

Curso Git e GitHub

aulaEAD (Professor José de Assis)

<https://www.youtube.com/watch?v=FF1f4bKYhoo&list=PLbEOwbQR9lqzK14I7OOeREEIE4k6rjglj>

Resumo do curso feito por Roberto Pinheiro

Aula 01 - O que é git? - O que é GitHub? Porquê aprender git?

Git é um sistema de controle de versões. **GitHub** é uma espécie de rede social de projetos.

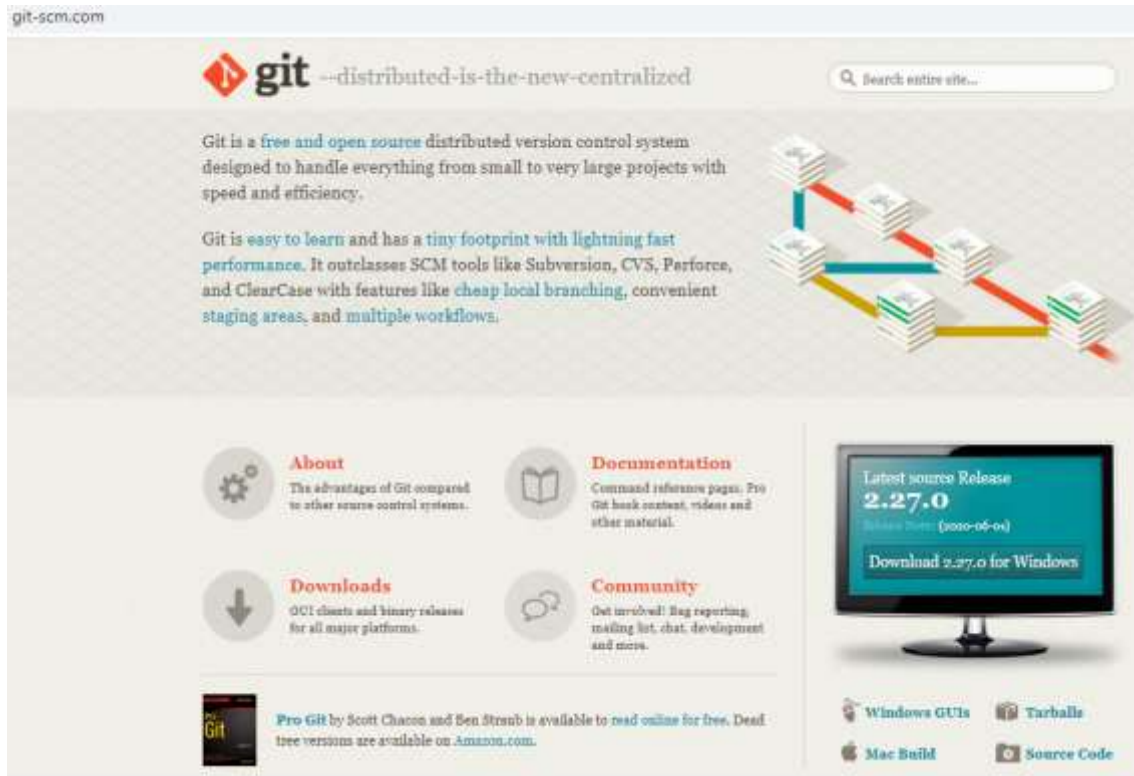
Outras alternativas ao GitHub: Subversion, Git Lab. Porém Git e Github são os mais populares e os mais usados.

Vantagens de usar essa ferramenta:

1. **Backup**: armazena uma cópia de seus projetos nas nuvens.
2. **Controle de versões**: o Git documenta e armazena cada modificação que é feita no código, através de commits.
3. **Trabalhar em equipe**: é uma das maiores vantagens. Um projeto pode ser clonado e trabalhado em equipe.
4. **Portfolio**: permite criar um portfolio com todos os seus projetos.

Aula 02 - Instalação e configuração do GIT no Windows

- Acesse o endereço git.scm.com, faça o download do aplicativo e instale em seu computador.



Verificando a versão do Git instalado

```
betol@DESKTOP-RROR61J MINGW64 ~
$ git version
git version 2.25.1.windows.1
```

Configurações iniciais

```
git config --global user.name "Roberto Pinheiro"
git config --global user.email "betopinheiro1005@yahoo.com.br"
```

```
betol@DESKTOP-RROR61J MINGW64 ~
$ git config --global user.name "Roberto Pinheiro"

betol@DESKTOP-RROR61J MINGW64 ~
$ git config --global user.email "betopinheiro1005@yahoo.com.br"

betol@DESKTOP-RROR61J MINGW64 ~
$ git config user.name
Roberto Pinheiro

betol@DESKTOP-RROR61J MINGW64 ~
$ git config user.email
betopinheiro1005@yahoo.com.br
```

Configurando o editor padrão

```
git config --global core.editor "C:\Users\beto1\AppData\Local\Programs\Microsoft VS Code\Code.exe"
```

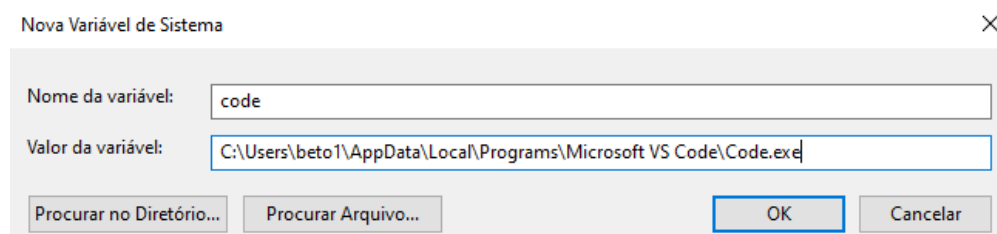
```
git config core.editor
```

```
beto1@DESKTOP-RROR61J MINGW64 ~
$ git config --global core.editor "C:\Users\beto1\AppData\Local\Programs\Microsoft VS Code\Code.exe"

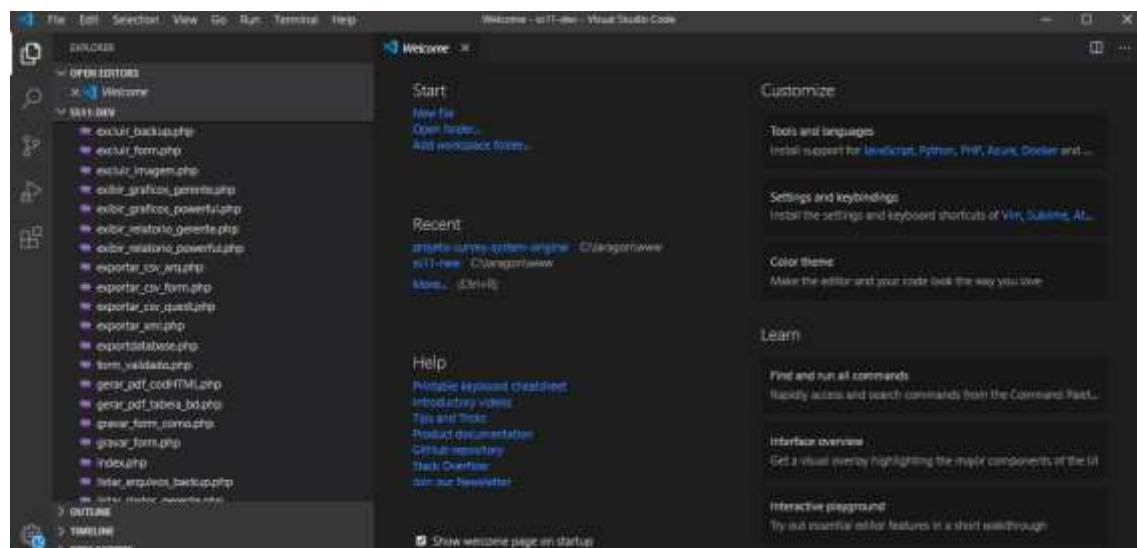
beto1@DESKTOP-RROR61J MINGW64 ~
$ git config core.editor
C:\Users\beto1\AppData\Local\Programs\Microsoft VS Code\Code.exe
```

Em “**variáveis de ambiente**”, criar uma variável de sistema com o nome "**code**" e com o seguinte caminho:

C:\Users\beto1\AppData\Local\Programs\Microsoft VS Code\Code.exe



```
beto1@DESKTOP-RROR61J MINGW64 ~
$ code
```



```
beto1@DESKTOP-RROR61J MINGW64 ~
$ pwd
/c/Users/beto1

beto1@DESKTOP-RROR61J MINGW64 ~
$ cd /c

beto1@DESKTOP-RROR61J MINGW64 /c
$ |
```

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git
$ cd /c

beto1@DESKTOP-RROR61J MINGW64 /c
$ pwd
/c

beto1@DESKTOP-RROR61J MINGW64 /c
$ mkdir testes_git

beto1@DESKTOP-RROR61J MINGW64 /c
$ cd testes_git

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git
$ pwd
/c/testes_git

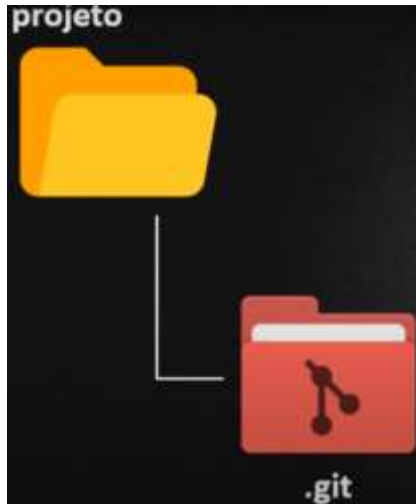
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git
$ touch teste1

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git
$ ls
teste1

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git
$ |
```

Aula 03 - Iniciando um repositório local / comandos básicos

O Git cria um repositório local no diretório do projeto. Este repositório local pode ser clonado posteriormente para um repositório remoto. Pode também ser feito o oposto.



Para criar um repositório local, na pasta do projeto digite:

`git init`

```
beto1@DESKTOP-RR0R61J MINGW64 ~
$ pwd
/c/Users/beto1

beto1@DESKTOP-RR0R61J MINGW64 ~
$ cd /c/testes_git

beto1@DESKTOP-RR0R61J MINGW64 /c/testes_git
$ pwd
/c/testes_git

beto1@DESKTOP-RR0R61J MINGW64 /c/testes_git
$ ls
teste1

beto1@DESKTOP-RR0R61J MINGW64 /c/testes_git
$ git init
Initialized empty Git repository in C:/testes_git/.git/

beto1@DESKTOP-RR0R61J MINGW64 /c/testes_git (master)
$ |
```

`ls -a`

```
beto1@DESKTOP-RR0R61J MINGW64 /c/testes_git (master)
$ ls -a
./ ../ .git/

beto1@DESKTOP-RR0R61J MINGW64 /c/testes_git (master)
$ |
```

Grafo de commits

Grafo é uma estrutura que contém objetos relacionados. **Commit** é um comentário que se escreve sempre que se modifica alguma coisa no projeto.



O Git precisa gerenciar as mudanças do projeto. Para isso existe uma espécie de container onde devemos colocar todos os arquivos e pastas que foram criados ou modificados. Isso é feito com o comando:

git add

Para verificar o que está armazenado no container é usado o comando:

git status

Para identificar e armazenar esse container no repositório local é usado o comando:

git commit

O **.git** é o repositório e dentro dele serão armazenados os containers que são identificados. Dentro desses containers há o conteúdo do seu projeto.

```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ touch a.txt b.txt

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ ls -a
./ ../ .git/ a.txt b.txt

```

Adicionando arquivos ao container

```

betol@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git status
On branch master

No commits yet

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

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

betol@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git add .

betol@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   a.txt
        new file:   b.txt

```

Identificando e armazenando o container no repositório

```

betol@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git commit -m "Criado os arquivos a.txt e b.txt"
[master (root-commit) e610e2e] Criado os arquivos a.txt e b.txt
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 a.txt
create mode 100644 b.txt

```

```

betol@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git status
On branch master
nothing to commit, working tree clean

```

Gerenciando o repositório

```

betol@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git log
commit e610e2e5df973d4c316d1d99405a8e341022b12f (HEAD -> master)
Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
Date:   Sun Jun 28 20:06:22 2020 -0300

    Criado os arquivos a.txt e b.txt

```

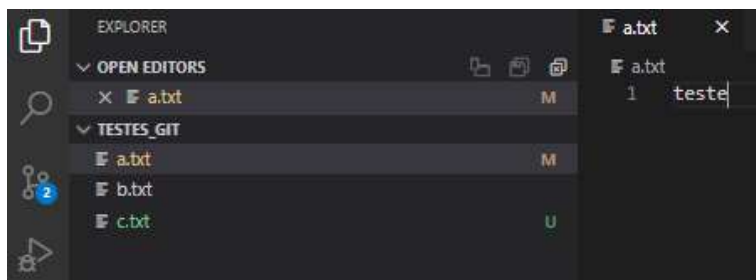
Fazendo alterações no projeto

Criando um novo arquivo

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ touch c.txt

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ code .
```

Alterando o arquivo a.txt



Criando um novo container e adicionando-o ao repositório

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (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:   a.txt

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

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

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git add .

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   a.txt
    new file:   c.txt

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git commit -m "Modificado o arquivo a.txt e criado o arquivo c.txt"
[master 1ef84d9] Modificado o arquivo a.txt e criado o arquivo c.txt
 2 files changed, 1 insertion(+)
 create mode 100644 c.txt
```


Gerenciando os commits criados

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git status
On branch master
nothing to commit, working tree clean

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git log
commit 1ef84d93123810df7717f3e118ce1b49189e9907 (HEAD -> master)
Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
Date:   Sun Jun 28 20:18:09 2020 -0300

    Modificado o arquivo a.txt e criado o arquivo c.txt

commit e610e2e5df973d4c316d1d99405a8e341022b12f
Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
Date:   Sun Jun 28 20:06:22 2020 -0300

    Criado os arquivos a.txt e b.txt
```

Exibindo os logs de uma maneira mais simples

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git log --oneline
1ef84d9 (HEAD -> master) Modificado o arquivo a.txt e criado o arquivo c.txt
e610e2e Criado os arquivos a.txt e b.txt
```

Aula 04 - Navegando entre as versões do projeto

Controle de versões

- Rastrear mudanças
- Desfazer alterações
- Recuperar versões anteriores do projeto

Head → Master

É pelo **head** que conseguimos rastrear os commits e recuperar uma versão anterior do projeto.

Master é o ramo principal.

`git log --graph`



```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git log --graph
* commit 1ef84d93123810df7717f3e118ce1b49189e9907 (HEAD -> master)
  Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
  Date:   Sun Jun 28 20:18:09 2020 -0300

    Modificado o arquivo a.txt e criado o arquivo c.txt

* commit e610e2e5df973d4c316d1d99405a8e341022b12f
  Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
  Date:   Sun Jun 28 20:06:22 2020 -0300

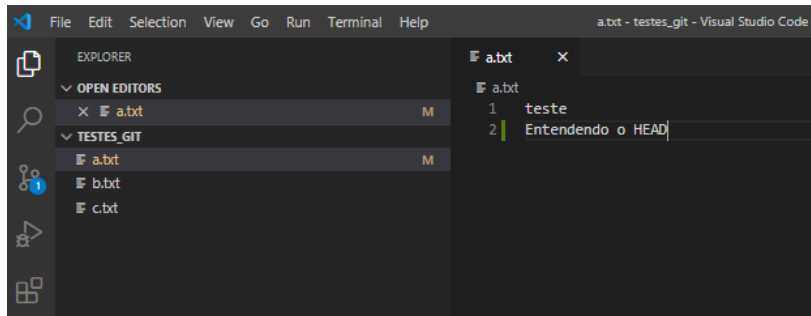
    Criado os arquivos a.txt e b.txt
  
```

```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git log --oneline --graph
* 1ef84d9 (HEAD -> master) Modificado o arquivo a.txt e criado o arquivo c.txt
* e610e2e Criado os arquivos a.txt e b.txt
  
```

Head é o ponto onde o seu projeto se encontra no grafo de commits. Sempre vai ser o último commit.

.code



Para adicionar ao container e simultaneamente criar um novo commit

git commit -am "No arquivo a.txt adicionado a linha Entendendo o HEAD"

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git commit -am "No arquivo a.txt adicionado a linha Entendendo o HEAD"
[master 4d1e494] No arquivo a.txt adicionado a linha Entendendo o HEAD
1 file changed, 2 insertions(+), 1 deletion(-)

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Gerenciando os commits criados

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git log --graph
* commit 4d1e494cf50d0d7cf66d82941ebc3c5a018f8fc4 (HEAD -> master)
  Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
  Date: Sun Jun 28 20:59:38 2020 -0300

    No arquivo a.txt adicionado a linha Entendendo o HEAD

* commit 1ef84d93123810df7717f3e118ce1b49189e9907
  Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
  Date: Sun Jun 28 20:18:09 2020 -0300

    Modificado o arquivo a.txt e criado o arquivo c.txt

* commit e610e2e5df973d4c316d1d99405a8e341022b12f
  Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
  Date: Sun Jun 28 20:06:22 2020 -0300

    Criado os arquivos a.txt e b.txt
```

O Git não gera nova cópia do projeto a cada novo commit. O Git rastreia as alterações do projeto, armazenando de forma inteligente somente as mudanças que foram feitas.

