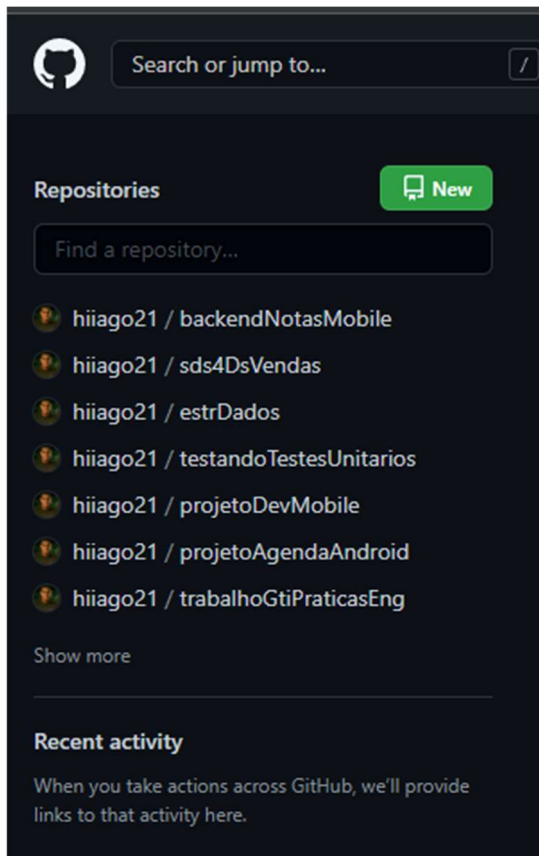
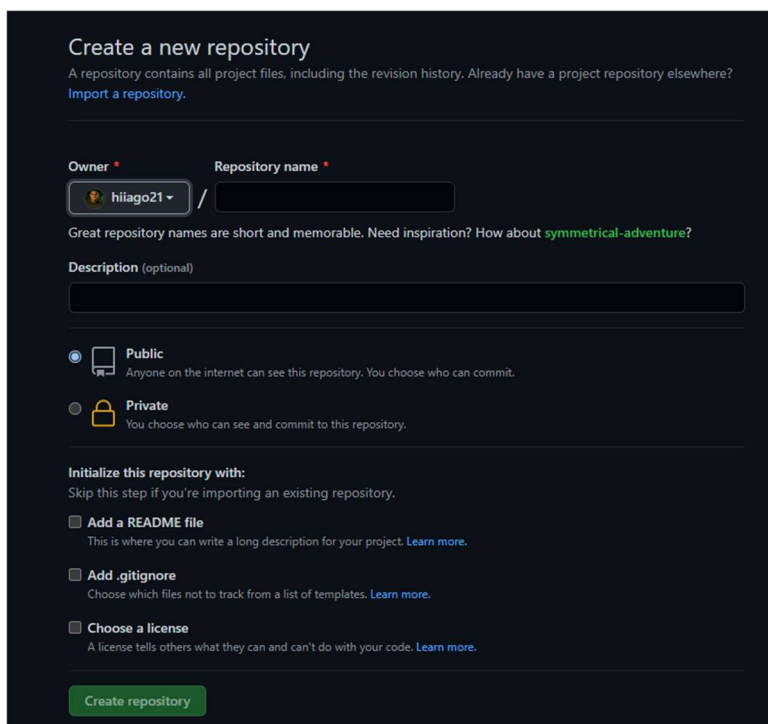


Criando uma branch remotamente, usando github:

Clique no new na sua página do github:



Escolha um nome para seu repositório:

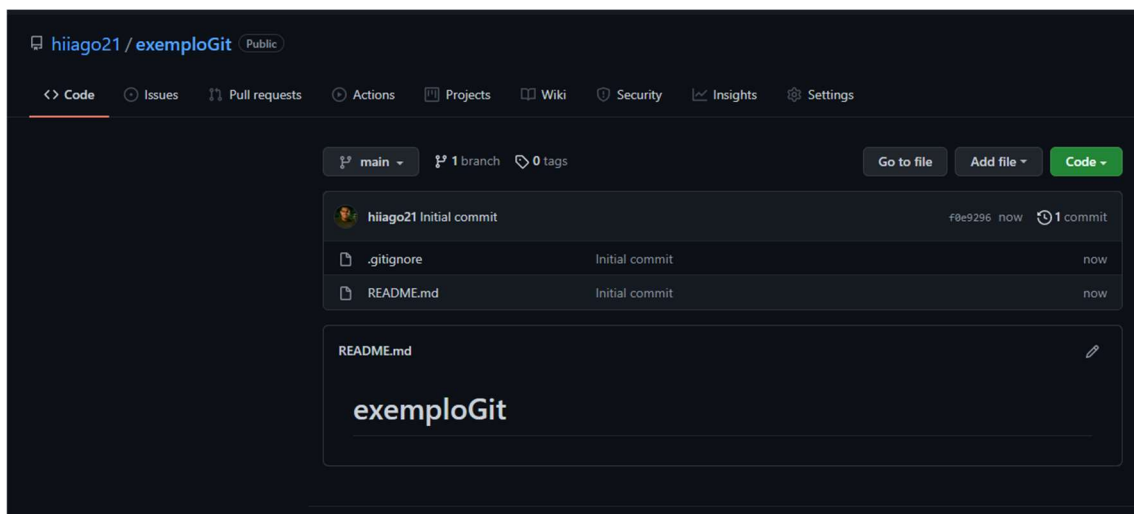


Obs: você já pode incluir um README file, que é um arquivo que vai passar para que acessar seu repositório como instalar, configurar e rodar seu código. Também pode incluir um gitignore, este arquivo serve para seu git bash instalado na máquina saber o que deve e não subir para o repositório remoto.

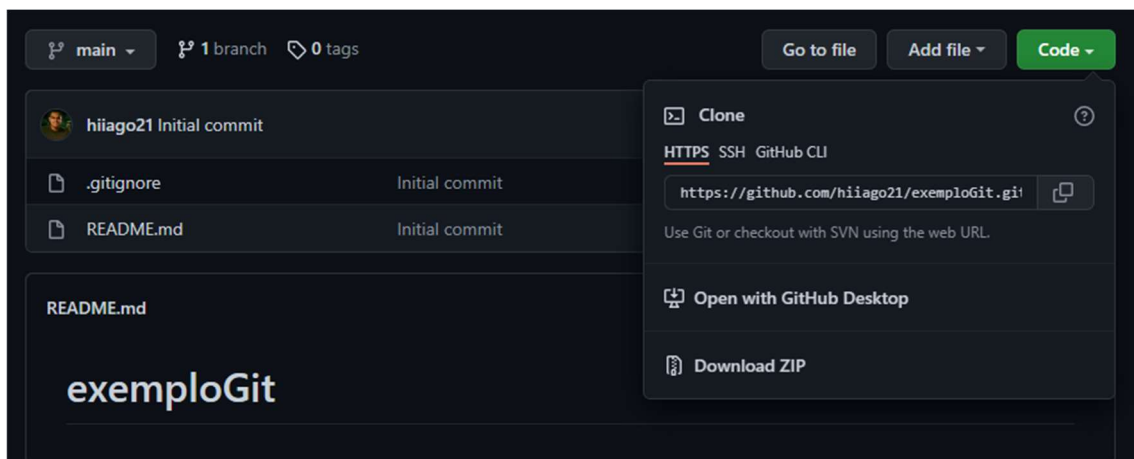
GitBash : <https://git-scm.com/about>

Gitbash é um terminal que deve ser instalado para que os comando do git sejam reconhecidos na máquina local.

Após o create teremos esta tela:

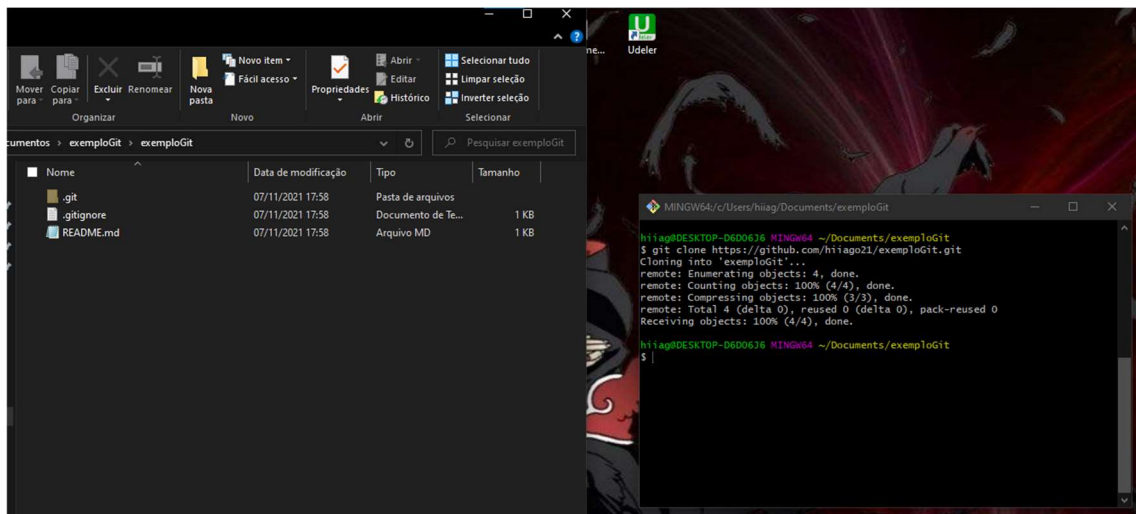


Clicando em code teremos esta tela :



