master|MERGING)
$ git add .

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/cursos-git-e-github/code/projeto1 (master|MERGING)
$ git status
On branch master
All conflicts fixed but you are still merging.
(use "git commit" to conclude merge)

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/cursos-git-e-github/code/projeto1 (master|MERGING)
$ git commit
hint: Waiting for your editor to close the file...

C:\Users\migue\Documents\cursos-git-e-github\code\projeto1\git\COMMIT_EDITMSG - Notepad...
Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro Executar
Plugins Janela ?
1 Merge branch 'teste'
2
3 # Conflicts:
4 #   estilo.css
5 #
6 # It looks like you may be committing a merge.
7 # If this is not correct, please run
8 #   git update-ref -d MERGE_HEAD
9 # and try again.
10
11
12 # Please enter the commit message for your changes. Lines starting
13 # with '#' will be ignored, and an empty message aborts the commit.
14 #
15 # On branch master
16 # All conflicts fixed but you are still merging.
17 #
18

migue@DESKTOP-F68GVQT MINGW64 ~/Documents/cursos-git-e-github/code/projeto1 (master|MERGING)
$ git commit
hint: Waiting for your editor to close the file...
[master fdda69e] Merge branch 'teste'
```

Imagens 8 e 10. Merge realizado após resolução de conflito em estilo.css

Para o repositório ficar salvo remotamente é necessário efetuar o comando `git push link-do-repositorio-github nome-da-branch`:

```
migue@DESKTOP-F68GVQT MINGW64 ~/Documents/curso-git-e-github/code/projeto1 (master)
$ git push https://github.com/miguelfmorandini/curso-git-github.git master
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 12 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (18/18), 1.79 KiB | 458.00 KiB/s, done.
Total 18 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/miguelfmorandini/curso-git-github/pull/new/master
remote:
To https://github.com/miguelfmorandini/curso-git-github.git
 * [new branch]      master -> master
```

Imagem 11. Linkando com repositório remoto

- `git remote add origin link-do-repositorio-github`: define o alias do repositório remoto como 'origin'.
- `git push origin branch-em-que-deseja-alterar`: envia para o GitHub as alterações.

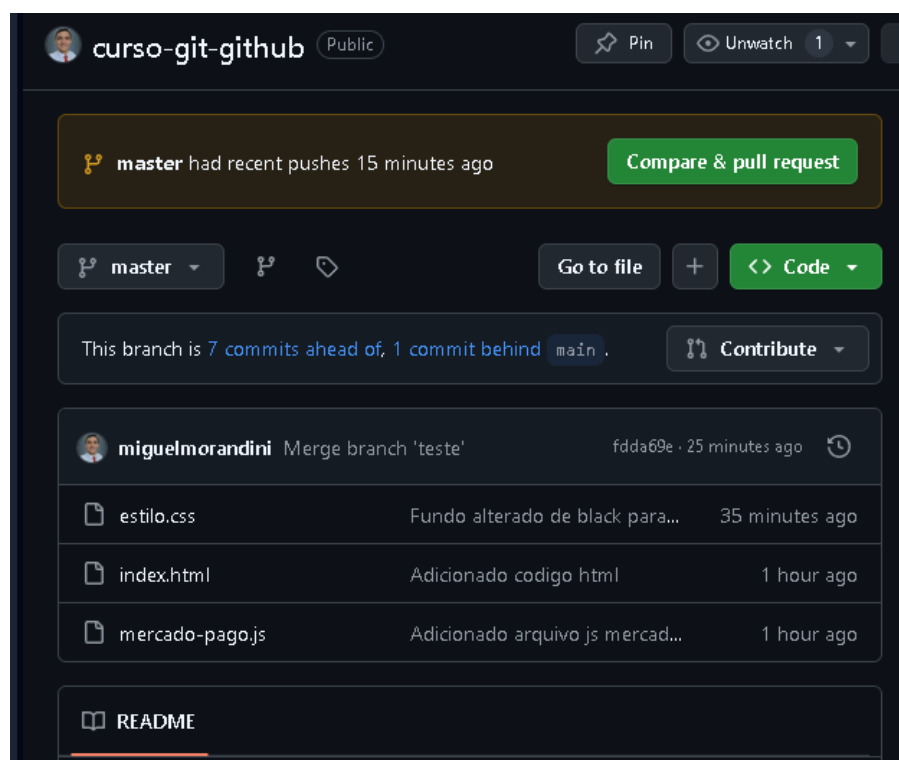


Imagem 12. Tudo enviado ao GitHub

Pull requests: Criando em '**Compare & pull request**', como na imagem acima, abrirá a tela de descrição da pull requests, onde se adiciona um título e uma descrição. E depois, criando

em **‘Create pull request’**, ela é enviada. O manager do projeto, pode analisá-la, comentá-la e, se estiver de acordo fazer o merge com a branch principal do projeto.

Fork: um fork é a cópia de um projeto, para posterior edição e contribuição ao repositório original. Para criar um fork, basta ir até o **repositório** desejado, clicar em **‘Fork’**, depois em **‘Create a new fork’**, como demonstra a imagem 13.