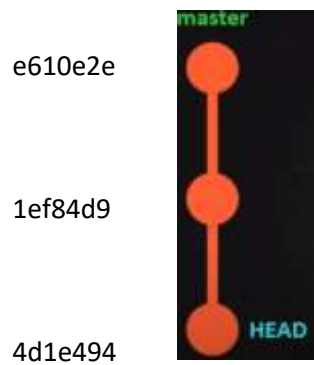
Identificando o ramo em que o projeto se encontra

Para identificar o ramo em que o projeto se encontra entre com o seguinte comando:

`git branch`

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git branch
* master
```

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git log --oneline
4d1e494 (HEAD -> master) No arquivo a.txt adicionado a linha Entendendo o HEAD
1ef84d9 Modificado o arquivo a.txt e criado o arquivo c.txt
e610e2e Criado os arquivos a.txt e b.txt
```



Rastreando as mudanças e recuperando uma versão anterior

Devemos apontar o HEAD para o respectivo commit.

`git checkout 1ef84d9`

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git checkout 1ef84d9
Note: switching to '1ef84d9'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

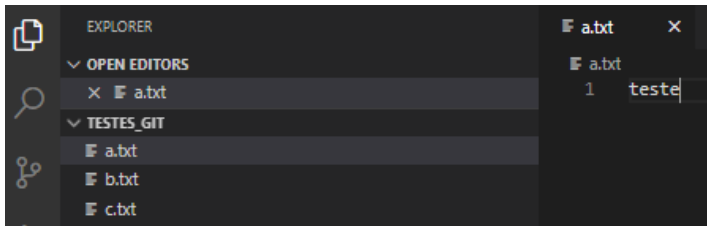
  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 1ef84d9 Modificado o arquivo a.txt e criado o arquivo c.txt
```



```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git ((1ef84d9...))
$ git log --oneline
1ef84d9 (HEAD) Modificado o arquivo a.txt e criado o arquivo c.txt
e610e2e Criado os arquivos a.txt e b.txt

```

```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git ((1ef84d9...))
$ git branch
* (HEAD detached at 1ef84d9)
master

```

Retornando à última versão

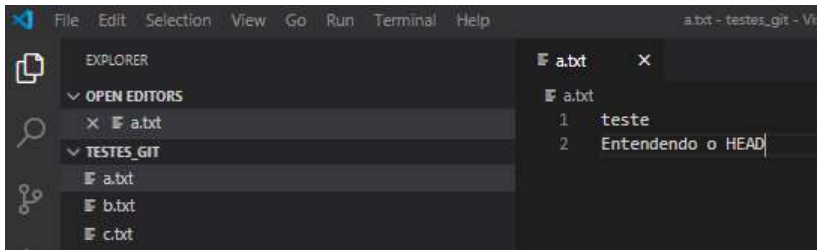
git checkout master

```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git ((1ef84d9...))
$ git branch
* (HEAD detached at 1ef84d9)
master

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git ((1ef84d9...))
$ git checkout master
Previous HEAD position was 1ef84d9 Modificado o arquivo a.txt e criado o arquivo c.txt
Switched to branch 'master'

```

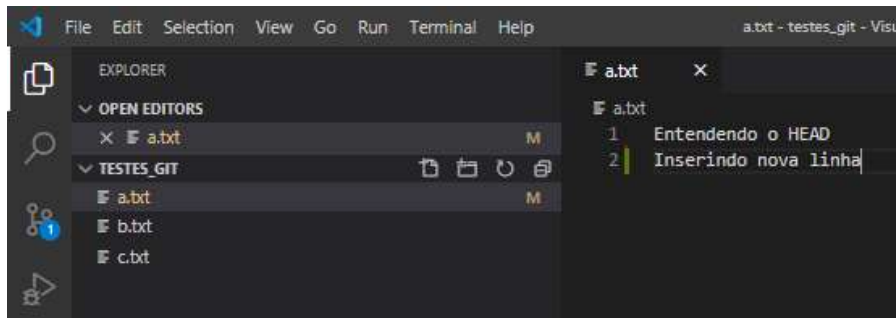


```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git log --oneline
4d1e494 (HEAD -> master) No arquivo a.txt adicionado a linha Entendendo o HEAD
1ef84d9 Modificado o arquivo a.txt e criado o arquivo c.txt
e610e2e Criado os arquivos a.txt e b.txt

```

Desfazendo alterações



Visualizando as últimas mudanças antes do commit

git diff

```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git diff
diff --git a/a.txt b/a.txt
index ccab5e6..7c36848 100644
--- a/a.txt
+++ b/a.txt
@@ -1,2 +1,2 @@
-teste
-Entendendo o HEAD
\ No newline at end of file
+Entendendo o HEAD
+Inserindo nova linha
\ No newline at end of file

```

Para descartar as mudanças feitas use o comando:

git checkout <file>

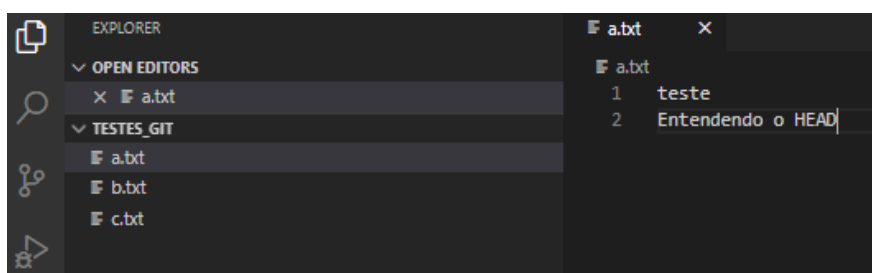
```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (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:   a.txt

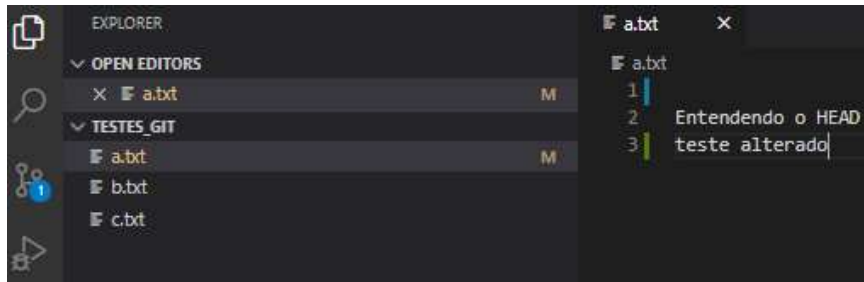
no changes added to commit (use "git add" and/or "git commit -a")

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git checkout a.txt
Updated 1 path from the index

```



Modificando novamente o arquivo a.txt



```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git diff
diff --git a/a.txt b/a.txt
index ccab5e6..721adb5 100644
--- a/a.txt
+++ b/a.txt
@@ -1,2 +1,3 @@
-teste
-Entendendo o HEAD
\ No newline at end of file
+Entendendo o HEAD
+teste alterado
\ No newline at end of file
```

Adicionando o arquivo a.txt alterado ao container

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (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:   a.txt

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

Adicionando o conteudo ao container

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git add a.txt

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   a.txt
```

Removendo o conteúdo do container

git reset HEAD

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git reset HEAD
Unstaged changes after reset:
M   a.txt
```

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (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:   a.txt
```

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git diff
diff --git a/a.txt b/a.txt
index ccab5e6..721adb5 100644
--- a/a.txt
+++ b/a.txt
@@ -1,2 +1,3 @@
-teste
-Entendendo o HEAD
\ No newline at end of file
+
+Entendendo o HEAD
+teste alterado
\ No newline at end of file
```

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (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:   a.txt

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

Criando um novo commit

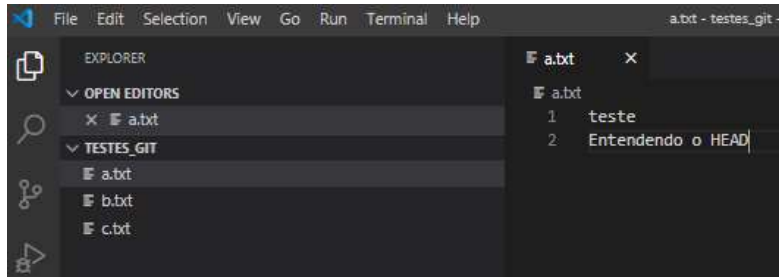
```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git commit -am "Removendo um commit"
[master 976a05a] Removendo um commit
1 file changed, 3 insertions(+), 2 deletions(-)

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git log --oneline
976a05a (HEAD -> master) Removendo um commit
4d1e494 No arquivo a.txt adicionado a linha Entendendo o HEAD
1ef84d9 Modificado o arquivo a.txt e criado o arquivo c.txt
e610e2e Criado os arquivos a.txt e b.txt
```


Removendo um commit

```
git reset --hard 4d1e494
```

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git reset --hard 4d1e494
HEAD is now at 4d1e494 No arquivo a.txt adicionado a linha Entendendo o HEAD
```



```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git log --oneline
4d1e494 (HEAD -> master) No arquivo a.txt adicionado a linha Entendendo o HEAD
1ef84d9 Modificado o arquivo a.txt e criado o arquivo c.txt
e610e2e Criado os arquivos a.txt e b.txt
```

Aula 05 - Criando ramificações do projeto

Criar uma ramificação permite que testes e mudanças no projeto sejam feitas sem interferir no projeto original.

É possível fazer a fusão das ramificações implantando as mudanças em uma única versão do projeto.

Para verificar o ramo em que se encontra digite:

`git branch`

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git branch
* master
```

Por padrão, quando se cria um projeto, o Git automaticamente cria o ramo “**master**”.

Criando uma nova ramificação

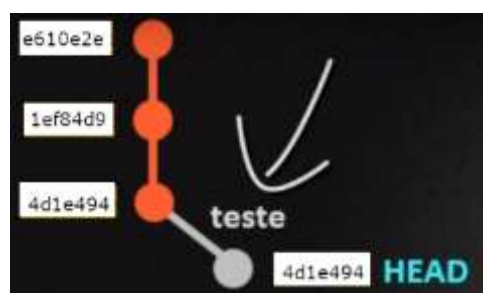
Para criar uma nova ramificação do projeto, entre com o comando:

`git checkout -b <nome_amo>`

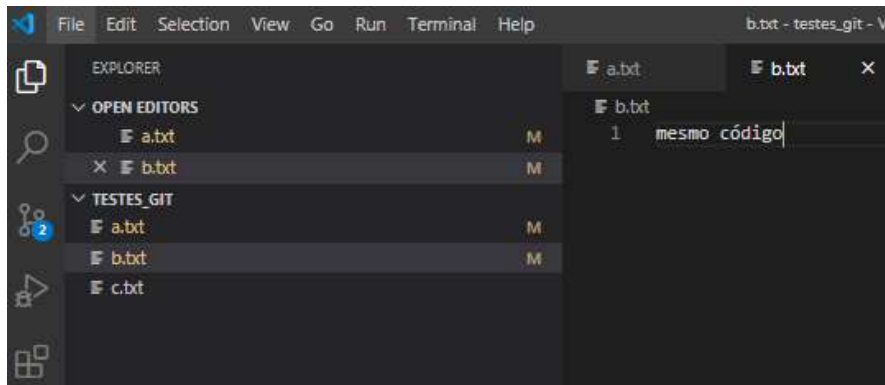
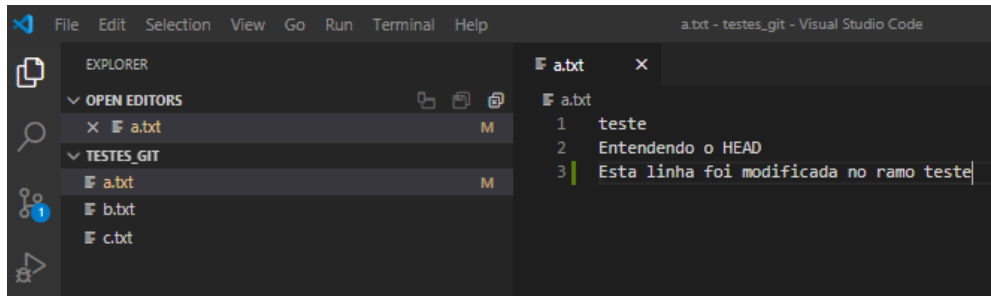
```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git checkout -b teste
Switched to a new branch 'teste'

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (teste)
$ git log --oneline
4d1e494 (HEAD -> teste, master) No arquivo a.txt adicionado a linha Entendendo o
HEAD
1ef84d9 Modificado o arquivo a.txt e criado o arquivo c.txt
e610e2e Criado os arquivos a.txt e b.txt
```

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (teste)
$ git branch
  master
* teste
```



code .



```

betoi@DESKTOP-RROR61J MINGW64 /c/testes_git (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:   a.txt
        modified:   b.txt

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

betoi@DESKTOP-RROR61J MINGW64 /c/testes_git (teste)
$ git add .

betoi@DESKTOP-RROR61J MINGW64 /c/testes_git (teste)
$ git status
On branch teste
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   a.txt
        modified:   b.txt

betoi@DESKTOP-RROR61J MINGW64 /c/testes_git (teste)
$ |

```

```

betoi@DESKTOP-RROR61J MINGW64 /c/testes_git (teste)
$ git commit -m "Modificado os arquivos a.txt e b.txt"
[teste 1f6ba30] Modificado os arquivos a.txt e b.txt
 2 files changed, 3 insertions(+), 1 deletion(-)

betoi@DESKTOP-RROR61J MINGW64 /c/testes_git (teste)
$ git log --oneline
1f6ba30 (HEAD -> teste) Modificado os arquivos a.txt e b.txt
4d1e494 (master) No arquivo a.txt adicionado a linha Entendendo o HEAD
1ef84d9 Modificado o arquivo a.txt e criado o arquivo c.txt
e610e2e Criado os arquivos a.txt e b.txt

```

```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (teste)
$ touch d.txt

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (teste)
$ git status
On branch teste
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        d.txt

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

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (teste)
$ git add .

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (teste)
$ git status
On branch teste
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   d.txt

```

```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (teste)
$ git commit -m "Criado o arquivo d.txt"
[teste d1a69b4] Criado o arquivo d.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 d.txt

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (teste)
$ git log --oneline
d1a69b4 (HEAD -> teste) Criado o arquivo d.txt
1f6ba30 Modificado os arquivos a.txt e b.txt
4d1e494 (master) No arquivo a.txt adicionado a linha Entendendo o HEAD
1ef84d9 Modificado o arquivo a.txt e criado o arquivo c.txt
e610e2e Criado os arquivos a.txt e b.txt

```

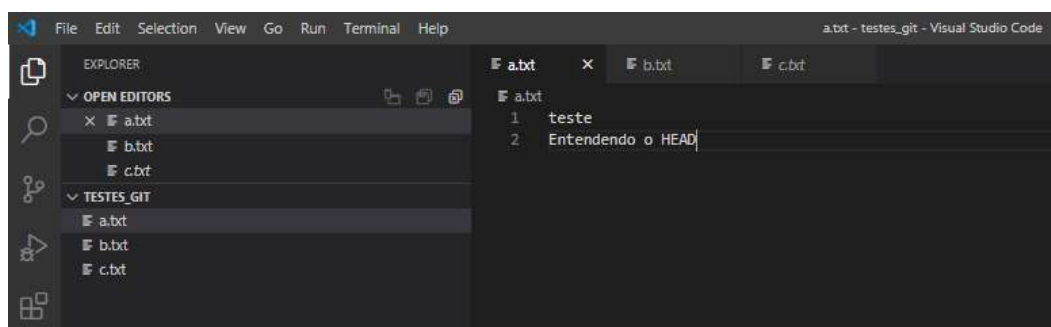
Retornando ao ramo master

git checkout master

```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (teste)
$ git checkout master
Switched to branch 'master'

```



Fazendo alterações nos arquivos do ramo master

```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (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:   a.txt
        modified:   b.txt
        modified:   c.txt

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

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git add .

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   a.txt
        modified:   b.txt
        modified:   c.txt

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git commit -m "Modificado os arquivos a.txt b.txt e c.txt"
[master 10382d6] Modificado os arquivos a.txt b.txt e c.txt
 3 files changed, 4 insertions(+), 1 deletion(-)