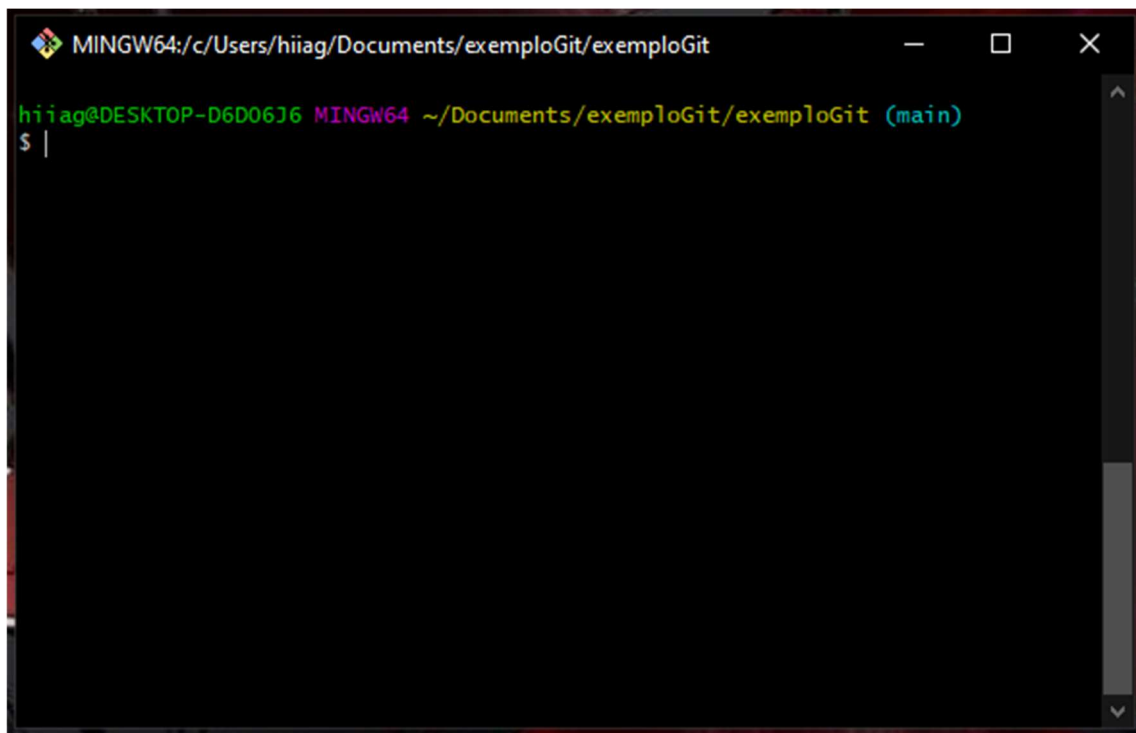
Aqui teremos o link para fazer o clone deste repositório remoto para máquina local.

Clonando um repositório remoto



Com o link copiado da web utilize ele para clonar o repositório.

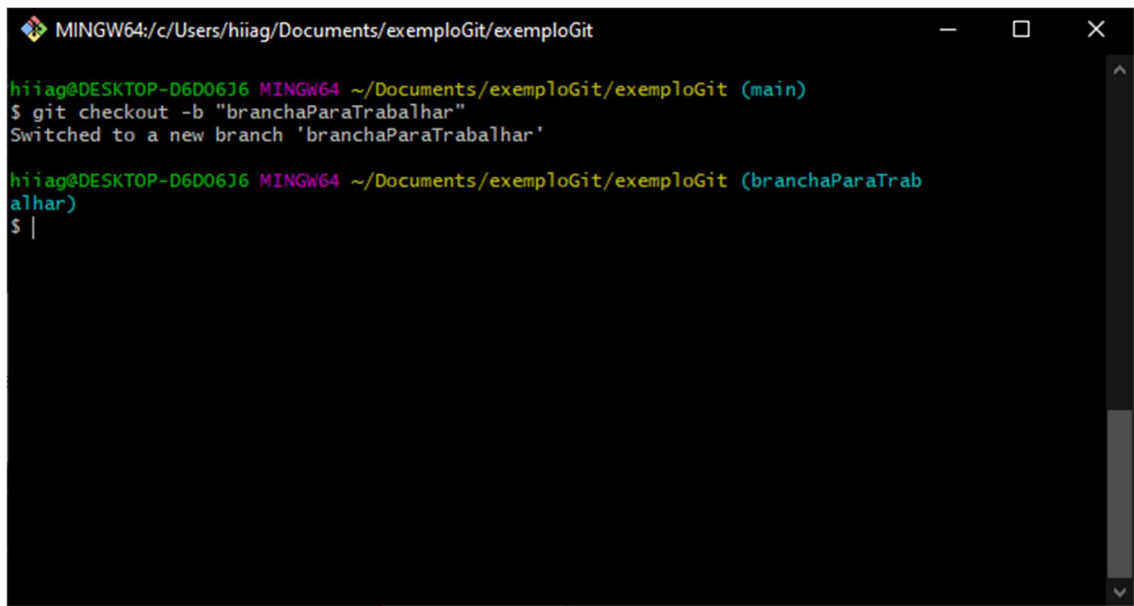
Acessando a pasta que tem o nome do repositório:



Podemos verificar que agora temos o nome da branch, main, após o endereço local do arquivo.

