

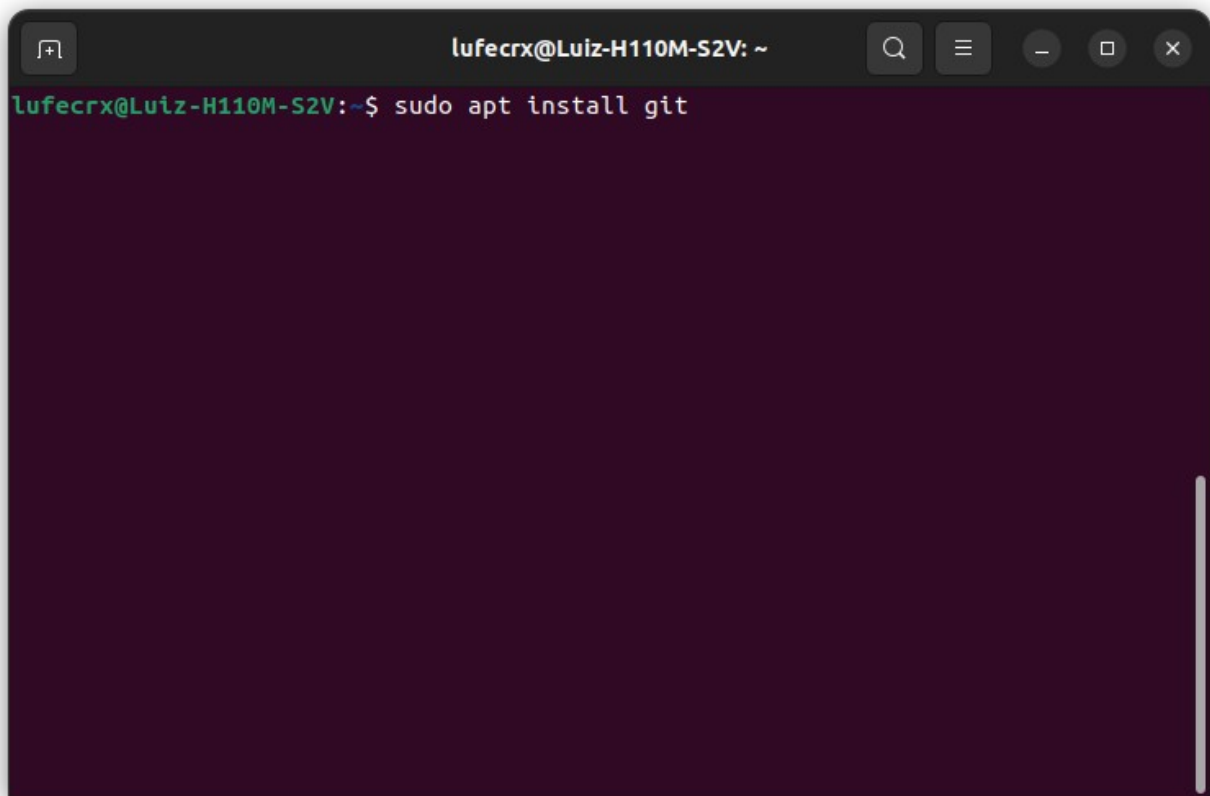
Luiz Felipe Correia Gomes – Turma 1

Instrução Prática 02

*O sistema operacional utilizado foi o Ubuntu 22.04.3

Exercício 1 - Instalando e configurando o Git.

Uma das maneiras de fazer a instalação do Git é usar o comando “sudo apt install git” dentro do terminal do Ubuntu, ou de outro sistema Linux, como no exemplo a seguir:

A screenshot of a terminal window with a dark background. The window title bar shows 'lufecrx@Luiz-H110M-S2V: ~' and standard window controls. The terminal prompt is 'lufecrx@Luiz-H110M-S2V:~\$' and the command 'sudo apt install git' has been entered. The rest of the terminal area is empty, indicating the command is still being processed or the output is not visible in this frame.

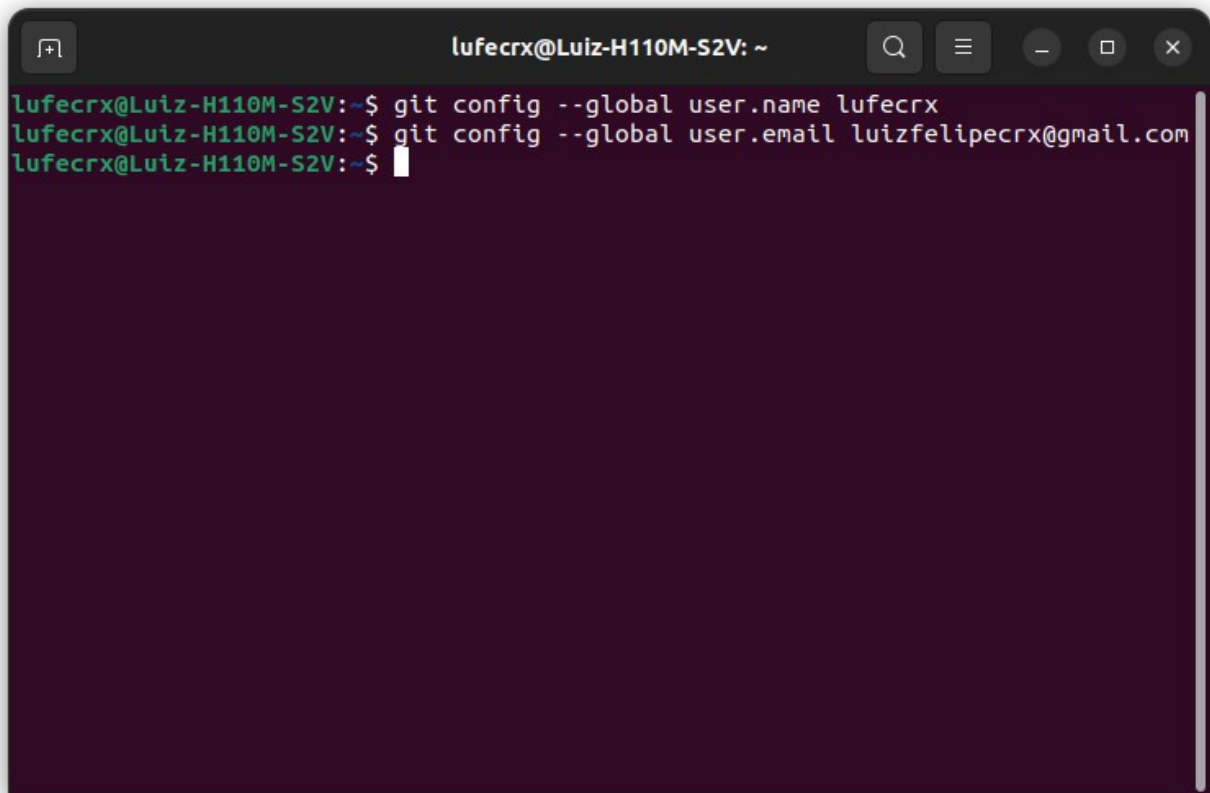
```
lufecrx@Luiz-H110M-S2V:~$ sudo apt install git
```