```

git log --oneline --graph --all

```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git log --oneline --graph --all
* 10382d6 (HEAD -> master) Modificado os arquivos a.txt b.txt e c.txt
| * d1a69b4 (teste) Criado o arquivo d.txt
| * 1f6ba30 Modificado os arquivos a.txt e b.txt
|/
* 4d1e494 No arquivo a.txt adicionado a linha Entendendo o HEAD
* 1ef84d9 Modificado o arquivo a.txt e criado o arquivo c.txt
* e610e2e Criado os arquivos a.txt e b.txt

```

Unindo o ramo teste ao ramo master (fusão dos dois ramos)

É necessário estar no ramo master.

git merge <ramo>

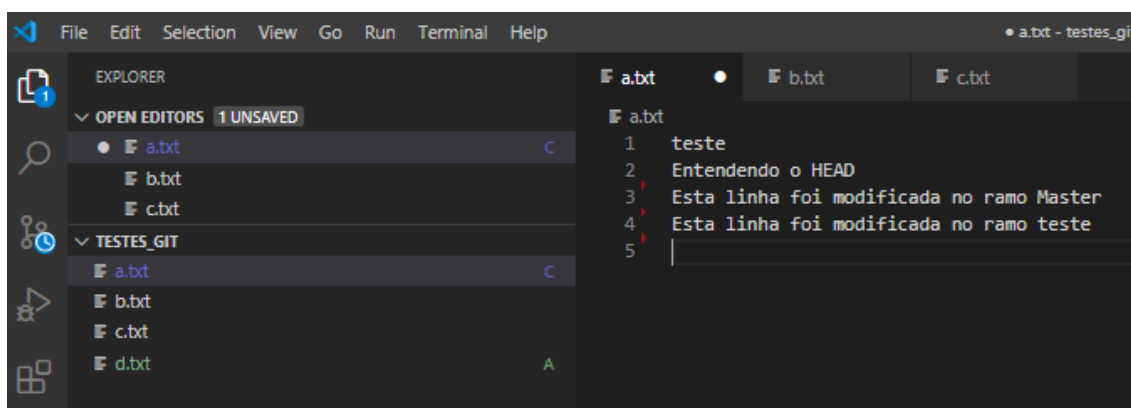
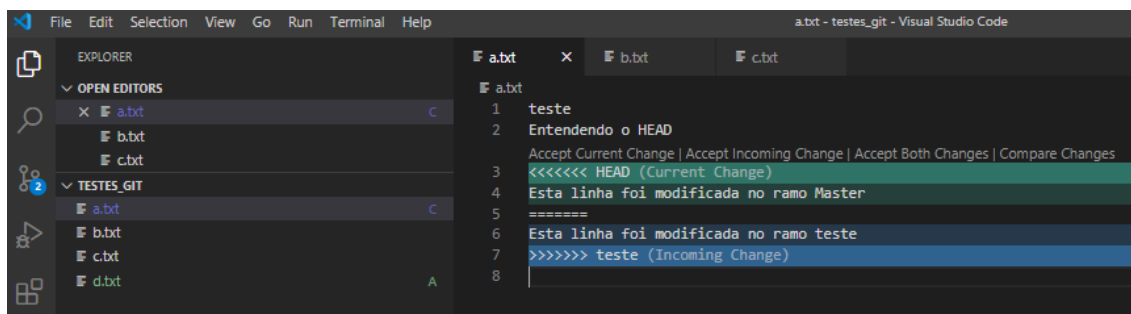
git merge teste

```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git merge teste
Auto-merging a.txt
CONFLICT (content): Merge conflict in a.txt
Automatic merge failed; fix conflicts and then commit the result.

```

Resolvendo o conflito



```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
  new file:   d.txt

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   a.txt

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master|MERGING)
$ git add .
```

```
beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master|MERGING)
$ git status
On branch master
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
  modified:   a.txt
  new file:   d.txt

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master|MERGING)
$ git commit -m "Fusão dos ramos e resolução do conflito no arquivo a.txt"
[master a244256] Fusão dos ramos e resolução do conflito no arquivo a.txt
```

```

beto1@DESKTOP-RR0R61J MINGW64 /c/testes_git (master)
$ git log --oneline
a244256 (HEAD -> master) Fusão dos ramos e resolução do conflito no arquivo a.txt
10382d6 Modificado os arquivos a.txt b.txt e c.txt
d1a69b4 (teste) Criado o arquivo d.txt
1f6ba30 Modificado os arquivos a.txt e b.txt
4d1e494 No arquivo a.txt adicionado a linha Entendendo o HEAD
1ef84d9 Modificado o arquivo a.txt e criado o arquivo c.txt
e610e2e Criado os arquivos a.txt e b.txt

```

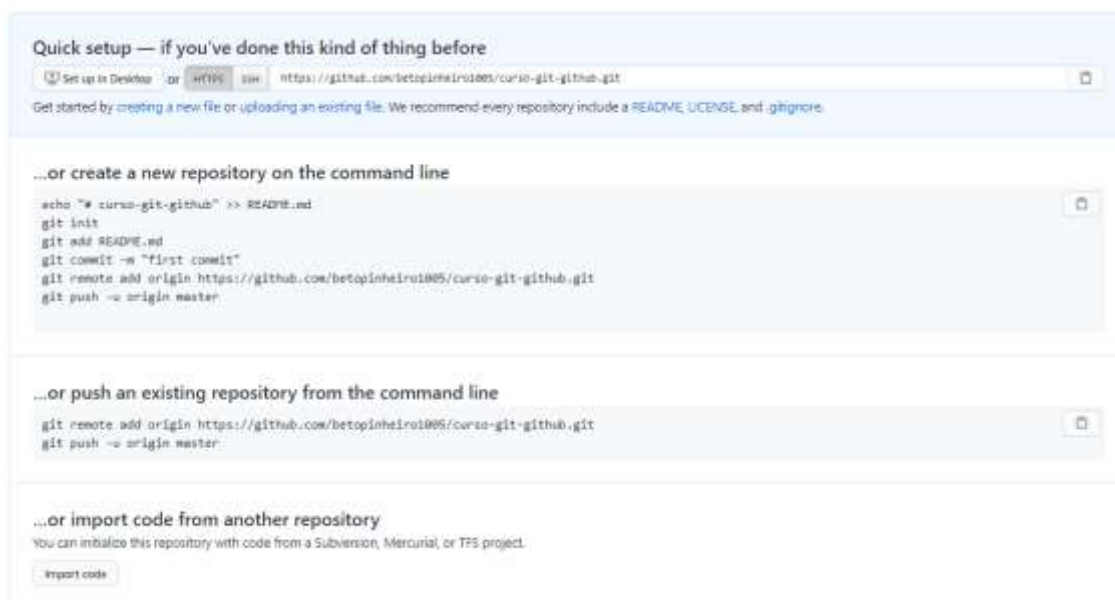
```

beto1@DESKTOP-RR0R61J MINGW64 /c/testes_git (master)
$ git log --oneline --graph --all
*   a244256 (HEAD -> master) Fusão dos ramos e resolução do conflito no arquivo a.txt
| \
|  * d1a69b4 (teste) Criado o arquivo d.txt
|  * 1f6ba30 Modificado os arquivos a.txt e b.txt
* | 10382d6 Modificado os arquivos a.txt b.txt e c.txt
|/
* 4d1e494 No arquivo a.txt adicionado a linha Entendendo o HEAD
* 1ef84d9 Modificado o arquivo a.txt e criado o arquivo c.txt
* e610e2e Criado os arquivos a.txt e b.txt

```

Aula 06 - Enviando um projeto local para o GitHub

Ligando o projeto ao repositório remoto



git remote add origin https://github.com/betopinheiro1005/curso-git-github.git

git push -u origin master

```

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git status
On branch master
nothing to commit, working tree clean

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git remote

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git remote add origin https://github.com/betopinheiro1005/curso-git-github.git

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git push -u origin master
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 2 threads
Compressing objects: 100% (17/17), done.
Writing objects: 100% (22/22), 2.00 KiB | 53.00 KiB/s, done.
Total 22 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/betopinheiro1005/curso-git-github.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git remote
origin

beto1@DESKTOP-RROR61J MINGW64 /c/testes_git (master)
$ git remote -v
origin https://github.com/betopinheiro1005/curso-git-github.git (fetch)
origin https://github.com/betopinheiro1005/curso-git-github.git (push)

```


Atualizando a página no Github:

The screenshot shows the GitHub interface for the repository 'betopinheiro1005 / curso-git-github'. The repository has 1 watcher, 0 stars, and 0 forks. The main navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The repository is currently on the 'master' branch. A commit history table shows four commits by 'betopinheiro1005' from 2 days ago, detailing file changes to 'a.txt', 'b.txt', 'c.txt', and 'd.txt'. On the right, the 'About' section identifies the repository as 'Curso Git/Github - Professor José de Assis - aulasad.com'. Below this, the 'Releases' section states 'No releases published' with a link to 'Create a new release'. The 'Packages' section at the bottom indicates 'No packages published' with a link to 'Publish your first package'. A blue banner at the bottom of the commit list encourages adding a README file.

betopinheiro1005 / curso-git-github

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Branch: master

Go to file Add file + Code -

betopinheiro1005 committed 2 days ago 7 commits 1 branch 0 tags

a.txt	Adição dos arquivos e resolução do conflito no arquivo a.txt	2 days ago
b.txt	Modificação no arquivo a.txt e b.txt	4 days ago
c.txt	Modificação no arquivo a.txt e b.txt	4 days ago
d.txt	Criação do arquivo d.txt	4 days ago

Help people interested in this repository understand your project by adding a README. Add a README

About

Curso Git/Github - Professor José de Assis - aulasad.com

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Aula 07 - Criando um repositório no Github e clonando no Windows

Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere?


[Import a repository.](#)

Owner * betopinheiro1005 / Repository name * roboticapratica ✓

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

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.

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

☒ **Initialize this repository with a README**
 This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None ⓘ

Create repository

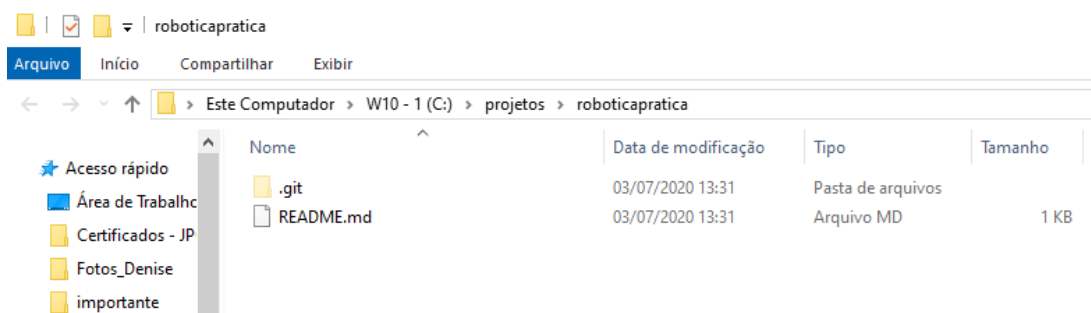
```

beto1@DESKTOP-RROR61J MINGW64 /c
$ mkdir projetos

beto1@DESKTOP-RROR61J MINGW64 /c
$ cd projetos
  
```

```

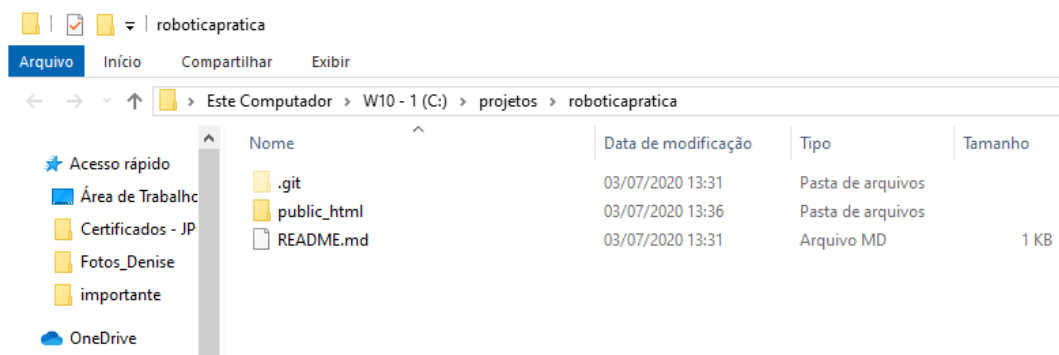
beto1@DESKTOP-RROR61J MINGW64 /c/projetos
$ git clone https://github.com/betopinheiro1005/roboticapratica.git
Cloning into 'roboticapratica'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 622 bytes | 5.00 KiB/s, done.
  
```



```

beto1@DESKTOP-RR0R61J MINGW64 /c/projetos/roboticapratica (master)
$ mkdir public_html

```



```

beto1@DESKTOP-RR0R61J MINGW64 /c/projetos/roboticapratica (master)
$ touch public_html/index.html

```

```

beto1@DESKTOP-RR0R61J MINGW64 /c/projetos/roboticapratica (master)
$ cd public_html

beto1@DESKTOP-RR0R61J MINGW64 /c/projetos/roboticapratica/public_html (master)
$ ls
index.html

```

```

beto1@DESKTOP-RR0R61J MINGW64 /c/projetos/roboticapratica (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

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

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

beto1@DESKTOP-RR0R61J MINGW64 /c/projetos/roboticapratica (master)
$ git add *

beto1@DESKTOP-RR0R61J MINGW64 /c/projetos/roboticapratica (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   public_html/index.html

```

```

beto1@DESKTOP-RR0R61J MINGW64 /c/projetos/roboticapratica (master)
$ git commit -m "Criado o diretório public_html e dentro dele o arquivo index.html"
[master e313385] Criado o diretório public_html e dentro dele o arquivo index.html
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 public_html/index.html

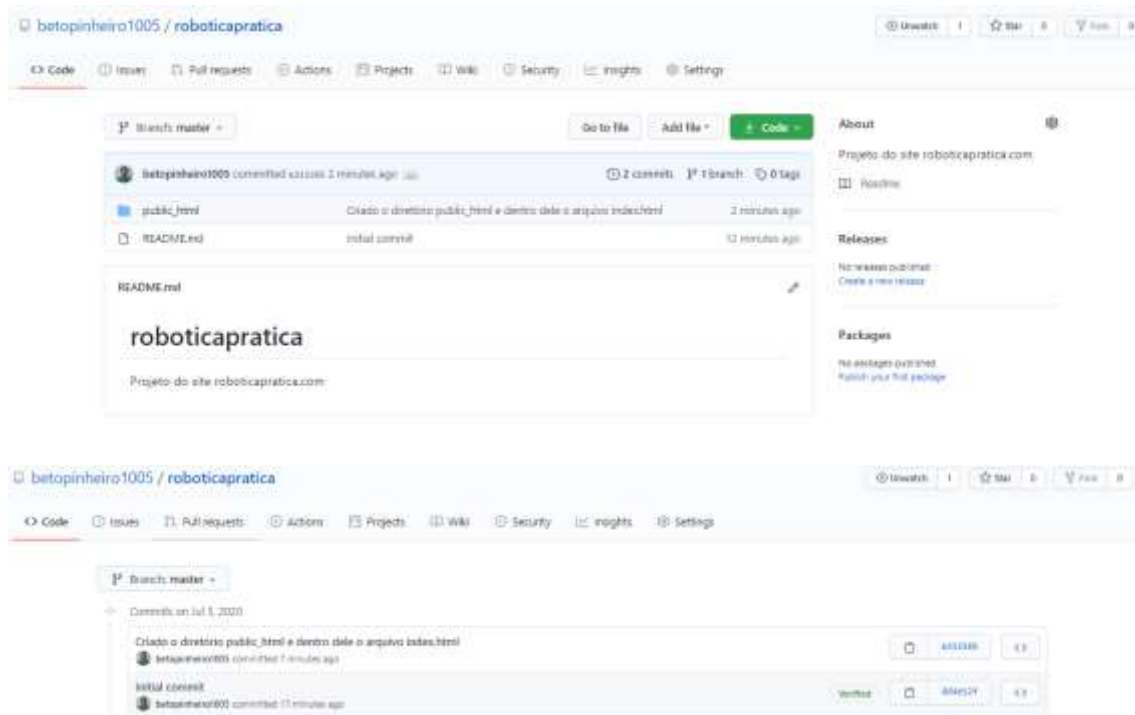
```

```

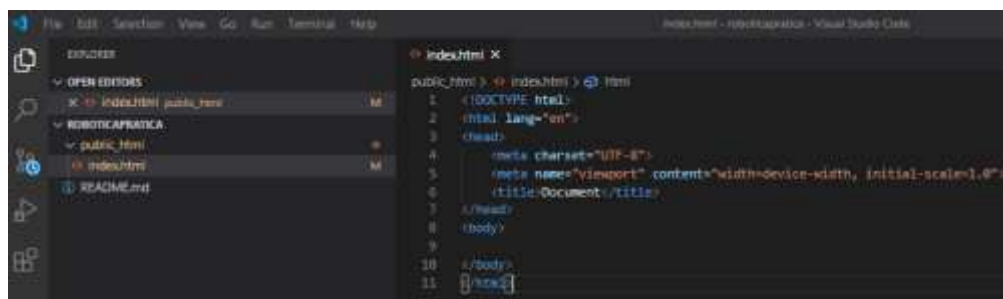
beto1@DESKTOP-RR0R61J MINGW64 /c/projetos/roboticapratica (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 380 bytes | 76.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/betopinheiro1005/roboticapratica.git
   dd4e12f..e313385  master -> master