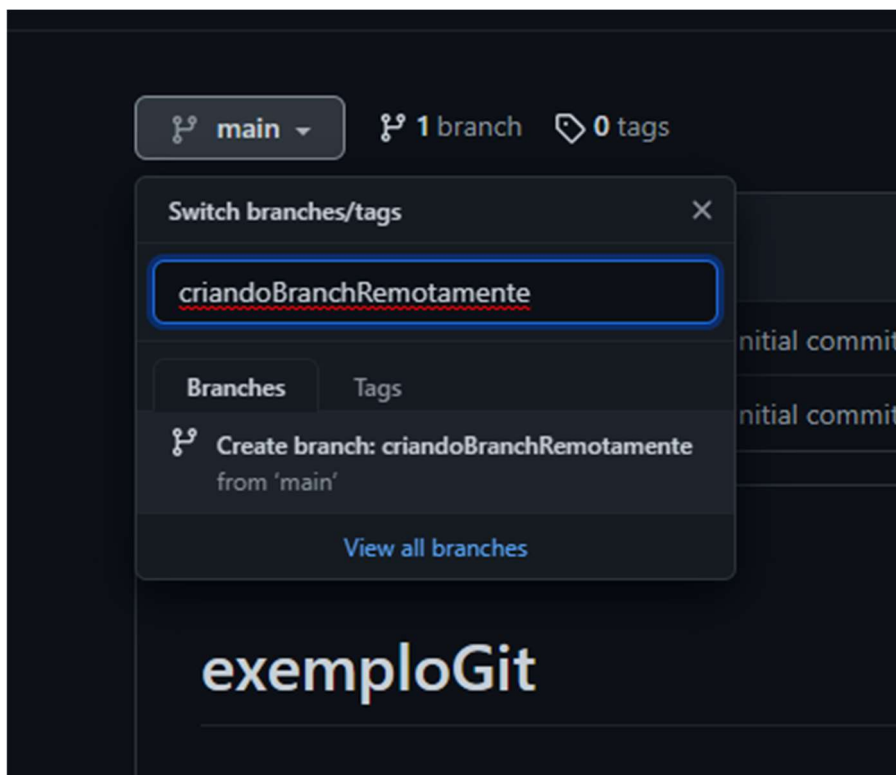
Podemos fazer a criação de uma nova branch localmente para evitar trabalhar na branch principal.

git checkout -b “nome da branch”

A terminal window with a dark background. The title bar shows the path 'MINGW64:/c/Users/hiiag/Documents/exemploGit/exemploGit'. The prompt is 'hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit/exemploGit (main)'. The command '\$ git checkout -b "branchaParaTrabalhar"' is entered. The output is 'Switched to a new branch 'branchaParaTrabalhar''. The prompt changes to 'hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit/exemploGit (branchaParaTrabalhar)' and a new line '\$ |' is shown.

```
MINGW64:/c/Users/hiiag/Documents/exemploGit/exemploGit  
hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit/exemploGit (main)  
$ git checkout -b "branchaParaTrabalhar"  
Switched to a new branch 'branchaParaTrabalhar'  
hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit/exemploGit (branchaParaTrabalhar)  
$ |
```

Podemos também criar remotamente um branch e trazer ela para a máquina local



E após criar teremos o remoto diferente do local então se faz necessário um comando para puxar as alterações remotas para o local

git pull é o comando para trazer as alterações remotas

```
MINGW64/c/Users/hiiag/Documents/exemploGit/exemploGit

hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit/exemploGit (branchaParaTrab
alhar)
$

hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit/exemploGit (branchaParaTrabalhar)
$ git pull
From https://github.com/hiiago21/exemploGit
* [new branch]      criandoBranchRemotamente -> origin/criandoBranchRemotamente
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

    git pull <remote> <branch>

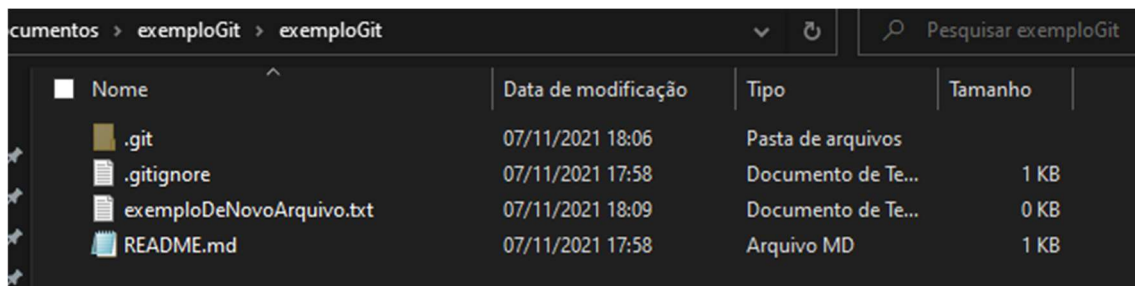
If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=origin/<branch> branchaParaTrabalhar
```