Depois de executado esse comando, o Git vai ser instalado no computador. Agora vamos configurar o Git, primeiro fazendo a configuração global do nome de usuário e e-mail do Git, em outras palavras, definir qual será o nome de usuário e e-mail usado em todos os repositórios do computador. É recomendado você escolher inserir os mesmos dados usados em sua conta do GitHub.

A configuração do nome de usuário e e-mail no Git são feitas, respectivamente, através desses comandos:

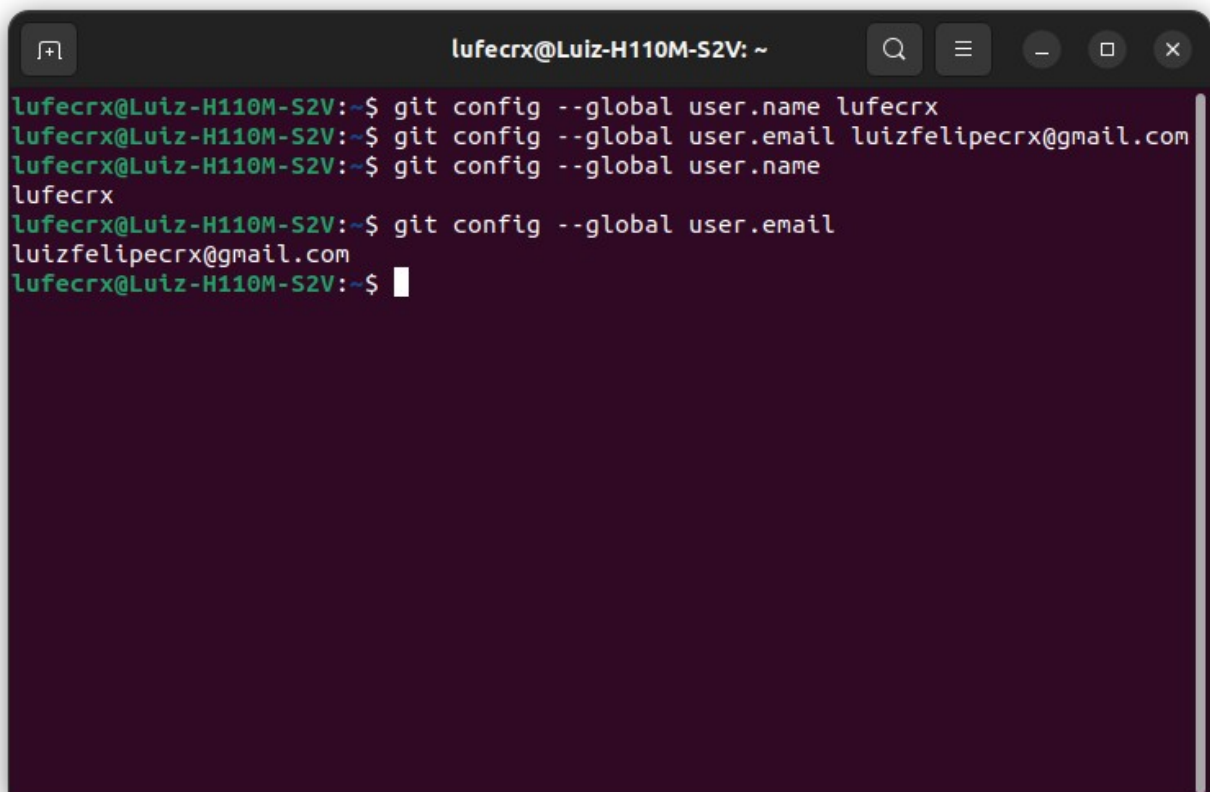
```
git config --global user.name "fulano"  
git config --global user.email "nome@mail.com"
```

Segue o exemplo:

A terminal window with a dark purple background. The title bar shows 'lufecrx@Luiz-H110M-S2V: ~'. The terminal contains three lines of text: a prompt followed by 'git config --global user.name lufecrx', another prompt followed by 'git config --global user.email luizfelipecrx@gmail.com', and a third prompt with a cursor. The text is in a green monospace font.

```
lufecrx@Luiz-H110M-S2V: ~  
lufecrx@Luiz-H110M-S2V:~$ git config --global user.name lufecrx  
lufecrx@Luiz-H110M-S2V:~$ git config --global user.email luizfelipecrx@gmail.com  
lufecrx@Luiz-H110M-S2V:~$
```