```

Recarregando a página no Github:



code .
html:5



```

beto1@DESKTOP-RROR61J MINGW64 /c/projetos/roboticapratica (master)
$ git status
On branch master
Your branch is up to date with 'origin/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:   public_html/index.html

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

beto1@DESKTOP-RROR61J MINGW64 /c/projetos/roboticapratica (master)
$ git add *

beto1@DESKTOP-RROR61J MINGW64 /c/projetos/roboticapratica (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   public_html/index.html

```

```

beto1@DESKTOP-RROR61J MINGW64 /c/projetos/roboticapratica (master)
$ git commit -m "Modificado o arquivo index.html"
[master 079cc11] Modificado o arquivo index.html
1 file changed, 11 insertions(+)

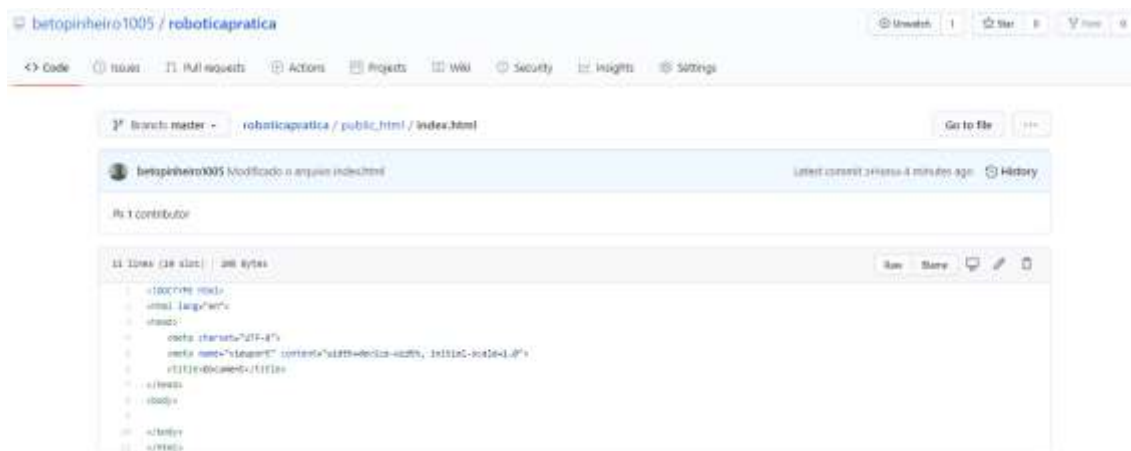
```

```

beto1@DESKTOP-RROR61J MINGW64 /c/projetos/roboticapratica (master)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 503 bytes | 100.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/betopinheiro1005/roboticapratica.git
e313385..079cc11 master -> master

```

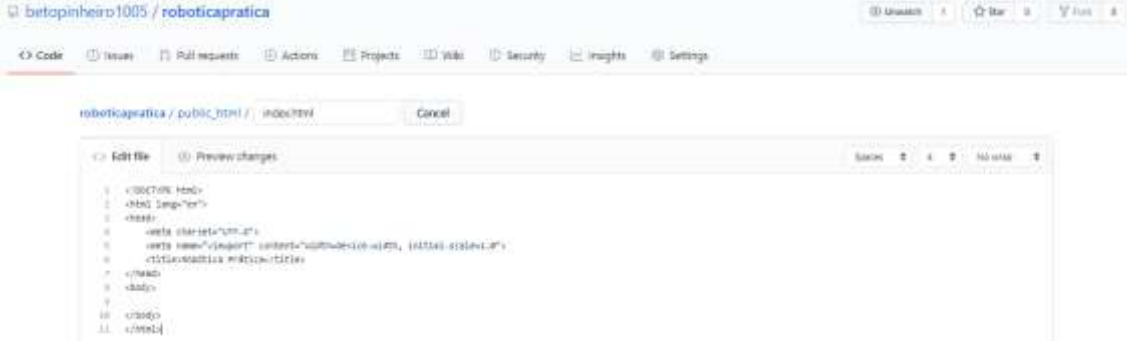
Recarregue a página no Github:



Aula 09 - Criando e editando arquivos e pastas no GITHUB - sincronizando alterações

Criando e editando arquivos e pastas no repositório remoto

Alterando o título da página index.html



betopinheiro1005 / roboticapratica

roboticapratica / public_html / index.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9
10 </body>
11 </html>
  
```

Commit changes

Alterado o título da página

Add an optional extended description...

☒ Commit directly to the `master` branch.

☐ Create a new branch for this commit and start a pull request: [Learn more about pull requests](#).

Commit changes **Cancel**

roboticapratica / public_html / index.html **Cancel**

<> Edit file		Preview changes
...	...	@@ -3,7 +3,7 @@
3	3	<head>
4	4	<meta charset="UTF-8">
5	5	<meta name="viewport" content="width=device-width, initial-scale=1.0">
6	-	<title>Document</title>
	6	+ <title>Robótica Prática</title>
7	7	</head>
8	8	<body>
9	9	
...	...	

betopinheiro1005 / roboticapratica

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Branch: master -

Commits on Jul 3, 2020

- Alterado o título da página
betopinheiro1005 committed 30 seconds ago
- Modificado o arquivo index.html
betopinheiro1005 committed 10 minutes ago
- Criado o diretório public_html e dentro dele o arquivo index.html
betopinheiro1005 committed 23 minutes ago
- Initial commit
betopinheiro1005 committed 33 minutes ago

Criando um novo arquivo

Commit new file

Criado o arquivo style.css

Add an optional extended description...

☒ Commit directly to the master branch

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit new file Cancel

betopinheiro1005 / roboticapratica

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

roboticapratica / public_html / style.css Cancel

Edit new file Preview

1

betopinheiro1005 / roboticapratica

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Branch: master - roboticapratica / public_html /

Go to file Add file +

betopinheiro1005 committed 1 minute ago

History

index.html	Alterado o título da página	6 minutes ago
style.css	Criado o arquivo style.css	now

Criando uma nova pasta

betopinheiro1005 / **roboticapratca**

<> Code ⓘ Issues 🔗 Pull requests ▶ Actions 📁 Projects 📖 Wiki ⓘ Security 📈 Insights ⚙ Settings

roboticapratca / public_html / scripts / Cancel

<> Edit new file ◁ Preview

1

Commit new file

Add an optional extended description...

☒ Commit directly to the `master` branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit new file Cancel

betopinheiro1005 committed 1 minute ago [History](#)

...		
📁 scripts	Criado a pasta scripts e o arquivo fixaMenu.js	now
📄 index.html	Alterado o título da página	11 minutes ago
📄 style.css	Criado o arquivo style.css	6 minutes ago

betopinheiro1005 / **roboticapratca** 🔍 Watch 1 🌟 Star 0 📄 Download 0

<> Code ⓘ Issues 🔗 Pull requests ▶ Actions 📁 Projects 📖 Wiki ⓘ Security 📈 Insights ⚙ Settings

🔗 Branch: master - roboticapratca / public_html / scripts / Go to file Add file +

betopinheiro1005 committed 1 minute ago [History](#)

...		
📄 fixaMenu.js	Criado a pasta scripts e o arquivo fixaMenu.js	1 minute ago

Sincronizando as mudanças feitas no Github com o repositório local

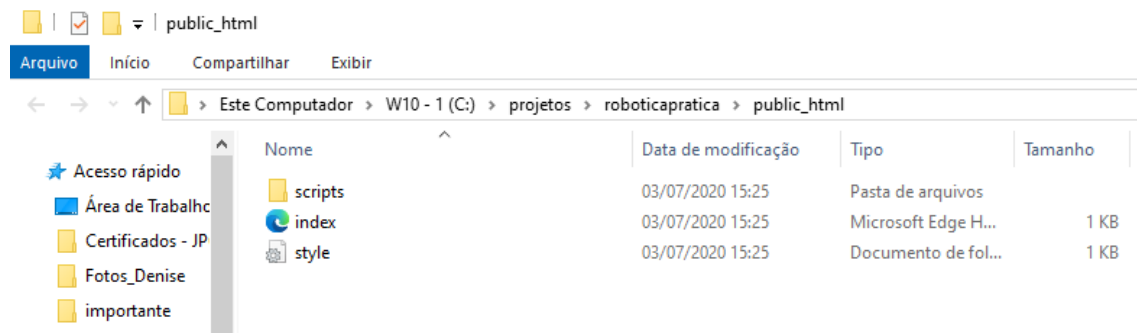
No terminal **bash** (do Git), entre com o seguinte comando:

git pull

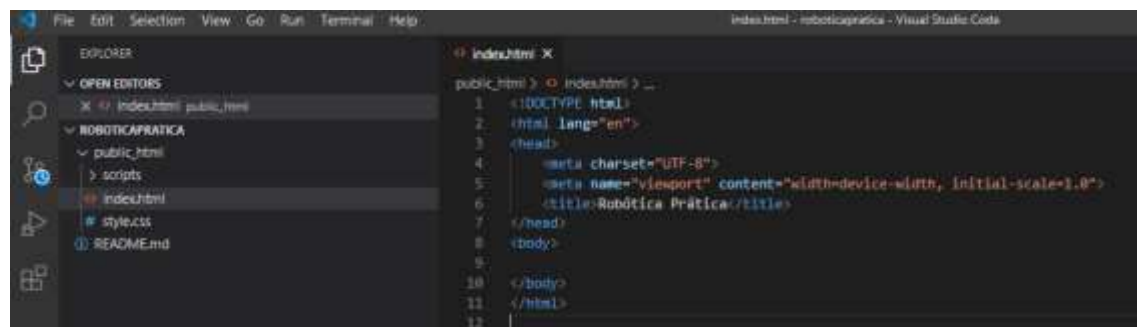
```

beto1@DESKTOP-RR0R61J MINGW64 /c/projetos/roboticapratica/public_html (master)
$ git pull
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (12/12), 2.29 KiB | 4.00 KiB/s, done.
From https://github.com/betopinheiro1005/roboticapratica
   079cc11..cf0e65e  master    -> origin/master
Updating 079cc11..cf0e65e
Fast-forward
 public_html/index.html      | 4 ++--
 public_html/scripts/fixaMenu.js | 1 +
 public_html/style.css       | 1 +
 3 files changed, 4 insertions(+), 2 deletions(-)
 create mode 100644 public_html/scripts/fixaMenu.js
 create mode 100644 public_html/style.css

```



No Visual Studio Code, o título da página já aparece alterado:



Aula 10 - Trabalhando com vários repositórios locais - Resolvendo conflitos

Vamos ver como resolver problemas de conflitos quando trabalhamos com a sincronização de um repositório remoto com vários repositórios locais.

Como exemplo prático vamos simular alguns conflitos do projeto do site roboticapratica.com.br

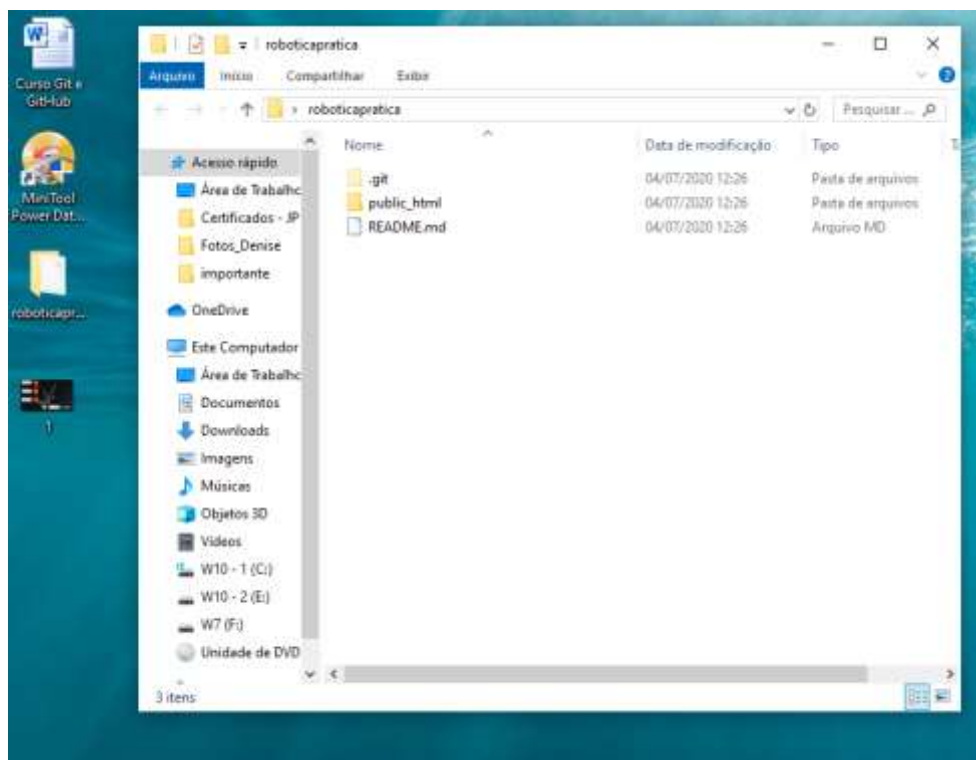
A ideia é trabalhar este mesmo projeto no seu computador pessoal e também no computador de informática da sua faculdade ou do seu escritório.

Clonando o projeto na área de trabalho

Simulando o trabalho em seu PC, clique com o botão direito do mouse na área de trabalho e selecione a opção “**Git Bash Here**”. Ao abrir o terminal, entre com o comando:

```
git clone https://github.com/betopinheiro1005/roboticapratica.git
```

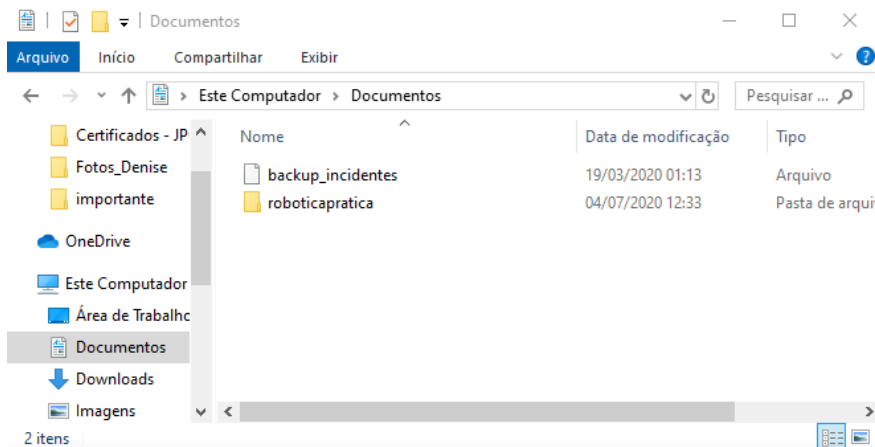
```
betol@DESKTOP-RROR61J MINGW64 ~/Desktop
$ git clone https://github.com/betopinheiro1005/roboticapratica.git
Cloning into 'roboticapratica'...
remote: Enumerating objects: 23, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 23 (delta 1), reused 7 (delta 0), pack-reused 0
Unpacking objects: 100% (23/23), 3.67 KiB | 3.00 KiB/s, done.
```



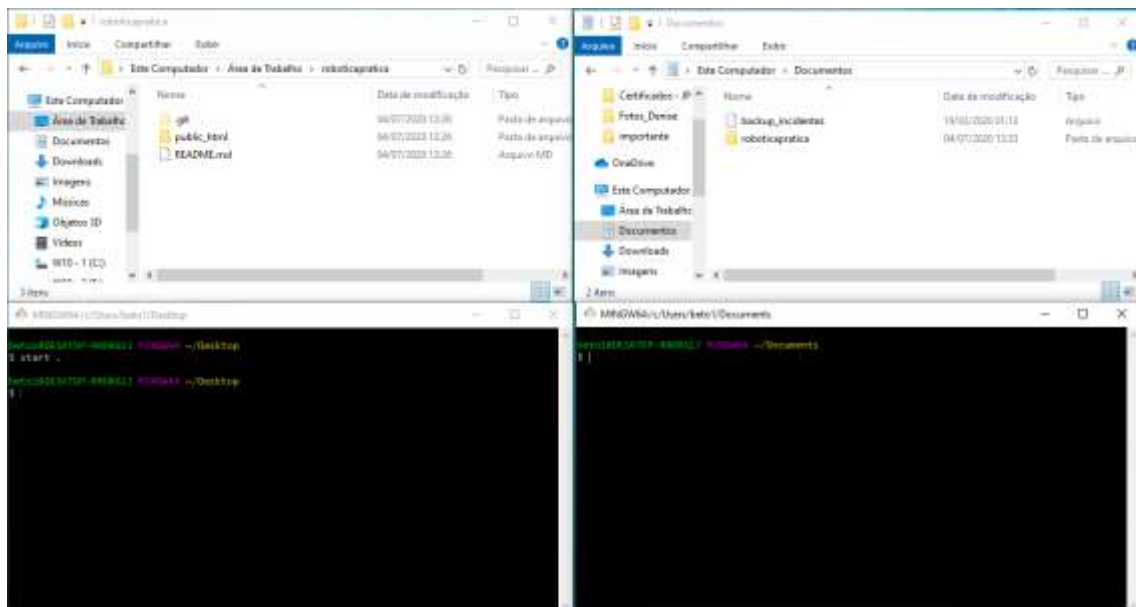
Clonando o projeto na pasta Documents

Simulando o trabalho no laboratório de informática de sua faculdade, repita a operação na pasta “Documents”:

```
beto1@DESKTOP-RROR61J MINGW64 ~/Documents
$ git clone https://github.com/betopinheiro1005/roboticapratica.git
Cloning into 'roboticapratica'...
remote: Enumerating objects: 23, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 23 (delta 1), reused 7 (delta 0), pack-reused 0
Unpacking objects: 100% (23/23), 3.67 KiB | 3.00 KiB/s, done.
```



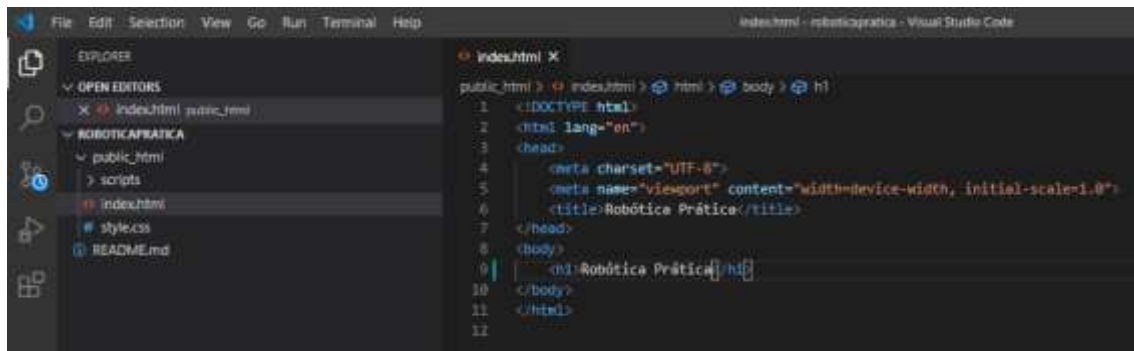
Visualizando os dois ambientes simultaneamente



No Desktop:

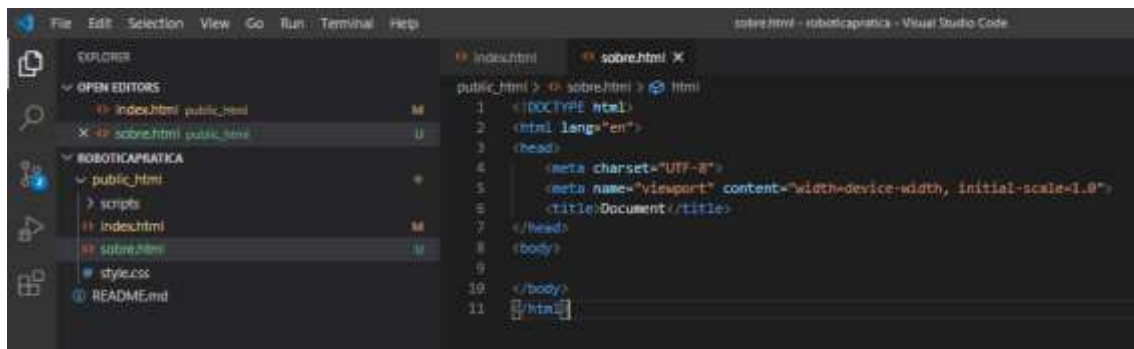
```
beto1@DESKTOP-RROR61J MINGW64 ~/Desktop/roboticapratica
$ code .
```

Alterar o arquivo `index.html`:



Criar o arquivo `sobre.html`:

html:5



```

beto1@DESKTOP-RROR61J MINGW64 ~/Desktop/roboticapratica
$ git status
On branch master
Your branch is up to date with 'origin/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:   public_html/index.html

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

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

beto1@DESKTOP-RROR61J MINGW64 ~/Desktop/roboticapratica
$ git add .

beto1@DESKTOP-RROR61J MINGW64 ~/Desktop/roboticapratica
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   public_html/index.html
    new file:   public_html/sobre.html

beto1@DESKTOP-RROR61J MINGW64 ~/Desktop/roboticapratica
$ git commit -m "Alterado o arquivo index.html e criado o arquivo sobre.html"
[master 58c55b2] Alterado o arquivo index.html e criado o arquivo sobre.html
2 files changed, 12 insertions(+), 1 deletion(-)
create mode 100644 public_html/sobre.html

```

```

beto1@DESKTOP-RR0R61J MINGW64 ~/Desktop/roboticapratica
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

beto1@DESKTOP-RR0R61J MINGW64 ~/Desktop/roboticapratica
$ git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 576 bytes | 115.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/betopinheiro1005/roboticapratica.git
   cf0e65e..58c55b2  master -> master

beto1@DESKTOP-RR0R61J MINGW64 ~/Desktop/roboticapratica
$ |

```

betopinheiro1005 / roboticapratica

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

branch: master + roboticapratica / public_html /

Go to file Add file +

betopinheiro1005 committed 6 minutes ago

History

- scripts Criado o pasta scripts e o arquivo finalVersao.js 4 days ago
- index.html Alterado o arquivo index.html e criado o arquivo sobre.html 4 minutes ago
- sobre.html Alterado o arquivo index.html e criado o arquivo sobre.html 4 minutes ago
- style.css Criado o arquivo style.css 4 days ago

Alterado o arquivo index.html e criado o arquivo sobre.html

betopinheiro1005 committed 6 minutes ago

Showing 2 changed files with 12 additions and 1 deletion.

Unified Split

public_html/index.html

```

@@ -6,8 @@
<title>Robótica Prática</title>
</head>
<body>
+ <div>Robótica Prática</div>
</body>
</html>

```

public_html/sobre.html

```

@@ -0,0 +1,11 @@
+ <DOCTYPE html>
+ <html lang="en">
+ <head>
+   <meta charset="UTF-8">
+   <meta name="viewport" content="width=device-width, initial-scale=1.0">
+   <title>Document</title>
+ </head>
+ <body>
+
+ </body>
+ </html>

```

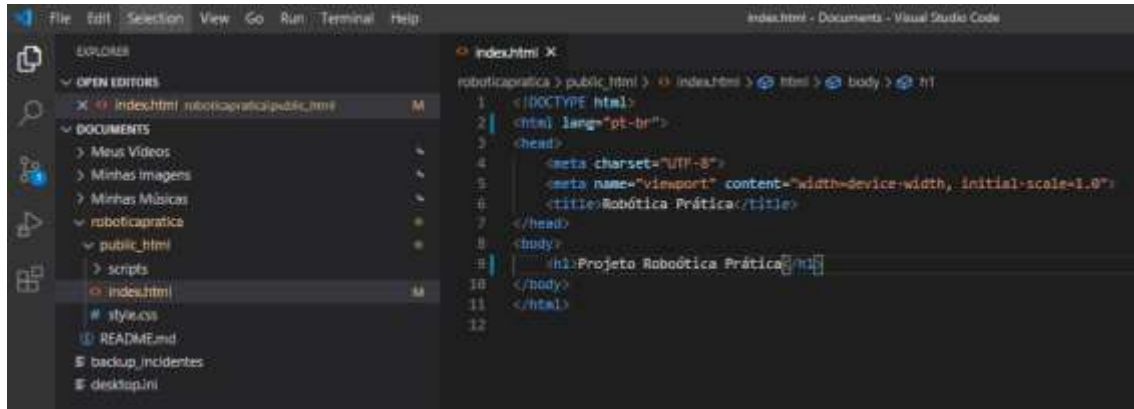
0 comments on commit 58c55b2

Lock conversation

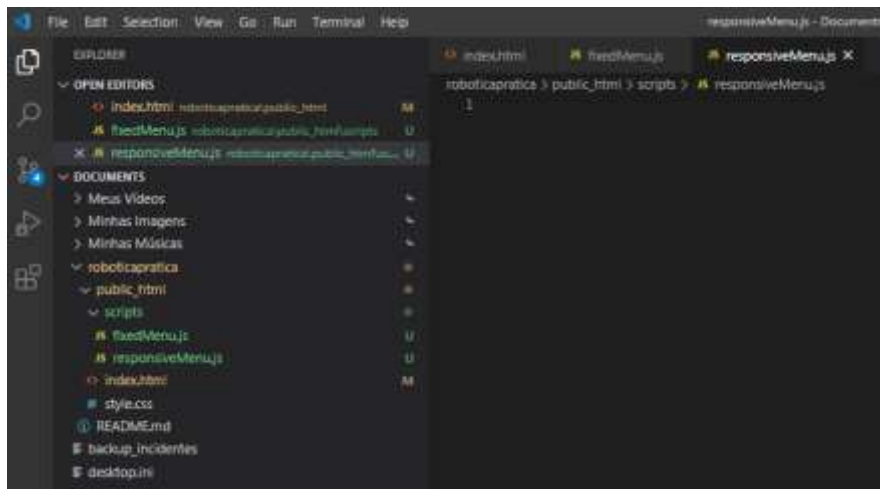
Na pasta “Documents”:

Alterar o arquivo `index.html`:

```
beto1@DESKTOP-RROR61J MINGW64 ~/Documents
$ code .
```



Dentro da subpasta “scripts” apagar o arquivo “fixaMenu.js” e criar o arquivo “fixedMenu.js” e o arquivo “responsiveMenu.js”



```
beto1@DESKTOP-RROR61J MINGW64 ~/Documents/roboticapratica (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   public_html/index.html
    deleted:    public_html/scripts/fixaMenu.js

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    public_html/scripts/fixedMenu.js
    public_html/scripts/responsiveMenu.js

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

```

betol@DESKTOP-RROR61J MINGW64 ~/Documents/roboticapratica (master)
$ git add .
betol@DESKTOP-RROR61J MINGW64 ~/Documents/roboticapratica (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   public_html/index.html
        deleted:    public_html/scripts/fixaMenu.js
        new file:   public_html/scripts/fixaMenu.js
        new file:   public_html/scripts/responsiveMenu.js

betol@DESKTOP-RROR61J MINGW64 ~/Documents/roboticapratica (master)
$ git commit -m "Alterado arquivo index.html - Na pasta scripts apagado arquivo fixaMenu.js e criado os arquivos fixaMenu.js e responsiveMenu.js"
[master e5326c6] Alterado arquivo index.html - Na pasta scripts apagado arquivo fixaMenu.js e criado os arquivos fixaMenu.js e responsiveMenu.js
4 files changed, 2 insertions(+), 3 deletions(-)
delete mode 100644 public_html/scripts/fixaMenu.js
create mode 100644 public_html/scripts/fixaMenu.js
create mode 100644 public_html/scripts/responsiveMenu.js

betol@DESKTOP-RROR61J MINGW64 ~/Documents/roboticapratica (master)
$

```

Sincronizando essas alterações com o repositório remoto

Como há conflitos entre o conteúdo local e o do repositório remoto:

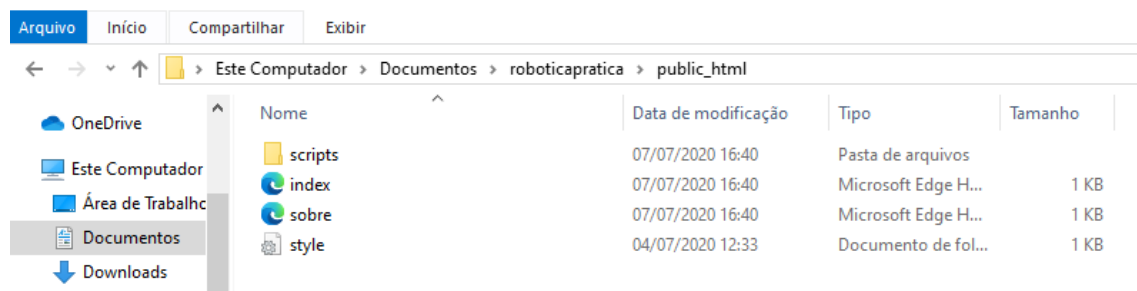
```

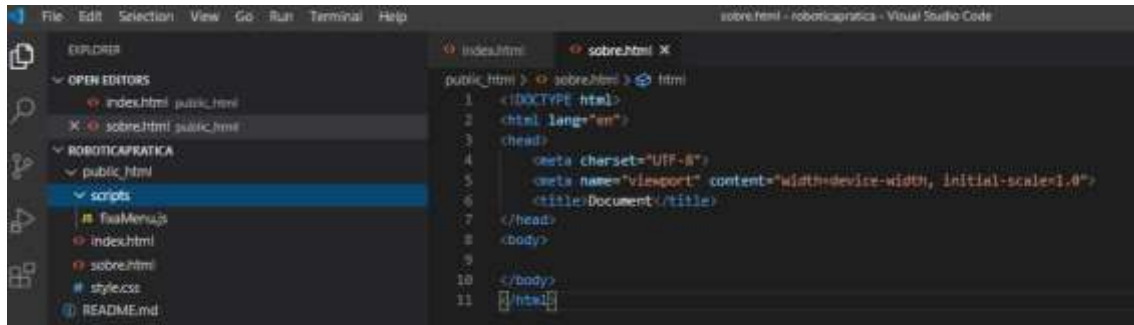
betol@DESKTOP-RROR61J MINGW64 ~/Documents/roboticapratica (master)
$ git push
To https://github.com/betopinheiro1005/roboticapratica.git
! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/betopinheiro1005/roboticapratica.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

```

git fetch

Este comando é usado para fazer o download das alterações que estão no repositório remoto para poder fazer uma análise do que foi modificado e resolver os problemas de conflito. Equivale ao git pull porém ele não faz o merge (fusão dos projetos).





git checkout master

```

betol@DESKTOP-RROR61J MINGW64 ~/Documents/roboticapratica ((58c55b2...))
$ git checkout master
Previous HEAD position was 58c55b2 Alterado o arquivo index.html e criado o arquivo sobre.html
Switched to branch 'master'
Your branch and 'origin/master' have diverged,
and have 1 and 1 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

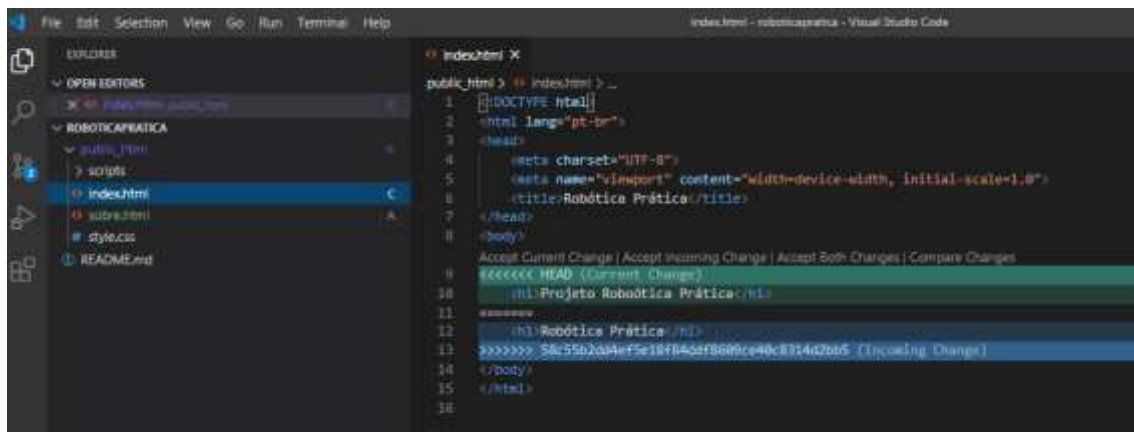
```

git pull

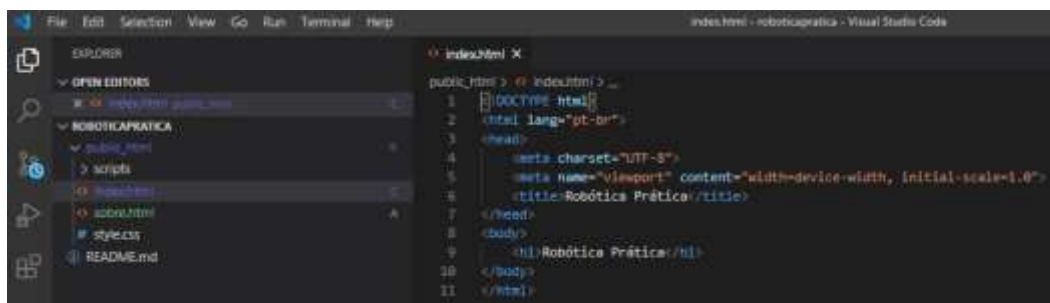
```

betol@DESKTOP-RROR61J MINGW64 ~/Documents/roboticapratica (master)
$ git pull
Auto-merging public_html/index.html
CONFLICT (content): Merge conflict in public_html/index.html
Automatic merge failed; fix conflicts and then commit the result.