Podemos ver agora as branch disponíveis, usando o comando **git branch -a**

```
hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit/exemploGit (branchaParaTrabalhar)
$ git branch -a
* branchaParaTrabalhar
main
remotes/origin/HEAD -> origin/main
remotes/origin/criandoBranchRemotamente
remotes/origin/main
```

Agora fizemos alterações locais e gostaríamos de guardar ela no repositório remotamente para que possamos acessá-las de outros locais ou para compartilhar elas.



Nome	Data de modificação	Tipo	Tamanho
.git	07/11/2021 18:06	Pasta de arquivos	
.gitignore	07/11/2021 17:58	Documento de Te...	1 KB
exemploDeNovoArquivo.txt	07/11/2021 18:09	Documento de Te...	0 KB
README.md	07/11/2021 17:58	Arquivo MD	1 KB

Com o comando **git status** podemos verificar quais arquivos foram alterados, criados ou excluídos localmente:

```
MINGW64/c/Users/hiiag/Documents/exemploGit/exemploGit

hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit/exemploGit (branchaParaTrabalhar)
$ git status
On branch branchaParaTrabalhar
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    exemploDeNovoArquivo.txt

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

hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit/exemploGit (branchaParaTrabalhar)
$
```

Usando o comando **git add .** adicionamos todos os arquivos locais para o status **staged**, arquivos neste estados irão subir para o repositório remoto.

```
MINGW64:/c/Users/hiiag/Documents/exemploGit/exemploGit
hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit/exemploGit (branchaParaTrabalhar)
$ git add .

hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit/exemploGit (branchaParaTrabalhar)
$ git status
On branch branchaParaTrabalhar
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   exemploDeNovoArquivo.txt

hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit/exemploGit (branchaParaTrabalhar)
$ |
```

Uma verificação visual que podemos ter é a cor do path do arquivos que irá mudar quando mudar seu status.

Então agora faremos o commit destas alterações, nesta etapa iremos colocar uma mensagem que defina as alterações que fizemos:

git commit -m "mensagem do commit"

```
hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit/exemploGit (branchaParaTrabalhar)
$ git commit -m "subindo as alteracoes"
[branchaParaTrabalhar c8ac20c] subindo as alteracoes
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 exemploDeNovoArquivo.txt
```

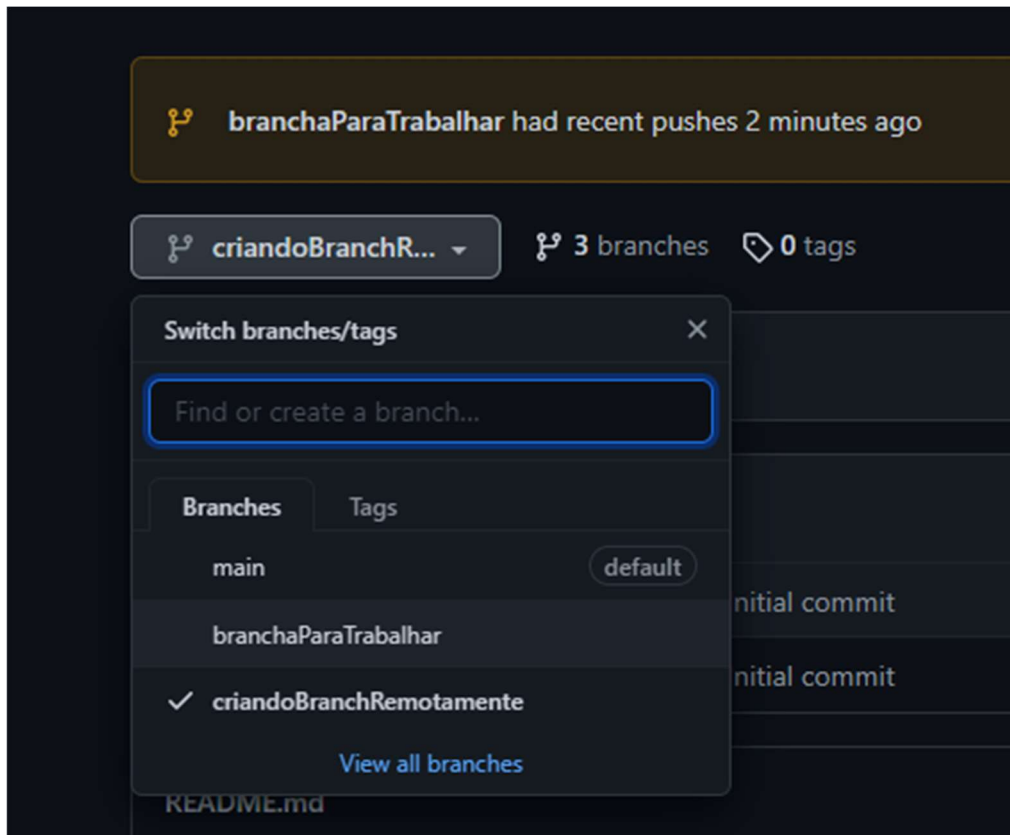
Posterior a isto faremos o **git push** este serve para subir os arquivos para repositório remoto

```
hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit/exemploGit (branchaParaTrabalhar)
$ git push --set-upstream origin branchaParaTrabalhar
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 329 bytes | 65.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'branchaParaTrabalhar' on GitHub by visiting:
remote:   https://github.com/hiiago21/exemploGit/pull/new/branchaParaTrabalhar
remote:
To https://github.com/hiiago21/exemploGit.git
 * [new branch]      branchaParaTrabalhar -> branchaParaTrabalhar
Branch 'branchaParaTrabalhar' set up to track remote branch 'branchaParaTrabalhar' from 'origin'
.
```