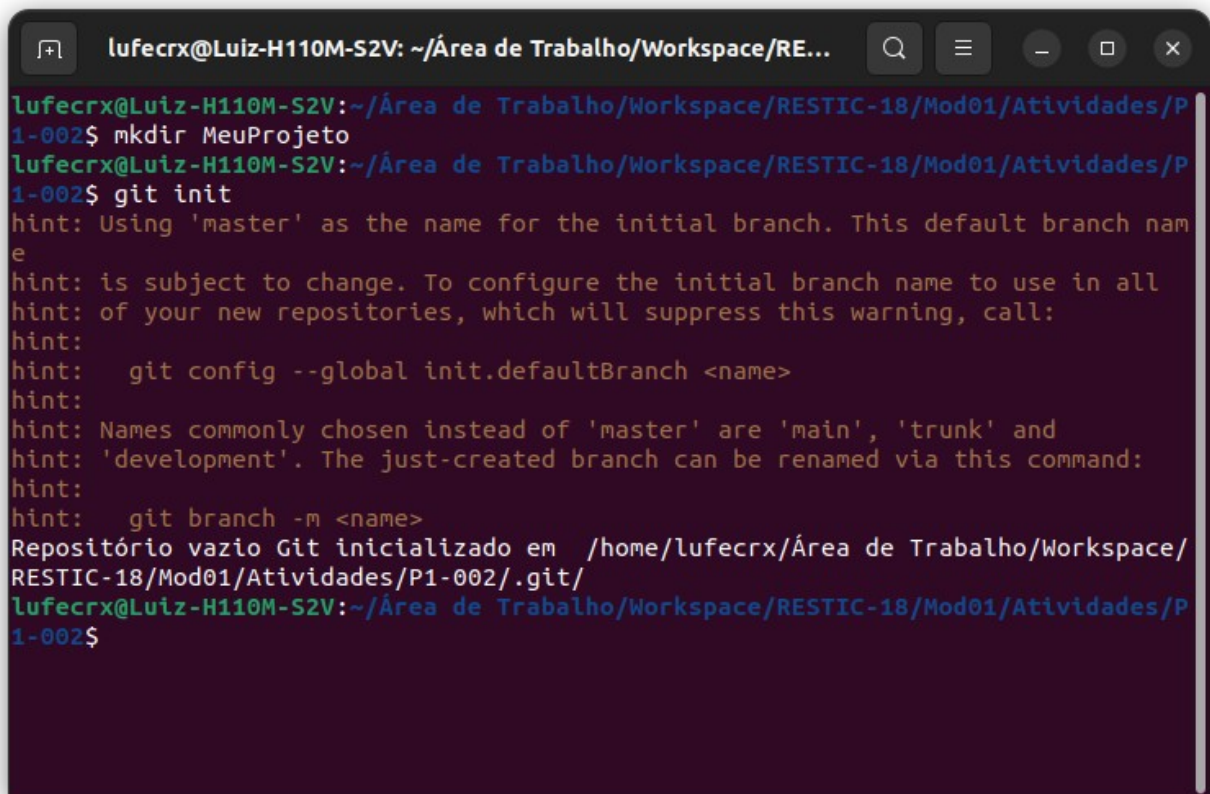
Portanto, nesse caso do exemplo, o nome de usuário e o e-mail que defini para as configurações globais do Git, foram respectivamente “lufecrx” e “luizfelipecrx@gmail.com”. Caso queira verificar se está tudo nos conformes, você pode usar comandos que retornam seus dados no terminal:

A terminal window with a dark purple background. The title bar shows 'lufecrx@Luiz-H110M-S2V: ~'. The terminal contains five lines of text: a prompt followed by 'git config --global user.name lufecrx', another prompt followed by 'git config --global user.email luizfelipecrx@gmail.com', a third prompt followed by 'git config --global user.name' and the output 'lufecrx', a fourth prompt followed by 'git config --global user.email' and the output 'luizfelipecrx@gmail.com', and a fifth prompt with a cursor. The text is in a green monospace font.

```
lufecrx@Luiz-H110M-S2V: ~  
lufecrx@Luiz-H110M-S2V:~$ git config --global user.name lufecrx  
lufecrx@Luiz-H110M-S2V:~$ git config --global user.email luizfelipecrx@gmail.com  
lufecrx@Luiz-H110M-S2V:~$ git config --global user.name  
lufecrx  
lufecrx@Luiz-H110M-S2V:~$ git config --global user.email  
luizfelipecrx@gmail.com  
lufecrx@Luiz-H110M-S2V:~$
```

Exercício 2 – Iniciando um repositório

Para começar a criação de um repositório, vamos usar o comando “git init” em um diretório vazio chamado “MeuProjeto”:

A terminal window with a dark background and light-colored text. The window title is 'lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...'. The terminal shows the user creating a directory 'MeuProjeto' and then running 'git init'. The output of 'git init' includes several hints about the default branch name 'master' and how to change it. The final message states that the repository is initialized in the specified path.

```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$ mkdir MeuProjeto
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Repositório vazio Git inicializado em /home/lufecrx/Área de Trabalho/Workspace/
RESTIC-18/Mod01/Atividades/P1-002/.git/
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$
```

Feito o git init é comum que seja mostrado no terminal algumas dicas e o repositório vazio é inicializado. Dentro desse diretório vazio, vamos criar um arquivo de texto e adicionar uma breve introdução do projeto:

```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ echo "Introdução de Git para o programa de Residência em TIC."
>> introducao.txt
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ ls
introducao.txt
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$
```

Pronto, agora temos um repositório com o Git inicializado em nosso diretório contendo um arquivo de texto. Vamos adicionar o comando “git add” para incluir o arquivo de texto que será submetido:

```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git status
No ramo main

No commits yet

Arquivos não monitorados:
  (utilize "git add <arquivo>..." para incluir o que será submetido)
  ./

nada adicionado ao envio mas arquivos não registrados estão presentes (use "git add" to registrar)
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git add introducao.txt
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git status
No ramo main

No commits yet

Mudanças a serem submetidas:
  (utilize "git rm --cached <arquivo>..." para não apresentar)
    new file:   introducao.txt
```

E em seguida fazer o commit:

```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P
1-002/MeuProjeto$ git status
No ramo main

No commits yet

Mudanças a serem submetidas:
  (utilize "git rm --cached <arquivo>..." para não apresentar)
    new file:   introducao.txt

lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P
1-002/MeuProjeto$ git commit -m "commit inicial"
[main (root-commit) 2f36954] commit inicial
 1 file changed, 1 insertion(+)
 create mode 100644 MeuProjeto/introducao.txt
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P
1-002/MeuProjeto$ git status
No ramo main
nothing to commit, working tree clean
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P
1-002/MeuProjeto$
```