```



Clicar em “Accept incoming Change”:



```

beto1@DESKTOP-RR0R61J MINGW64 ~/Documents/roboticapratica (master|MERGING)
$ git status
On branch master
Your branch and 'origin/master' have diverged,
and have 1 and 1 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

You have unmerged paths.
(fix conflicts and run "git commit")
(use "git merge --abort" to abort the merge)

Changes to be committed:
  new file:   public_html/sobre.html

Unmerged paths:
  (use "git add <file>..." to mark resolution)
  both modified: public_html/index.html

```

```

beto1@DESKTOP-RR0R61J MINGW64 ~/Documents/roboticapratica (master|MERGING)
$ git add .

beto1@DESKTOP-RR0R61J MINGW64 ~/Documents/roboticapratica (master|MERGING)
$ git status
On branch master
Your branch and 'origin/master' have diverged,
and have 1 and 1 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

All conflicts fixed but you are still merging.
(use "git commit" to conclude merge)

Changes to be committed:
  modified:   public_html/index.html
  new file:   public_html/sobre.html

```

```

beto1@DESKTOP-RR0R61J MINGW64 ~/Documents/roboticapratica (master|MERGING)
$ git commit -m "alterado index.html e adicionado sobre.html"
[master 95c2f4c] alterado index.html e adicionado sobre.html

beto1@DESKTOP-RR0R61J MINGW64 ~/Documents/roboticapratica (master)
$ git push
Enumerating objects: 18, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 2 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (10/10), 1.06 KiB | 63.00 KiB/s, done.
Total 10 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/betopinheiro1005/roboticapratica.git
  58c55b2..95c2f4c master -> master

```



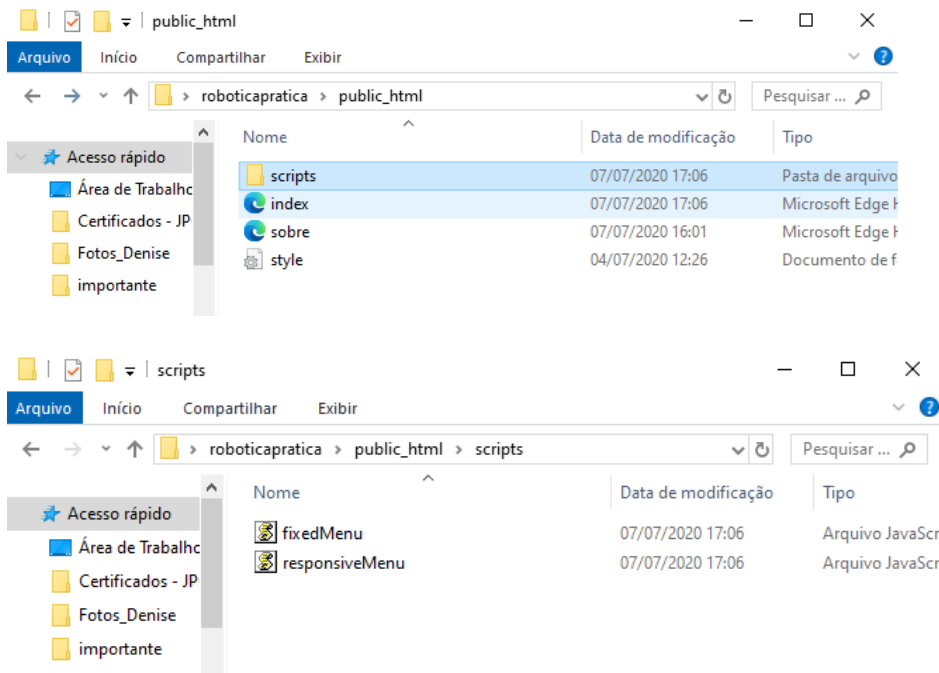
Sincronizando o repositório remoto com o repositório local (Desktop)

git pull

```

beto1@DESKTOP-RROR61J MINGW64 ~/Desktop/roboticapratica/public_html (master)
$ git pull
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 10 (delta 2), reused 10 (delta 2), pack-reused 0
Unpacking objects: 100% (10/10), 1.04 KiB | 3.00 KiB/s, done.
From https://github.com/betopinheiro1005/roboticapratica
   58c55b2..95c2f4c  master    -> origin/master
Updating 58c55b2..95c2f4c
Fast-forward
 public_html/index.html      | 2 +-
 public_html/scripts/fixaMenu.js | 1 -
 public_html/scripts/fixaMenu.js | 0
 public_html/scripts/responsiveMenu.js | 0
4 files changed, 1 insertion(+), 2 deletions(-)
delete mode 100644 public_html/scripts/fixaMenu.js
create mode 100644 public_html/scripts/fixaMenu.js
create mode 100644 public_html/scripts/responsiveMenu.js

```

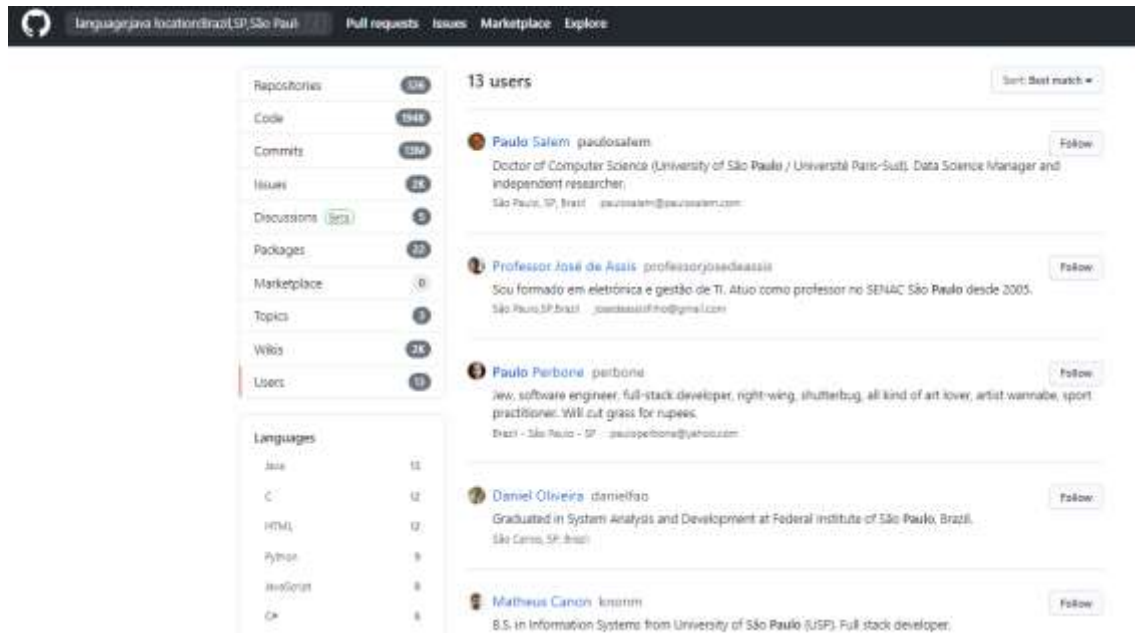


Agora o repositório remoto está sincronizado corretamente com todos os repositórios locais (Desktop e pasta Documents).

Aula 11 - Usando o GITHUB como currículo e portfólio

Github como currículo

Como as empresas encontram os profissionais



The screenshot shows a GitHub repository page for 'languagejava locationindnaoESP,São Paul'. The repository has 13 users. The left sidebar shows repository statistics: 136 Repositories, 1344 Code, 1344 Commits, 2K Issues, 5 Discussions, 22 Packages, 0 Marketplace, 9 Topics, 2K Wikis, and 13 Users. The 'Users' section lists 13 users, including Paulo Salem, Professor José de Assis, Paulo Perbone, Daniel Oliveira, and Matheus Canon.

Repository	Code	Commits	Issues	Discussions	Packages	Marketplace	Topics	Wikis	Users
languagejava locationindnaoESP,São Paul	136	1344	1344	2K	5	22	0	9	2K

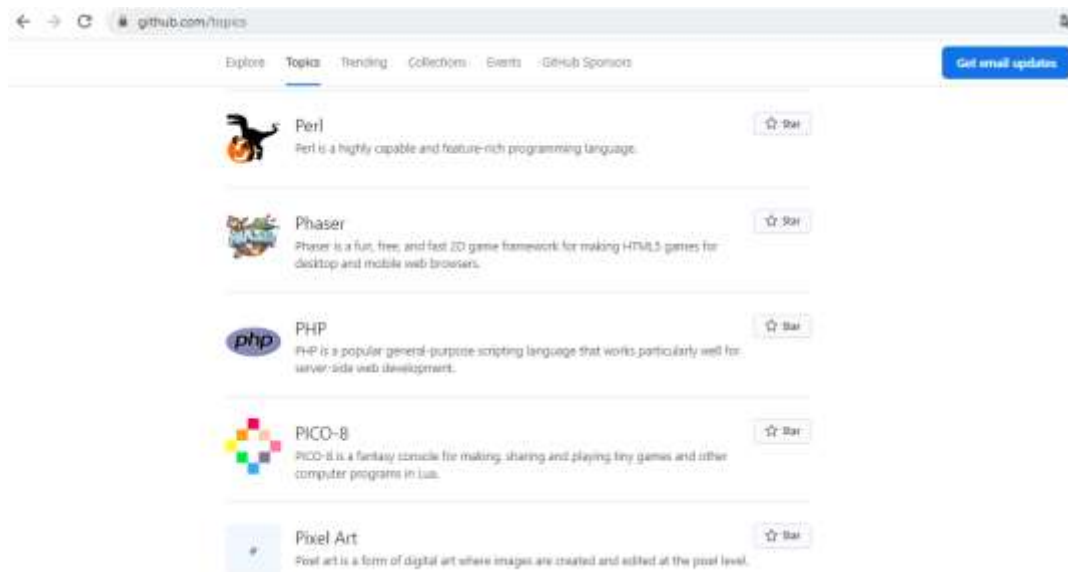
13 users

- Paulo Salem** paulosalem
Doctor of Computer Science (University of São Paulo / Université Paris-Sud), Data Science Manager and independent researcher.
São Paulo, SP, Brazil paulosalem@paulosalem.com
- Professor José de Assis** professorjosedassis
Sou formado em eletrônica e gestão de TI. Atuo como professor no SENAC São Paulo desde 2005.
São Paulo, SP, Brazil joseassisofm@gmail.com
- Paulo Perbone** perbone
Jew, software engineer, full-stack developer, right-wing, shutterbug, all kind of art lover, artist wannabe, sport practitioner. Will cut grass for rupees.
Brazil - São Paulo - SP pauloperbone@yahoo.com
- Daniel Oliveira** danielhao
Graduated in System Analysis and Development at Federal Institute of São Paulo, Brazil.
São Carlos, SP, Brazil
- Matheus Canon** knonim
B.S. in Information Systems from University of São Paulo (USP). Full stack developer.

Pesquisa usando Topics

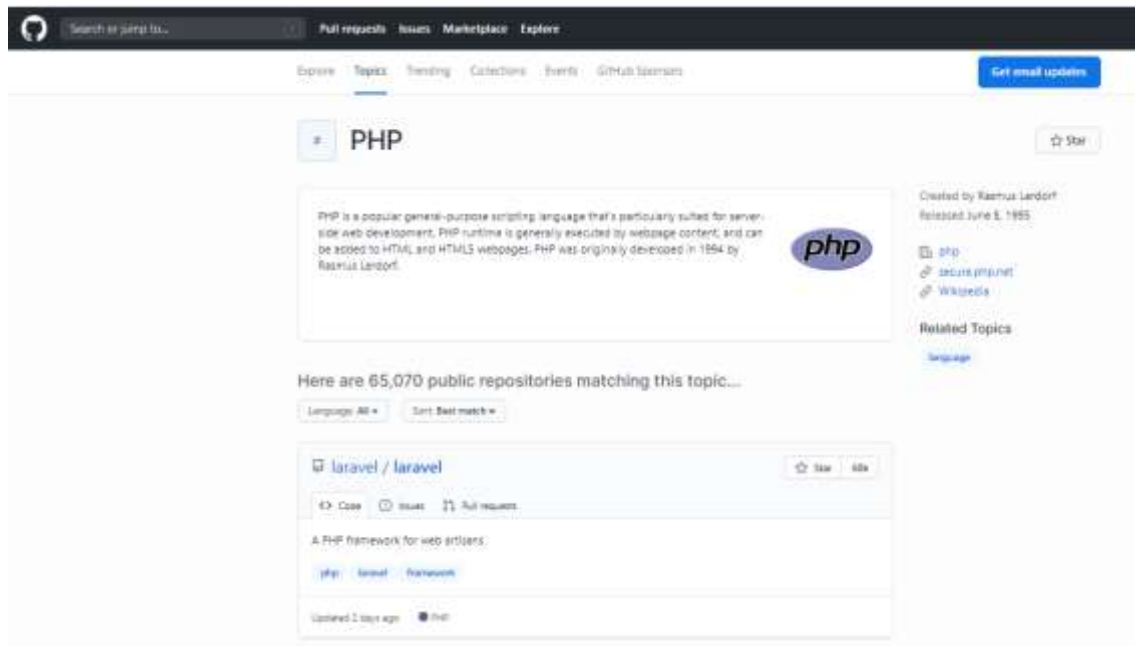
Clique no link “Explore” e posteriormente no link “Topics”.

Ex: Filtrando por PHP



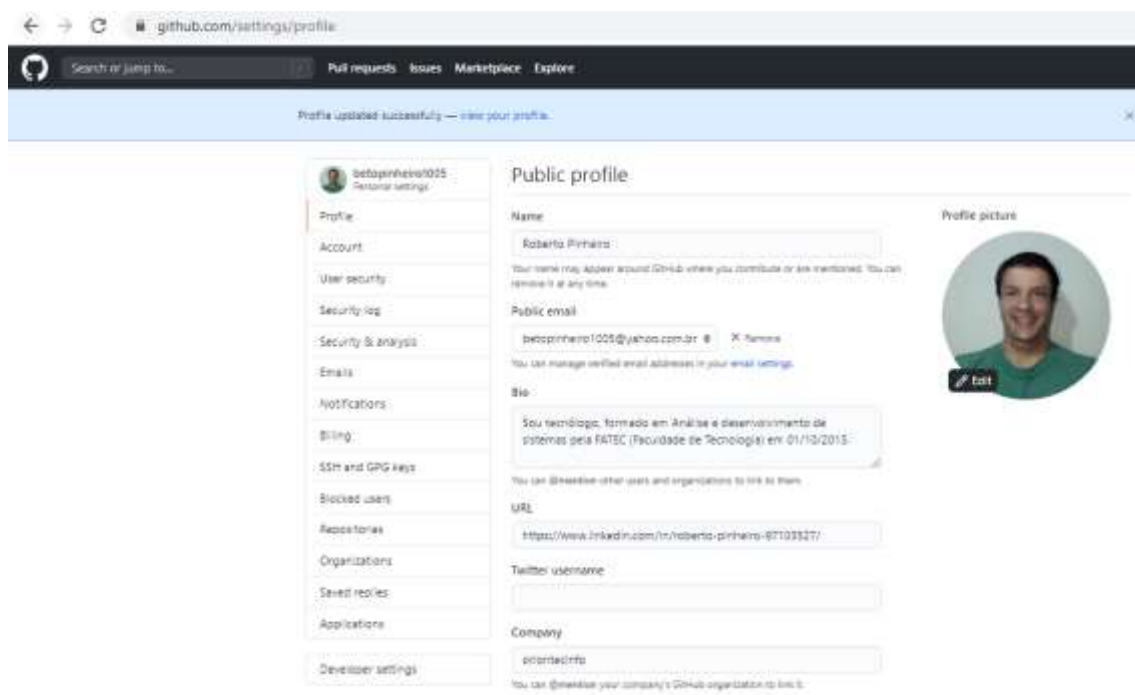
The screenshot shows the GitHub Topics page. The top navigation bar includes links for Explore, Topics, Trending, Collections, Events, and GitHub Sponsors. A 'Get email updates' button is visible. The main content area displays a list of topics, each with a logo, name, description, and a 'Star' button.