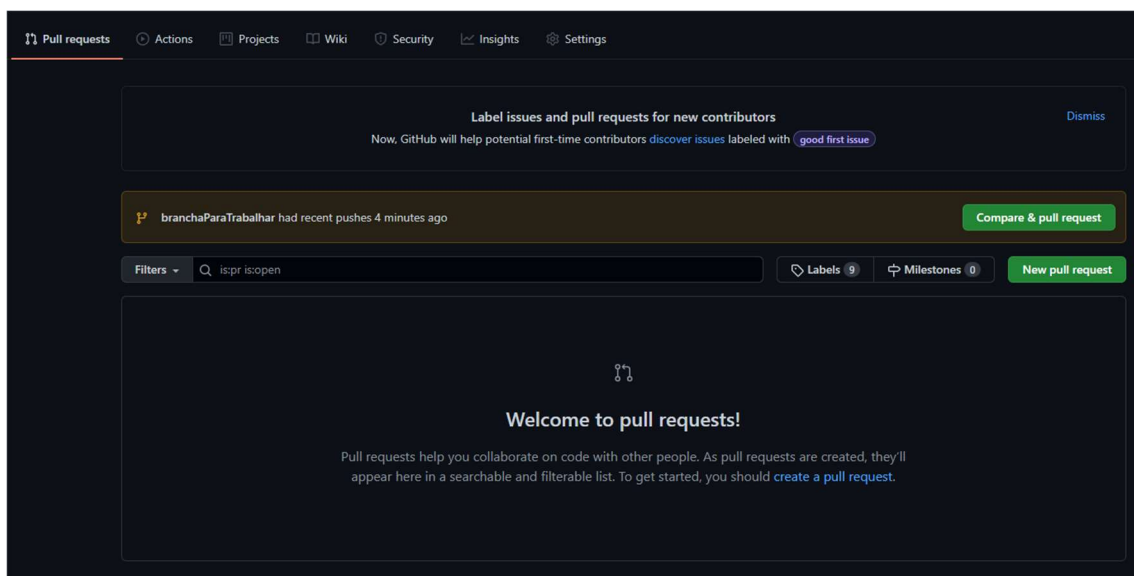
Repare que na primeira execução o comando não é apenas **git push** e sim: **git push --set-upstream origin branchaParaTrabalhar**

Como é o primeiro commit de um branch local que ainda não estava nas branch do repositório remoto isto é necessário para o apontamento correto.

Agora analisando as branch's remotas podemos ver que são 3



Então levaremos as alterações da branch “**branchaParaTrabalhar**” para a “**main**”, na aba **pull request**,




Iremos fazer um new pull request

Open a pull request


Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: main ← compare: `branchaParaTrabalhar` ✓ **Able to merge.** These branches can be automatically merged.



Write Preview H B I ≡ <> 🔗 ≡ ≡ ☑ @ ↻ ↶

Leave a comment


Attach files by dragging & dropping, selecting or pasting them. 

Create pull request ▾



Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Aqui colocaremos um titulo e um comentário para deixar claro o pull request, então depois de clicar em create pull request, virá a tela para efetuar um merge, que é unificar os dados da branch que foi usada para criar alterações na branch principal.


subindo as alteracoes #1

Open hiiago21 wants to merge 1 commit into `main` from `branchaParaTrabalhar` 


Conversation 0 Commits 1 Checks 0 Files changed 1


 **hiiago21** commented now Owner 

No description provided.


↔  subindo as alteracoes ✓ c8ac28c

Add more commits by pushing to the `branchaParaTrabalhar` branch on `hiiago21/exemploGit`.


**✓ All checks have passed** Show all checks
1 successful check


**✓ This branch has no conflicts with the base branch**
Merging can be performed automatically.

Merge pull request ▾ You can also open this in [GitHub Desktop](#) or view [command line instructions](#).