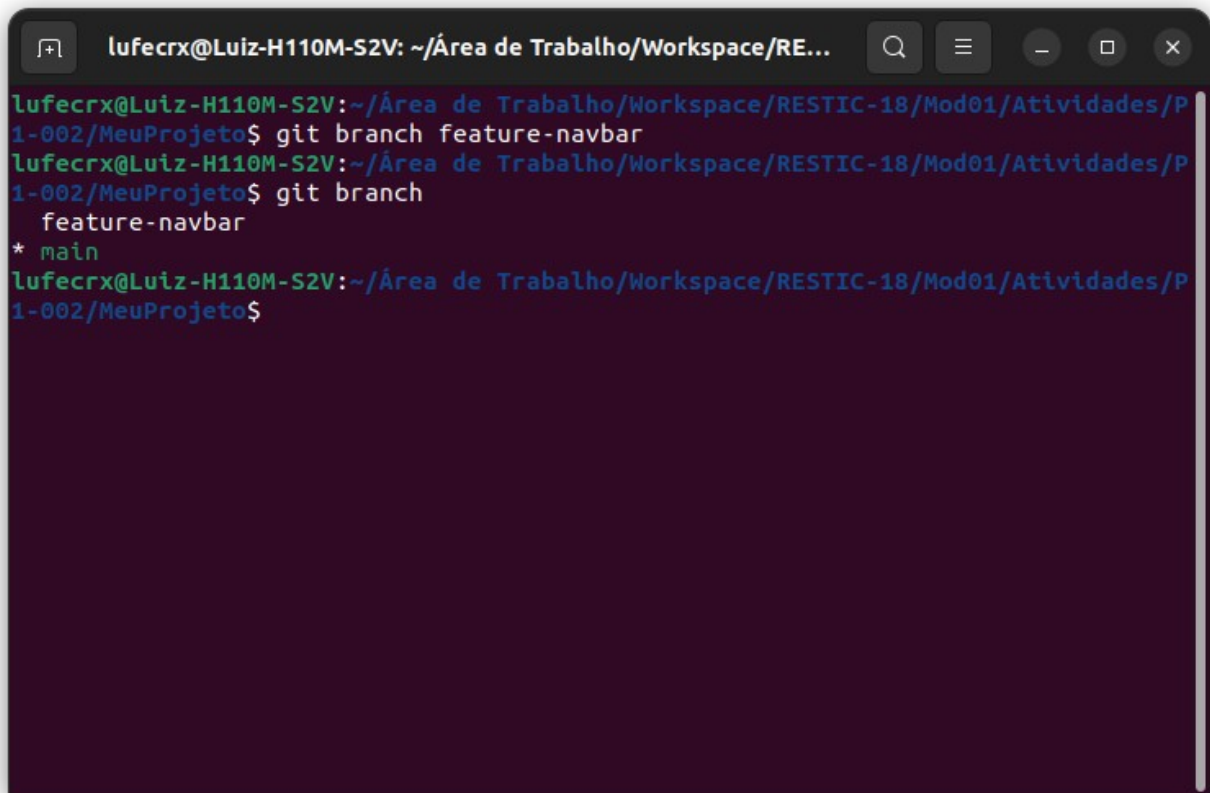
Feito isso, vamos seguir para as próximas etapas.

Exercício 3 – Trabalhando com Branches

Vamos criar um branch chamado “feature-navbar”, para isso vamos executar o seguinte código em nosso terminal:

```
git branch nome-da-branch
```

Esse comando vai criar a nova branch com o nome especificado por parâmetro. Segue o exemplo:

A terminal window with a dark background and light-colored text. The window title is 'lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...'. The terminal shows the following commands and output: 'git branch feature-navbar' is executed, followed by 'git branch', which outputs '* main'.

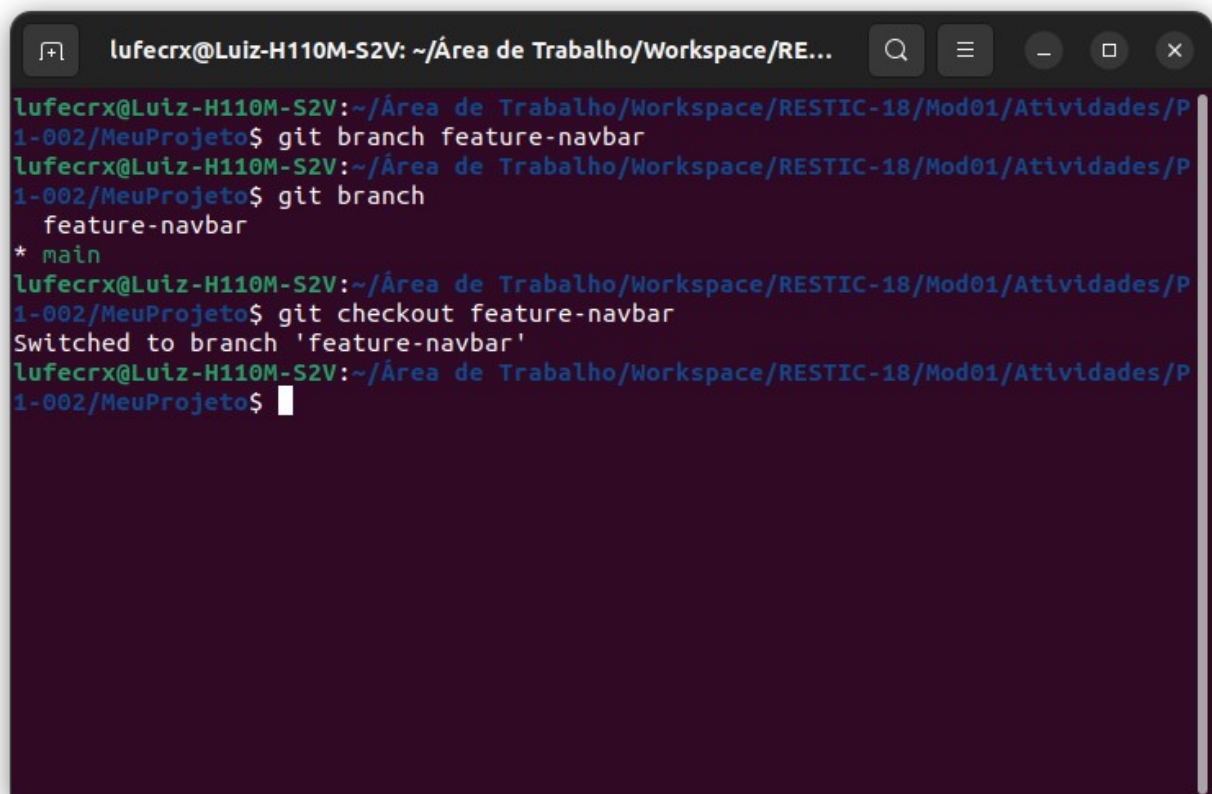
```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git branch feature-navbar
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git branch
feature-navbar
* main
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$
```

Nosso repositório git agora tem duas branches: a branch main e a branch feature-navbar que acabamos de criar. O comando “git branch” pode ser usado para listar as branches existentes no repositório.