Topic	Description
Perl	Perl is a highly capable and feature-rich programming language.
Phaser	Phaser is a fun, free, and fast 2D game framework for making HTML5 games for desktop and mobile web browsers.
PHP	PHP is a popular general-purpose scripting language that works particularly well for server-side web development.
PICO-8	PICO-8 is a fantasy console for making, sharing and playing tiny games and other computer programs in Lua.
Pixel Art	Pixel art is a form of digital art where images are created and edited at the pixel level.



Configurando o seu perfil

Clique em “Settings”.



Location

São Paulo, SP, Brazil

All of the fields on this page are optional and can be deleted at any time, and by filling them out you're giving us consent to share this data wherever your user profile appears. Please see our [privacy statement](#) to learn more about how we use this information.

Update profile

Contributions

☐ Include private contributions on my profile

Get credit for all your work by showing the number of contributions to private repositories on your profile without any repository or organization information. [Learn how we count contributions.](#)

Update contributions

GitHub Developer Program

Building an application, service, or tool that integrates with GitHub? [Join the GitHub Developer Program](#), or read more about it at our [Developer site](#).

Check out the [Developer site](#) for guides, our API reference, and other resources for building applications that integrate with GitHub. Make sure your contact information is up-to-date below. Thanks for being a member!

Jobs profile

☒ Available for hire

Save jobs profile

Jobs profile

☒ Available for hire

Save jobs profile

Trending settings

Preferred spoken language

No Preference

We'll use this language preference to filter the trending repository lists on [Explore](#) and our [Trending Repositories](#) page.

Save Trending settings

GitHub Jobs


[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)


Create a new repository

A repository contains all project files, including the revision history. [Already have a project repository elsewhere? Import a repository.](#)

Owner *

Repository name *


 betopinheiro1005 ▾


/ c 

Great repository names are short and memorable. Need inspiration? How about [effective-octo-spork?](#)

Description (optional)

Primeiros Passos com a Linguagem C


☒  **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.

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

☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾

Add a license: MIT license ▾ 

Create repository

The screenshot shows a GitHub repository page for the user 'betopinheiro1005' and the repository 'c'. The browser address bar shows 'github.com/betopinheiro1005/c'. The repository page includes a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, there are buttons for 'Go to file', 'Add file', and '+ code'. The repository is described as 'Primeros Passos com a Linguagem C'. The file list shows 'LICENSE' and 'README.md', both with 'Initial commit' and 'now' timestamps. The 'README.md' file is expanded, showing the title 'C' and the subtitle 'Primeros Passos com a Linguagem C'. The right sidebar contains sections for 'About', 'Releases', and 'Packages', all indicating no content has been published yet.

github.com/betopinheiro1005/c

betopinheiro1005 / c

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

branch: master

Go to file Add file + code

About

Primeros Passos com a Linguagem C

1 commit 1 branch 0 tags

LICENSE Initial commit now

README.md Initial commit now

README.md

C

Primeros Passos com a Linguagem C

Releases

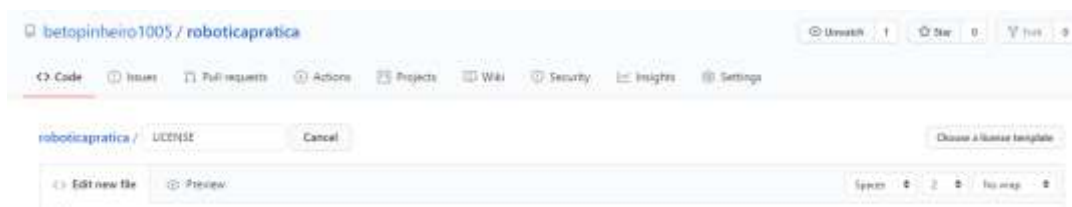
No releases published
Create a new release

Packages

No packages published
Publish your first package

Alterando o tipo de licença de um repositório

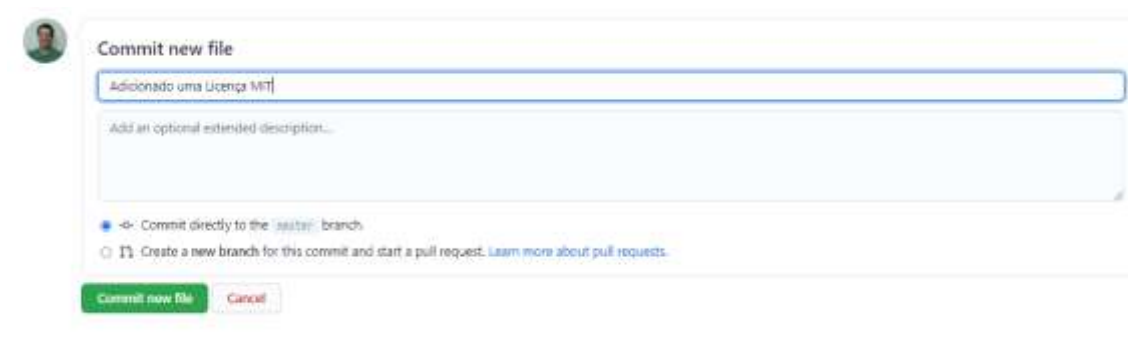
- Abra o seu repositório.
- Clique em “Add file” e em seguida em “Create new file”.
- Crie um arquivo com o nome de **LICENSE**:

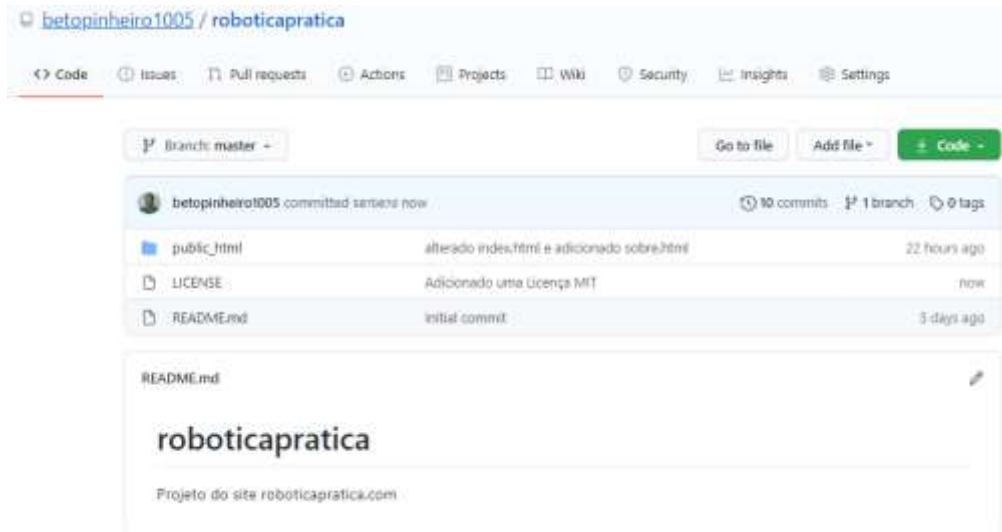


- Clique no botão “Choose a license template” e selecione a licença.
- Clique no botão “Review and Submit”.



- Faça o commit. Escolha o ramo **Master**:





Adicionando um emoji à descrição

Edit repository details

Description

:computer: Primeiros Passos com a Linguagem C

Website

<https://www.youtube.com/watch?v=COgylca8qYw>

Topics (separate with spaces)

c x linguagem-c x language-c x

Include in the home page

☒ Releases

☒ Packages

☒ Environments

Cancel Save changes

About



Primeiros Passos com a Linguagem C

www.youtube.com/watch?v=cogylca8qYw

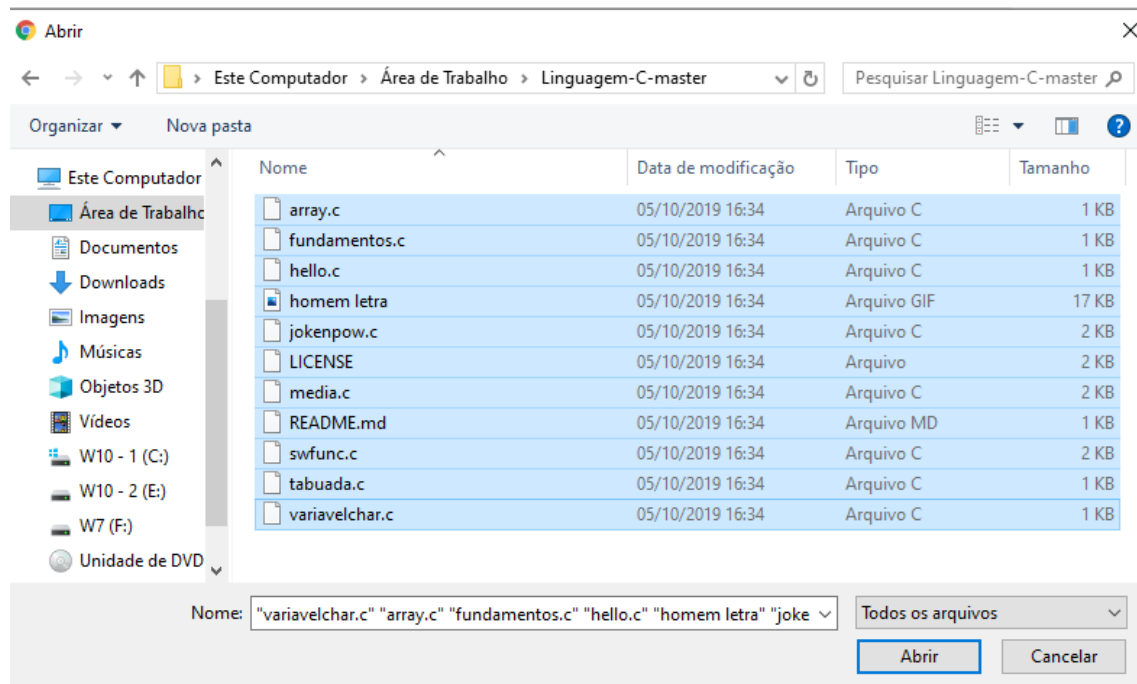
c linguagem-c language-c

Readme

MIT License

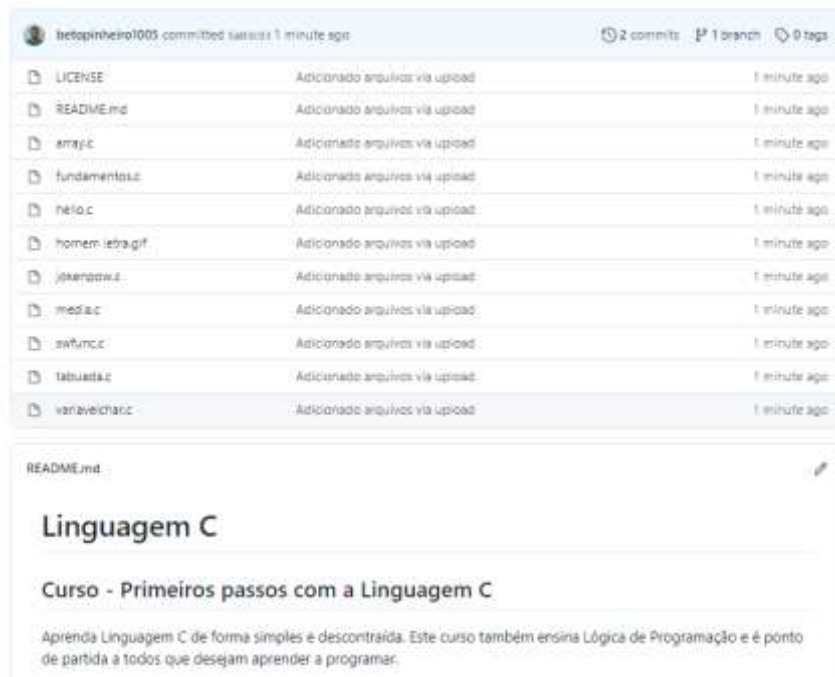
Fazendo o upload de arquivos do projeto

- Clique em “Add files” e posteriormente em “Upload files”



- Em “Commit changes” entre com:

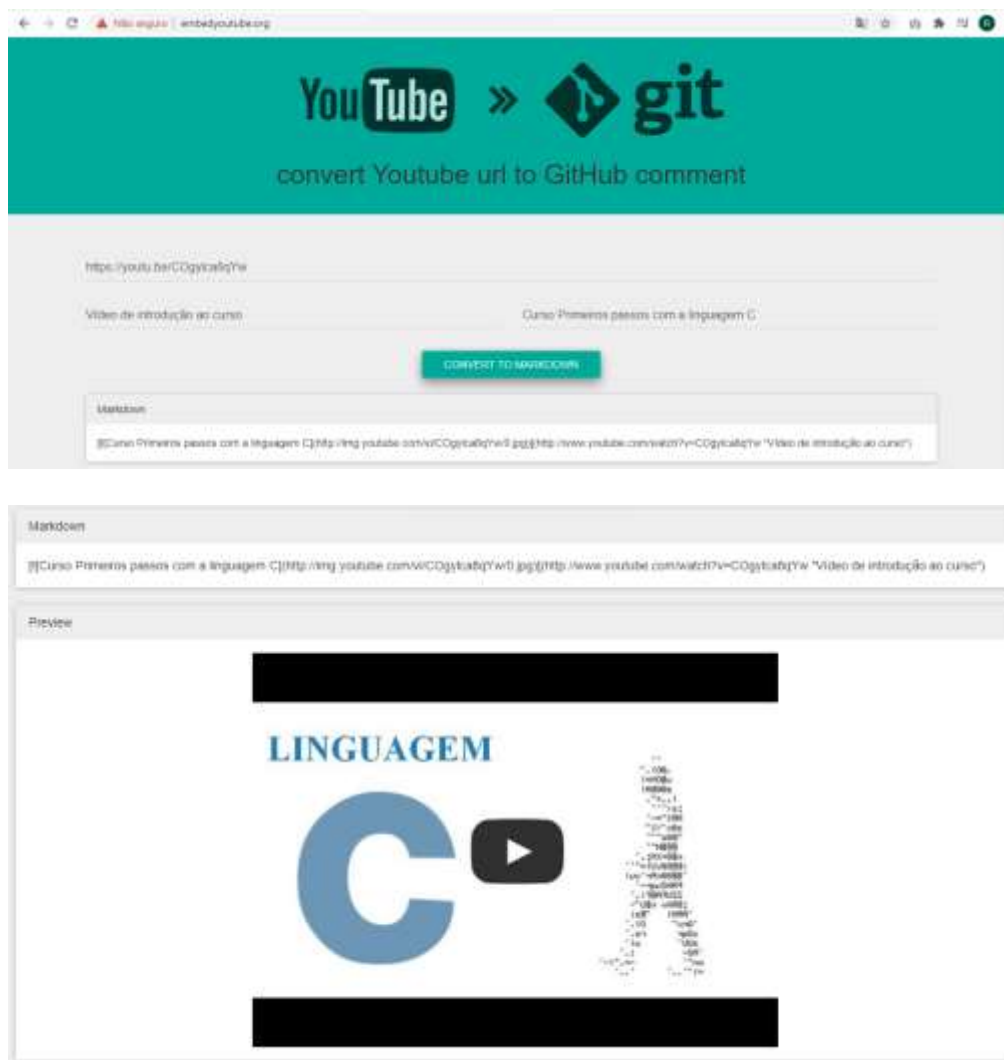
Adicionado arquivos via upload.



O próximo passo é personalizar o arquivo **README.md**

Aula 12 - Personalizando o README.md

<http://embedyoutube.org/>



Linguagem C

Curso - Primeiros passos com a Linguagem C

Aprenda Linguagem C de forma simples e descontraída. Este curso também ensina Lógica de Programação e é ponto de partida a todos que desejam aprender a programar.