 Write Preview H B I ≡ <> 🔗 ≡ ≡ ☑ @ ↻ ↶

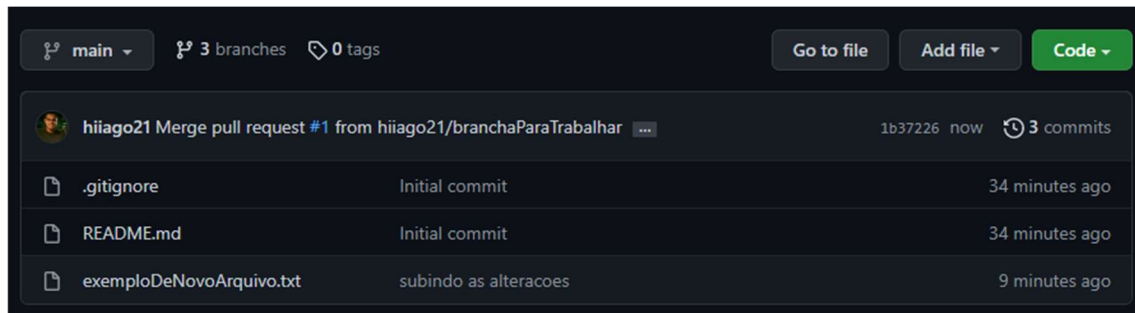
Leave a comment

Attach files by dragging & dropping, selecting or pasting them. 

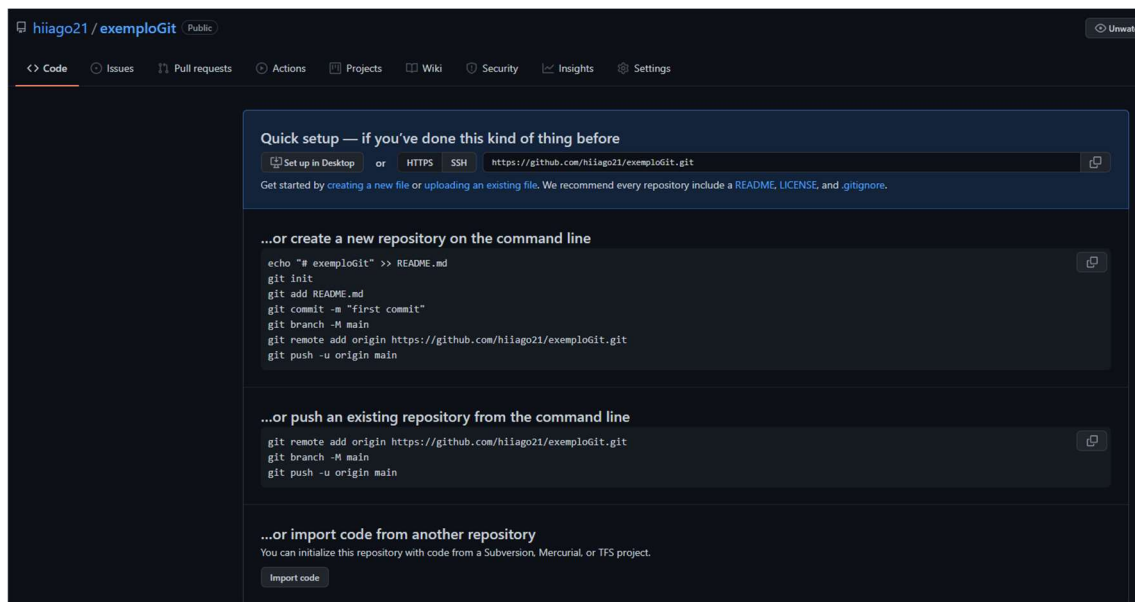
 **Close pull request** **Comment**

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

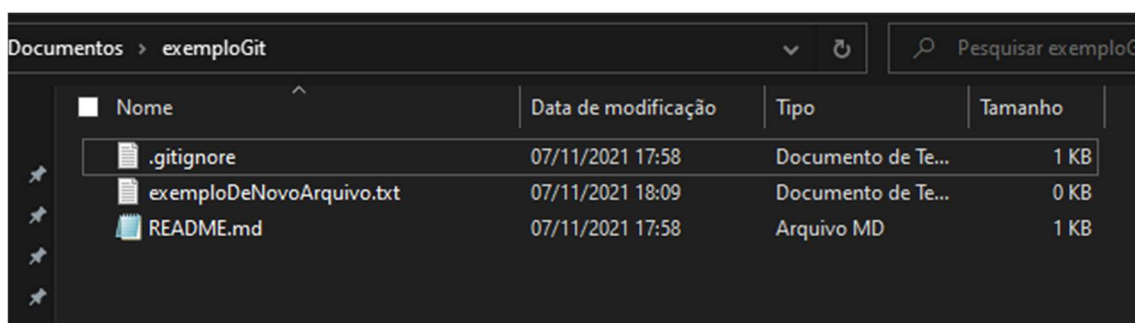
Olhando a branch “main”, agora podemos ver o arquivo que estava na branch local



Uma outra abordagem para subir as alterações que é possível seria, após concluir as alterações localmente no fim subir as alterações para um repositório, se cria um repositório vazio.