Para mudarmos do repositório atual para o repositório que criamos, pode ser usado o seguinte comando:

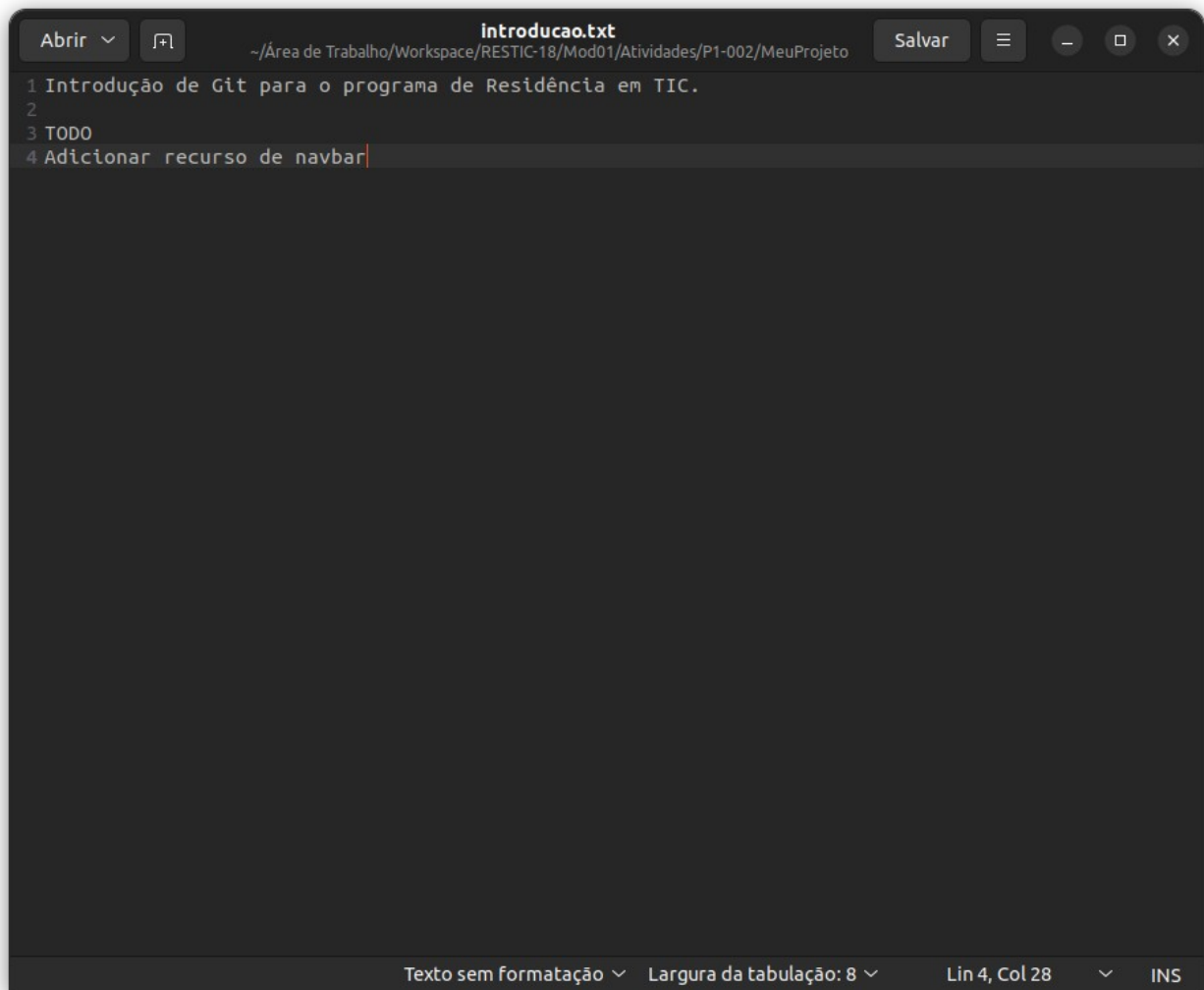
git checkout nome-da-branch

Exemplo:



```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git branch feature-navbar
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git branch
* main
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git checkout feature-navbar
Switched to branch 'feature-navbar'
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$
```

Agora que estamos nessa branch, vamos adicionar algumas informações no arquivo de texto que criamos anteriormente nesse diretório:



```
1 Introdução de Git para o programa de Residência em TIC.  
2  
3 TODO  
4 Adicionar recurso de navbar
```

Texto sem formatação ▾ Largura da tabulação: 8 ▾ Lin 4, Col 28 ▾ INS

Vamos incluir para o commit as alterações que foram feitas usando o comando:
`git add nome-do-arquivo`

E feito isso, vamos realizar um commit das alterações feitas no branch “feature-navbar”. Para tal, verificamos se estamos na branch correta e fazer o commit com o seguinte comando:

`git commit -m “mensagem do commit”`

Exemplo:


```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git add introducao.txt
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git branch
* feature-navbar
  main
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git commit -m "novos recursos para adicionar"
[feature-navbar 3d1eeda] novos recursos para adicionar
1 file changed, 3 insertions(+)
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git status
No ramo feature-navbar
nothing to commit, working tree clean
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$
```

Se voltarmos para o branch principal e procurar as alterações do branch feature-navbar, perceberemos que essas alterações não estarão visíveis.

```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git checkout main
Switched to branch 'main'
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$
```

Abrir ▾

introducao.txt
~/Área de Trabalho/...