![Homem letra](https://github.com/professorjosededeassis/c/blob/master/homem%20letra.gif)

Assistir ao vídeo de introdução

[[Curso primeiros passos com a Linguagem

C](http://img.youtube.com/vi/COgylca8qYw/0.jpg)](http://www.youtube.com/watch?v=COgylca8qYw "Vídeo de introdução ao curso")

Matricule-se grátis:

[aulaead](https://www.aulaead.com/courses/curso-gratis-linguagem-c)

gist.github.com/professorjosedeeass/d1c08e6667f43be18a1f25c40e1ac2de

Give the example

And repeat

until finished

End with an example of getting some data out of the system or using it for a little demo

Running the tests

Explain how to run the automated tests for this system

Break down into end to end tests

Explain what these tests test and why

Give an example

And coding style tests

Explain what these tests test and why

Give an example

gist.github.com/professorjosedeeass/d1c08e6667f43be18a1f25c40e1ac2de

Deployment

Add additional notes about how to deploy this on a live system

Built With

- [Dropwizard](#) - The web framework used
- [Maven](#) - Dependency Management
- [ROME](#) - Used to generate RSS feeds

Contributing

Please read [CONTRIBUTING.md](#) for details on our code of conduct, and the process for submitting pull requests to us.

Versioning

We use [SemVer](#) for versioning. For the versions available, see the [tags](#) on this repository.

Authors

- [Billie Thompson](#) - Initial work - [PurpleBooth](#)

See also the list of [contributors](#) who participated in this project.

gist.github.com/professorjosedeeass/d1c08e6667f43be18a1f25c40e1ac2de

License

This project is licensed under the MIT License - see the [LICENSE.md](#) file for details

Acknowledgments

- Hat tip to anyone whose code was used
- Inspiration
- etc

Project Title

One Paragraph of project description goes here

Getting Started

These instructions will get you a copy of the project up and running on your local machine for development and testing purposes. See deployment for notes on how to deploy the project on a live system.

Prerequisites

What things you need to install the software and how to install them

...

Give examples

...

Installing

A step by step series of examples that tell you how to get a development env running

Say what the step will be

...

Give the example

...

And repeat

...

until finished

...

End with an example of getting some data out of the system or using it for a little demo

Running the tests

Explain how to run the automated tests for this system

Break down into end to end tests

Explain what these tests test and why

...

Give an example

...

And coding style tests

Explain what these tests test and why

...

Give an example

...

Deployment

Add additional notes about how to deploy this on a live system

Built With

- * [Dropwizard](http://www.dropwizard.io/1.0.2/docs/) - The web framework used
- * [Maven](https://maven.apache.org/) - Dependency Management
- * [ROME](https://rometools.github.io/rome/) - Used to generate RSS Feeds

Contributing

Please read [CONTRIBUTING.md](https://gist.github.com/PurpleBooth/b24679402957c63ec426) for details on our code of conduct, and the process for submitting pull requests to us.

Versioning

We use [SemVer](http://semver.org/) for versioning. For the versions available, see the [tags on this repository](https://github.com/your/project/tags).

Authors

* **Billie Thompson** - *Initial work* - [PurpleBooth](https://github.com/PurpleBooth)

See also the list of [contributors](https://github.com/your/project/contributors) who participated in this project.

License

This project is licensed under the MIT License - see the LICENSE.md file for details

Acknowledgments

- * Hat tip to anyone whose code was used
- * Inspiration
- * etc

Aula 13 - Gerenciando projetos com a ferramenta KANBAN nativa do GitHub

O **Kanban** é um sistema visual de gerenciamento de projetos.



Criando um projeto de Kanban

Clique no link “**Projects**” e posteriormente no botão “**Create a new project**”.

Create a new project
Coordinate, track, and update your work in one place, so projects stay transparent and on schedule.

Project board name
Planejamento do Projeto do site robotica praticas

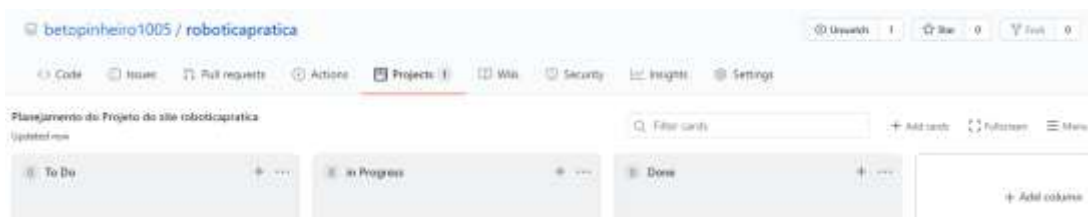
Description (optional)

Project template
Save yourself time with a pre-configured project board template.
Template: None ▾

Create project

- Clique no botão “**Create project**”.

- Crie três colunas: To Do, in Progress e Done.



Criando os cartões

betopinheiro1005 / roboticapratica

<> Code Issues Pull requests Actions Projects 1 Wiki Security Insights Settings

Planejamento do Projeto do site roboticapratica
Updated 10 minutes ago

2 To Do

- Identidade visual
 - ☐ logotipo
 - ☐ estilo
 - ☐ cores
 - ☐ fontes
- Briefing

Added by betopinheiro1005

1 in Progress

- Criar um novo projeto

Added by betopinheiro1005

0 Done

Edit note

Note

Identidade visual

- [] logotipo
- [] estilo
- [] cores
- [] fontes

Save note

betopinheiro1005 / roboticapratica

<> Code Issues Pull requests Actions Projects 1 Wiki Security Insights Settings

Planejamento do Projeto do site roboticapratica
Updated 32 minutes ago

9 To Do

Enter a note

Add Cancel

- Briefing
- Identidade visual
 - ☐ logotipo
 - ☐ estilo
 - ☐ cores
 - ☐ fontes

Added by betopinheiro1005

1 in Progress

Enter a note

Add Cancel

- Preparação do ambiente de desenvolvimento
 - ☒ Git e GitHub
 - ☐ Ferramentas
 - ☐ Editor de código fonte
 - ☐ Validação
 - ☐ Google Console Search
 - ☐ Compactadores
 - ☐ Hospedagem
 - ☐ Domínio
 - ☐ Servidor Web
 - ☒ GitHub Pages

1 Done

- Criar um novo projeto

Added by betopinheiro1005

Aula 14 - Fork e Pull Request

O **fork** faz a clonagem de um repositório diretamente no seu GitHub. Ao fazer um fork você pode fazer qualquer tipo de alteração no projeto clonado sem interferir no projeto original.

Caso queira sugerir alguma alteração no projeto original você pode fazer um “**pull request**” sinalizando as alterações ao dono do projeto. Quando o dono do projeto receber o pull request ele pode aceitar ou não as mudanças. O pull request é uma forma de você contribuir com projetos da comunidade.

Aula 15 - Selos Dinâmicos de metadados usando shield.io

Instalando selos de metadados no README.md

<https://shields.io/>

Existem muitos tipos de selos, subdivididos em categorias. Estes selos usam as informações de seus repositórios. Alguns destes selos também dependem de serviços externos para poderem funcionar.

The screenshot shows the Shields.io website with the URL `shields.io/category/build` in the browser address bar. The page features a navigation bar with links like `Build`, `Code Coverage`, `Analysis`, `Test`, `Dependencies`, `Size`, `Downloads`, `Package`, `Issue Tracking`, and `License`. Below the navigation bar, there is a section titled `Build` which displays a list of badges and their corresponding GitHub repository URLs.

Badge	Repository URL
AppVeyor	<code>/wooviyot/build/visualstudio</code>
AppVeyor branch	<code>/wooviyot/build/visualstudio/branch</code>
AppVeyor Job	<code>/wooviyot/job/build/visualstudio/job</code>
AppVeyor Job branch	<code>/wooviyot/job/build/visualstudio/job/branch</code>
AppVeyor tests	<code>/wooviyot/tests/visualstudio</code>
AppVeyor tests (branch)	<code>/wooviyot/tests/visualstudio/branch</code>
AppVeyor tests (compact)	<code>/wooviyot/tests/visualstudio/compact_message</code>
AppVeyor tests with custom labels	<code>/wooviyot/tests/visualstudio/expanded_label=buildstatus_label=modified_label=MSVS</code>
Azure DevOps builds	<code>/azure-devops/build/organization/projectId/definitionId</code>
Azure DevOps builds (branch)	<code>/azure-devops/build/organization/projectId/definitionId/branch</code>
Azure DevOps builds	<code>/azure-devops/build/organization/projectId/definitionId:branch*/?tag=SuccessfulBuild</code>

Categoria Size

Build	Code Coverage	Analysis	Chat	Dependencies	Size	Downloads	Function	Issue Tracking	License
Baking	Social	Version	Platform & Version Support	Monitoring	Activity	Other			

	Size	
npm bundle size:	verified size: 8.65 kB	/bundlename/:format/:packageName
npm bundle size (scoped):	verified size: 3.56 kB	/bundlename/:format/:scope/:packageName
npm bundle size (version):	verified size: 30.5 kB	/bundlename/:format/:packageName/:version
npm bundle size (scoped version):	verified size: 3.56 kB	/bundlename/:format/:scope/:packageName/:version
Docker Image Size (latest by date):	image size: 128 MB	/docker/image-size/:user/:repo/latest-date
Docker Image Size (latest semver):	image size: 136 MB	/docker/image-size/:user/:repo/semver
Docker Image Size (tag):	image size: 103 MB	/docker/image-size/:user/:repo/tag
GitHub code size in bytes:	code size: 1.43 MB	/github/languages/code-size/:user/:repo
GitHub repo size:	repo size: 127 MB	/github/repo-size/:user/:repo
GitHub file size in bytes:	size: 8.17 kB	/github/size/:user/:repo/:path*
MicroBadger Layers:	layers: 15	/microbadger/layers/:user/:repo
MicroBadger Layers (tag):	layers: 13	/microbadger/layers/:user/:repo/:tag
MicroBadger Size:	image size: 196 MB	/microbadger/image-size/:user/:repo
MicroBadger Size (tag):	image size: 103 MB	/microbadger/image-size/:user/:repo/:tag
Spiget Download Size:	size: 3.5 MB	/spiget/download-size/:resourceId
Steam File Size:	size: 20 MB	/steam/size/:fileId
Visual Studio App Center Size:	size: 8.37 MB	/visual-studio-app-center/releases/size/:owner/:app/:token

GitHub repo size

GitHub repo size

If your GitHub badge errors, it might be because you hit GitHub's rate limits. You can increase Shields.io's rate limit by [adding the Shields GitHub application](#) using your GitHub account.

user

repo

e.g. atom

atom

style

flat

override label

optional

override color

optional

named logo

optional

override logo color

optional

repo size: 3.95 MB

Copy Badge URL

Selecione a opção “Copy Markdown”.



- Faça o **commit**.

Commit changes

Inserido pelo "size"

Add an optional extended description...

- ☒ Commit directly to the `master` branch.
- ☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes

Cancel

Categoria Analysis

GitHub language count

GitHub language count

If your GitHub badge errors, it might be because you hit GitHub's rate limits. You can increase Shields.io's rate limit by [adding the Shields GitHub application](#) using your GitHub account.

The screenshot shows the 'GitHub language count' badge generator. At the top, there are input fields for 'user' (betopinheiro1005) and 'repo' (crud-incidentes). Below these are examples: 'e.g. badges' and 'shields'. A 'style' dropdown menu is set to 'flat'. There are four optional override fields: 'override label', 'override color', 'named logo', and 'override logo color'. Below these fields is a preview of the badge showing 'languages 4'. At the bottom is a blue button labeled 'Copy Badge URL' with a dropdown arrow.

Selecione a opção “Copy Markdown”.

GitHub top language

GitHub top language

If your GitHub badge errors, it might be because you hit GitHub's rate limits. You can increase Shields.io's rate limit by [adding the Shields GitHub application](#) using your GitHub account.

The screenshot shows the 'GitHub top language' badge generator. At the top, there are input fields for 'user' (betopinheiro1005) and 'repo' (crud-incidentes). Below these are examples: 'e.g. badges' and 'shields'. A 'style' dropdown menu is set to 'flat'. There are four optional override fields: 'override label', 'override color', 'named logo', and 'override logo color'. Below these fields is a preview of the badge showing 'php 72.4%'. At the bottom is a blue button labeled 'Copy Badge URL' with a dropdown arrow.

- Selecione a opção “Copy Markdown”.

crud-incidentes / readme.md Cancel

< Edit file Preview changes

```

1  :[GitHub_repo_size](https://img.shields.io/github/repo-size/betopinheiro1885/crud-incidentes)
2  :[GitHub_language_count](https://img.shields.io/github/languages/count/betopinheiro1885/crud-incidentes)
3  :[GitHub_top_language](https://img.shields.io/github/languages/top/betopinheiro1885/crud-incidentes)
4
5  # CRUD DE INCIDENTES
6
7  ## Sobre
8  |
9  É um CRUD básico que permite criar, ler, atualizar e excluir registros de incidentes.
10
11 ## Características
12
13 - Sistema desenvolvido com PHP (versão 7.2.11) orientado a objetos
14 - Utiliza o framework Laravel versão 5.6
15 - Arquitetura da aplicação: padrão MVC (Model-View-Controller).
16 - Utiliza o pacote Laravel Collective versão 5.6.0 para criação de formulários.
17 - Design visual: Utiliza o framework front-end Bootstrap. É responsivo.
18 - Ícones: utiliza o framework Font Awesome Icons

```

- Faça o commit.

Commit changes

Inserindo selos da Categoria "Analysis"

Add an optional extended description...

☒ Commit directly to the `master` branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes Cancel

repo size 3.95 MB languages 4 php 72.4%

CRUD DE INCIDENTES

Sobre

É um CRUD básico que permite criar, ler, atualizar e excluir registros de incidentes.

Características

- Sistema desenvolvido com PHP (versão 7.2.11) orientado a objetos
- Utiliza o framework Laravel versão 5.6
- Arquitetura da aplicação: padrão MVC (Model-View-Controller).
- Utiliza o pacote Laravel Collective versão 5.6.0 para criação de formulários
- Design visual: Utiliza o framework front-end Bootstrap. É responsivo.
- Ícones: utiliza o framework Font Awesome Icons

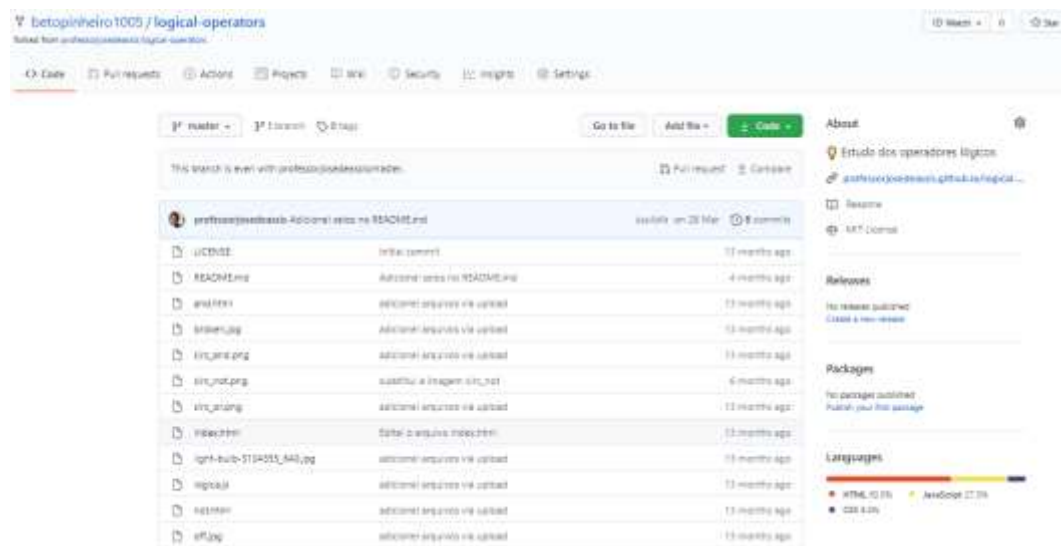
Aula 16 - Hospedagem de site grátis no GITHUB

GitHub Pages

O **GitHub Pages** permite a hospedagem de um site estático, ou seja: com arquivos HTML, CSS, Javascript sem suporte ao PHP, sem um banco de dados, sem a possibilidade de instalar, por exemplo, o WordPress.

Limite de no máximo 1 Gb para seu site e um limite de largura de banda de 100G/mês. Não permite gerar nenhum tipo de comércio eletrônico.

- No github do Prof. José de Assis, faça o fork do projeto logical-operators.



- Clique em **Settings**:
- Em **Source**, selecione a opção **“master branch”**.

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is ready to be published at <https://betopinheiro1005.github.io/logical-operators/>.

Source
 Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)
 master branch ▼

Theme Chooser
 Select a theme to publish your site with a Jekyll theme. [Learn more.](#)
 Choose a theme

Custom domain
 Custom domains allow you to serve your site from a domain other than betopinheiro1005.github.io. [Learn more.](#)
 Save

Endereço URL gerado:

<https://betopinheiro1005.github.io/logical-operators/>



<https://www.credential.net/credential-redirect/20499081#gs.9znsft>

<https://www.linkedin.com/pulse/certificado-curso-git-e-github-roberto-pinheiro>