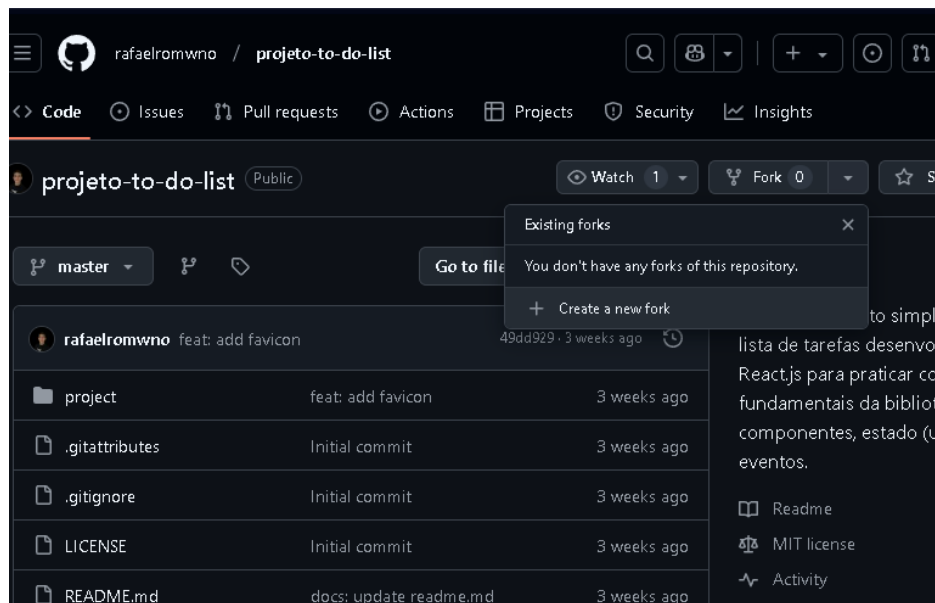


Imagem 13. Criando fork

Depois basta, adicionar um nome e criá-lo em definitivo.

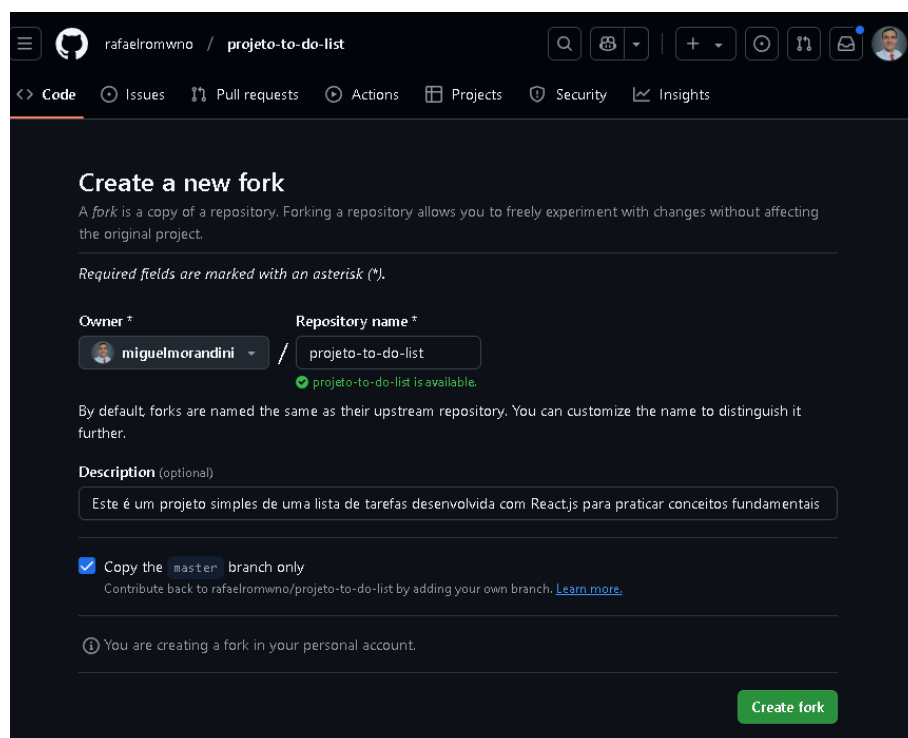


Imagem 14. Criando fork

Pronto, o fork está criado em seu perfil do GitHub, agora basta dar um comando `git clone`, no Git Bash e pode-se trabalhar com o repositório gerado do fork normalmente fazendo as alterações desejadas. Depois, pode-se se criar uma pull request para o desenvolvedor inicial, se desejar adicionar uma feature ao projeto de origem, se ele concordar, ele deve fazer o merge para a branch principal.

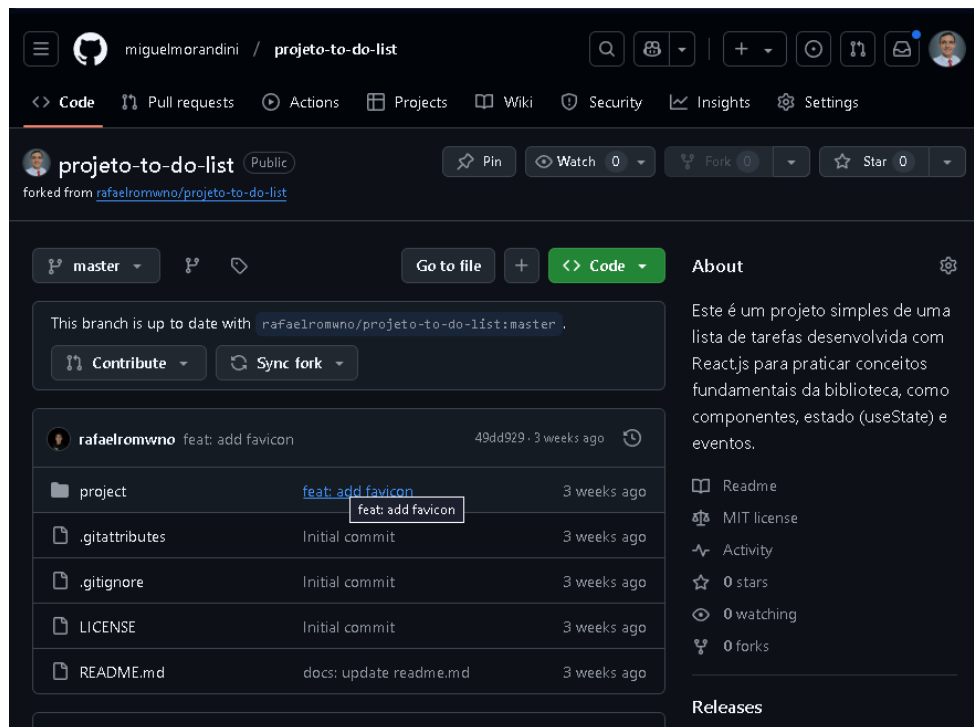


Imagem 15. Fork adicionado aos meus repositórios

Considerações Finais:

De forma detalhada, o curso me ajudou bastante, clareando minha mente principalmente em comandos de push, merge, criação de pull requests e forks e seu gerenciamento. Também me lembrou vários dos comandos básicos e como iniciar o Git e posteriormente o GitHub em uma máquina. Muito proveito este trabalho e a playlist indicada!

Links do Material Utilizado:

Playlist do Curso de Git e GitHub: https://youtube.com/playlist?list=PLcoYAcR89n-qbO7YAVj5S0aIABLIis_QVU&si=x4IdoPz1kH-OryRD

Repositório base para os comandos realizados: <https://github.com/miguelfmorandini/curso-git-github.git>

Fork realizado: <https://github.com/miguelfmorandini/projeto-to-do-list.git>

Repositório originador do Fork: <https://github.com/rafaelromwno/projeto-to-do-list.git>