Salvar

≡

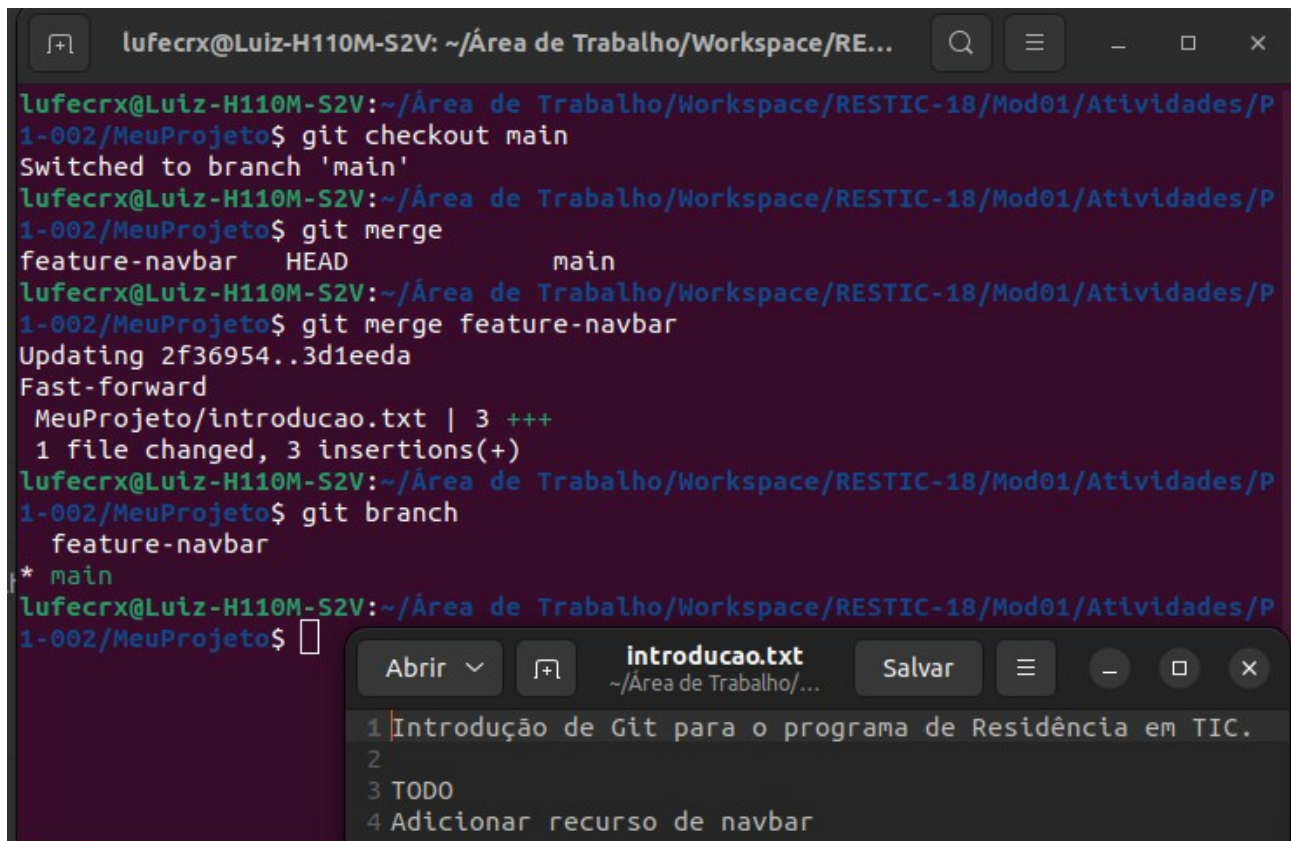
—

□

×

```
1 Introdução de Git para o programa de Residência em TIC.
```

Para trazer as alterações feitas na outra branch para a atual, uma possibilidade é fazer o merge.



```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git checkout main
Switched to branch 'main'
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git merge
feature-navbar HEAD main
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git merge feature-navbar
Updating 2f36954..3d1eeda
Fast-forward
 MeuProjeto/introducao.txt | 3 +++
 1 file changed, 3 insertions(+)
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git branch
  feature-navbar
* main
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$
```

Abrir ▾ introducao.txt ~/Área de Trabalho/... Salvar ≡ - □ ×

```
1 Introdução de Git para o programa de Residência em TIC.
2
3 TODO
4 Adicionar recurso de navbar
```

O comando “git merge branch-alvo” vai fazer que o Git crie um novo commit de merge que combina as alterações dos branches envolvidos.

Exercício 4 – Ignorando arquivos

Para evitar que determinados arquivos e pastas do diretório sejam versionados, o Git tem um recurso interessante chamado gitignore. É um arquivo onde você pode listar os arquivos e pastas que não devem ser versionados pelo Git. Vamos criar um arquivo “secreto.txt” e adicionar ele ao .gitignore para evitar que seja versionado, segue o exemplo:

```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ echo "arquivo que n vai ser versionado" >> secreto.txt
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ echo "secreto.txt" >> .gitignore
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ ls
introducao.txt  secreto.txt
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ ls -a
.  ..  .gitignore  introducao.txt  secreto.txt
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$
```

O que fizemos foi criar o arquivo de texto “secreto”, criar o .gitignore já passando o nome do arquivo que vai ser ignorado pelo versionamento. E depois listamos os arquivos do diretório, o .gitignore é um arquivo oculto por padrão.

```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
1-002$ git status
No ramo main
Arquivos não monitorados:
  (utilize "git add <arquivo>..." para incluir o que será submetido)
    P1-2.odt

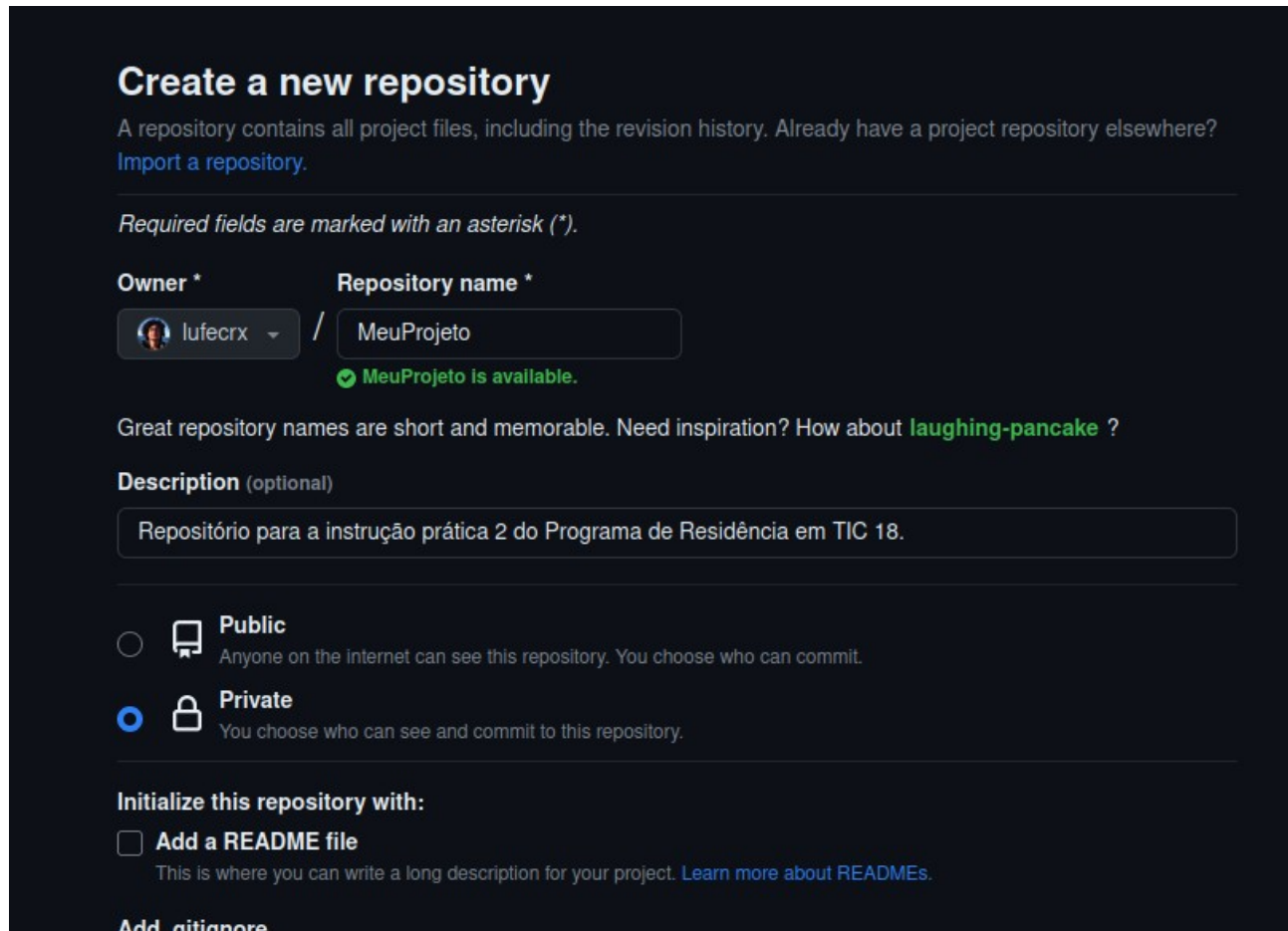
nada adicionado ao envio mas arquivos não registrados estão presentes (use "git add" to registrar)
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$ git add *
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$ git status
No ramo main
Mudanças a serem submetidas:
  (use "git restore --staged <file>..." to unstage)
    new file:   P1-2.odt

lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$ git commit -m "atualizando arquivo P1-2.odt"
[main 2a2774d] atualizando arquivo P1-2.odt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 P1-2.odt
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$
```