Podemos verificar que na pasta agora com os arquivos finais não temos uma pasta oculta **.git**, que guarda as informações do repositório remoto.



Então usamos o comando **git init**, para informar que aquela pasta será rastreada pelo git

```
MINGW64: c:/Users/hiiag/Documents/exemploGit
hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit
$ git init
Initialized empty Git repository in C:/Users/hiiag/Documents/exemploGit/.git/
```

Após isto apontaremos para o repositório remoto com o comando

git remote add origin <https://github.com/hiiago21/exemploGit.git>

que é o endereço web do repositório, então agora teremos o local conectado com remoto.

```
hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit (main)
$ git remote add origin https://github.com/hiiago21/exemploGit.git
```

Agora novamente temos o nome da branch ao lado do path da pasta local.

Então agora seguiremos o fluxo de git status, git add ., git commit "mensagem" e por fim git push para subir as alterações.

```
MINGW64: c:/Users/hiiag/Documents/exemploGit
hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit (main)
$ git status
On branch main
No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit (main)
$ git add .
hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit (main)
$ git status
On branch main
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   README.md
        new file:   exemploDeNovoArquivo.txt
hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit (main)
$ git commit -m "subindo alterações"
[main (root-commit) 83664aa] subindo alterações
3 files changed, 24 insertions(+)
create mode 100644 .gitignore
create mode 100644 README.md
create mode 100644 exemploDeNovoArquivo.txt
hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit (main)
$ git push
fatal: The current branch main has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin main
hiiag@DESKTOP-D6D06J6 MINGW64 ~/Documents/exemploGit (main)
$ git push --set-upstream origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 536 bytes | 536.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/hiiago21/exemploGit.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Segue uma lista de outros comandos que são muito úteis para o dia a dia.

Comandos de Git

CRIAÇÃO

Copiar um repositório

git clone

<https://repositoriox.git>

Criar um novo repositório local

git init

MUDANÇAS LOCAIS

Arquivos alterados em seu diretório de trabalho

git status

Mudanças em arquivos rastreados

git diff

Adicione todas as mudanças atuais para o próximo commit

git add .

Adicione algumas mudanças em <file> para o próximo commit

git add . -p <file>

Confirmar todas as alterações locais em arquivos rastreados

git commit -a

Confirmar alterações previamente preparadas

git commit

Mude o último commit

git commit --amend

HISTÓRICO

Mostrar todos os commits, começando pelo mais novo

git log

Mostrar mudanças ao longo do tempo para um arquivo específico

git log -p <file>

Mostra quem alterou o quê e quando em <file>

git blame <file>

BRANCHES

listar todos os ramos existentes

git branch -av

Mudar ramo HEAD

git checkout <branch>

Crie uma nova filial com base

git checkout <new-branch>

Crie uma nova filial com base

git checkout --track <remote/branch>

Deletar uma filial local

git branch -d <branch>

Marque o commit atual com uma tag

git tag <tag-name>

ATUALIZAR & PUBLICAR

Lista todos os controles remotos configurados atualmente

git remote -v

Mostrar informações sobre um controle remoto

git remote show <remote>

Adicione um novo repositório remoto, denominado <remote>

git remote add <shortname> <url>

Baixe todas as alterações de <remote>, mas não integre no HEAD

git fetch <remote>

Baixe as alterações e marque / integre diretamente no HEAD

git pull <remote> <branch>

Publique as alterações locais remotamente

git push <remote> <branch>

Exclua uma filial no controle remoto

git branch -dh <remote/branch>

Publique suas tags

git push --tags

FUNDIR & REBASE

Funda <branch> em seu HEAD atual

git merge <branch>

Rebase seu HEAD atual em <branch>

git rebase <branch>

Abortar um rebase

git rebase --abort

Continue um rebase após resolver os conflitos

git rebase --continue

Use a ferramenta de fusão configurada do tour para resolver conflitos

git mergetool

Use o editor de tour para resolver conflitos manualmente e (após resolver) marcar o arquivo como resolvido

git add <resolved-file>

git rm <resolved-file>

DESEFAZER

Descarte todas as mudanças locais em seu diretório de trabalho

git reset --hard HEAD

Descartar as alterações locais em um arquivo específico

git checkout HEAD <file>

Reverter um commit (produzindo um novo commit com mudanças contrárias)

git revert <commit>

Redefina o ponteiro HEAD para um commit anterior ... e descarte todas as alterações desde então

git reset --hard <commit>

... e preserva todas as mudanças como mudanças não planejadas

git reset <commit>

... e preserva as mudanças locais não comprometidas

git reset --keep <commit>