Como pode ser visto acima, o arquivo “secreto.txt” foi ignorado pelo versionamento. Esse recurso é muito útil quando trabalhamos em programas mais complexos e não queremos versionar para o nosso repositório dependências desnecessárias do projeto.

Exercício 5 – Trabalhando com repositórios remotos

Trabalhando com repositórios remotos, vamos começar criando o repositório no GitHub:




Create a new repository

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

Required fields are marked with an asterisk ().*

Owner * **Repository name ***

 lufecrx /

✔ **MeuProjeto is available.**

Great repository names are short and memorable. Need inspiration? How about **laughing-pancake** ?

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.

Initialize this repository with:

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

Add .gitignore

Feito isso, vamos para a próxima etapa de adicionar o repositório remoto em nosso repositório local. Para isso, usamos o seguinte comando no terminal:

```
git remote add origin git@github.com:user.name/NomeRepositorio.git
```

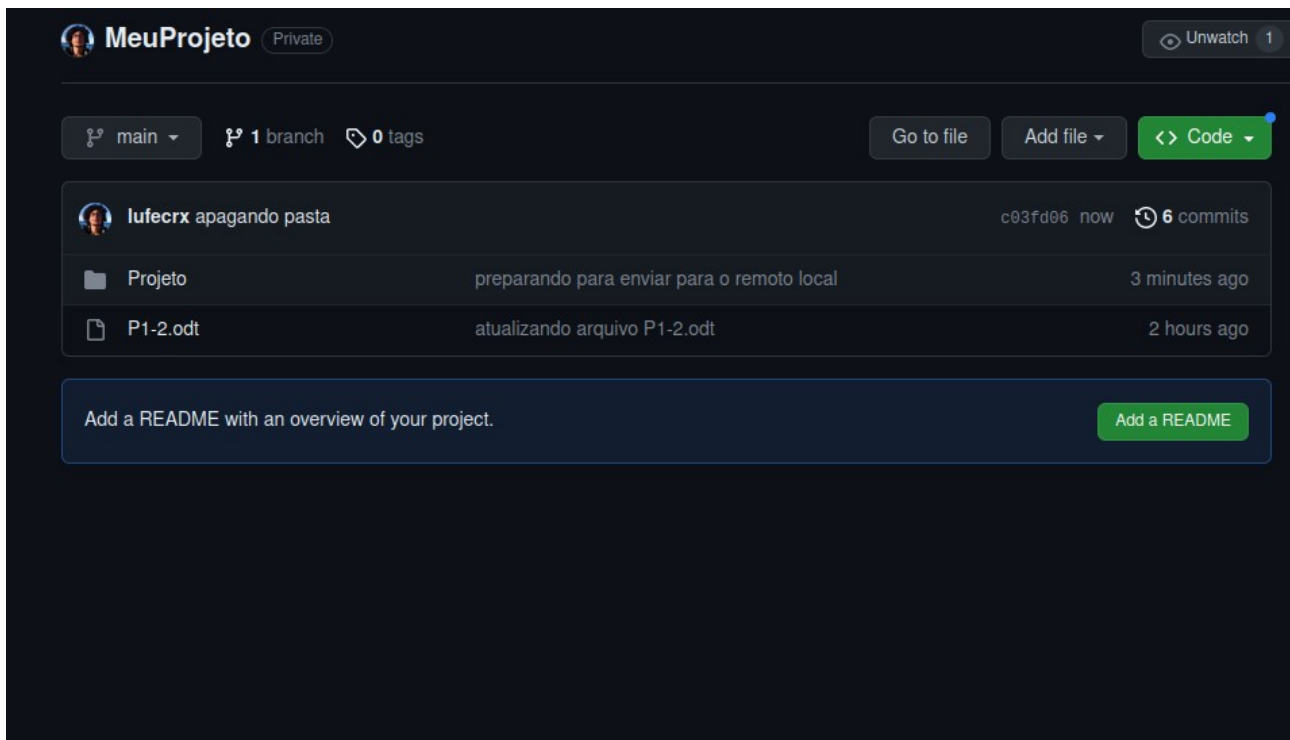
*Substitua “user.name” e “NomeRepositorio”, pelas informações corretas. Segue o exemplo:

```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git remote add origin git@github.com:lufecrx/MeuProjeto.git
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git remote
origin
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$ git remote -v
origin  git@github.com:lufecrx/MeuProjeto.git (fetch)
origin  git@github.com:lufecrx/MeuProjeto.git (push)
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/MeuProjeto$
```

Agora estamos com o repositório local conectado ao repositório remoto no GitHub. Vamos dar um push e enviar as alterações locais para o repositório remoto:

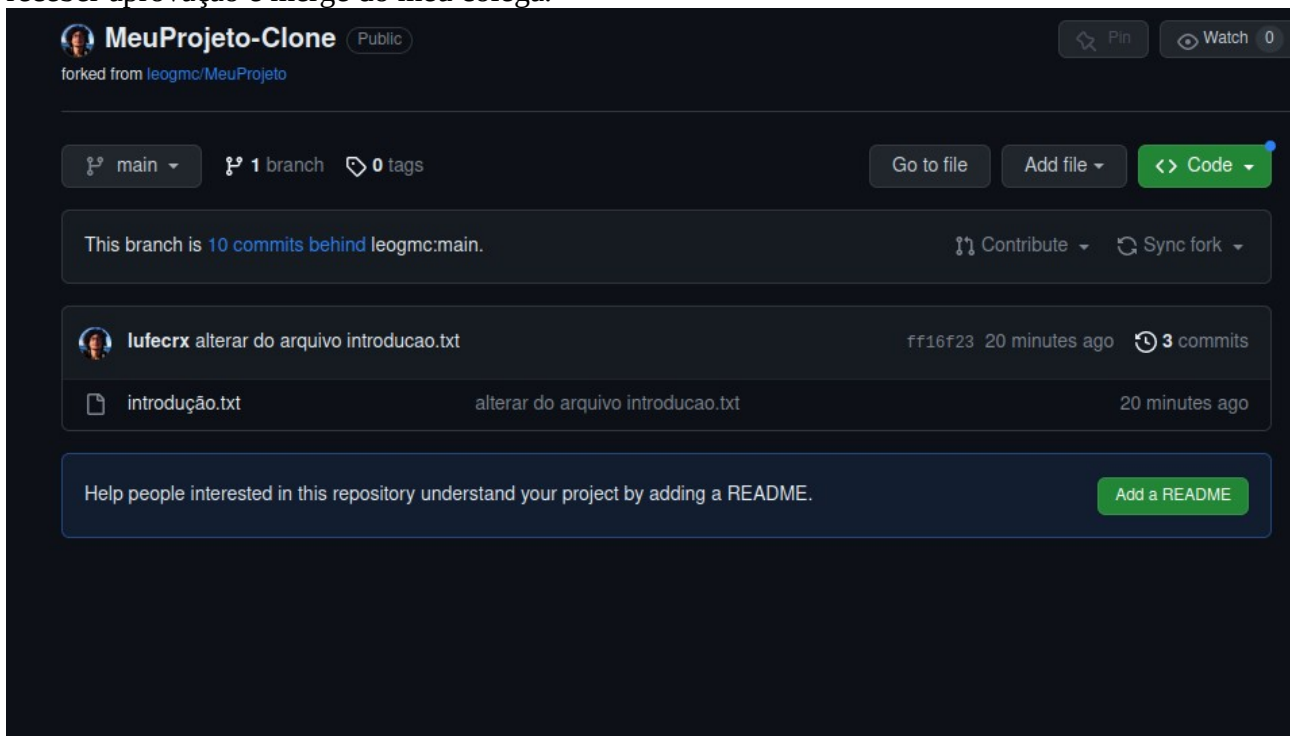
```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$ git remote -v
origin  git@github.com:lufecrx/MeuProjeto.git (fetch)
origin  git@github.com:lufecrx/MeuProjeto.git (push)
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$ git push -u origin main
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 4 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (17/17), 953.72 KiB | 5.96 MiB/s, done.
Total 17 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To github.com:lufecrx/MeuProjeto.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$
```

Pronto! Se formos para o GitHub vamos poder ver nosso projeto no repositório:



Exercício 6 – Clonando e Forking

Clonei o repositório remoto de meu colega, fiz alterações no arquivo “introducao.txt” e upei as alterações para o repositório remoto, fazendo um pull request para que minhas alterações pudessem receber aprovação e merge do meu colega.



Fiz pull do repositório de um colega do projeto, fiz alterações de um dos arquivos e dei push para o repositório remoto. Inicialmente, haviam acontecido conflitos entre o meu pull request com o de colegas que também fizeram pull request, mas os conflitos foram resolvidos e deu tudo certo.

colaborando com o projeto do colega #5

Open

lufecrx wants to merge 1 commit into leogmc:main from lufecrx:main

Conversation 0

Commits 1

Checks 0

Files changed 1

+2 -1

lufecrx commented 4 minutes ago

Contributor

Opal!!!!

👍

colaborando com o projeto do colega

2abf61d

Add more commits by pushing to the main branch on lufecrx/MeuProjeto-Clone

Require approval from specific reviewers before merging

Branch protection rules ensure specific people approve pull requests before they're merged.

Add rule

Continuous integration has not been set up

GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

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.

Reviews

Suggestions

leogmc

Request

Still in progress? Convert to draft

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
1-002$ git log
commit c03fd06a7089daf66f37eeadb088e3fcf6d30821 (HEAD -> main, origin/main)
Author: lufecrx <luizfelipecrx@gmail.com>
Date: Thu Aug 31 01:07:09 2023 -0300

    apagando pasta

commit a06b33e044745f9d1f74d21fdc293fe8aa7ec33b
Author: lufecrx <luizfelipecrx@gmail.com>
Date: Thu Aug 31 01:04:02 2023 -0300

    preparando para enviar para o remoto local

commit 2a2774d95eb0694d4ecc0b622c60edfa334da857
Author: lufecrx <luizfelipecrx@gmail.com>
Date: Wed Aug 30 23:08:21 2023 -0300

    atualizando arquivo P1-2.odt

commit 54db061eda24fd5986a783055148580d98f533ca
Author: lufecrx <luizfelipecrx@gmail.com>
Date: Wed Aug 30 21:30:31 2023 -0300

    adicionado .gitignore

commit 3d1eeda538c4edd70367f7ac9a4eb753691c6db6 (feature-navbar)
Author: lufecrx <luizfelipecrx@gmail.com>
Date: Wed Aug 30 02:57:53 2023 -0300

    novos recursos para adicionar

commit 2f369548b8f9be766eab423a7b7ff6c8df153a1c
Author: lufecrx <luizfelipecrx@gmail.com>
Date: Wed Aug 30 02:40:30 2023 -0300

    commit inicial
(END)
```

De baixo para cima:

- commit inicial: primeiro commit do repositório;
- novos recursos para adicionar: mudança da branch main para o feature-navbar;
- adicionado .gitignore: adicionar arquivo gitignore;
- atualizando arquivo P1-2.odt: atualizar documento de texto sobre a atividade;
- preparando para enviar para o remoto local: alterações básicas no repositório;
- apagando pasta: deletado pasta desnecessária do repositório.

Exercício 9 – Revertendo Alterações

Vamos imaginar que supostamente uma alteração indesejada seja feita em nosso repositório e realize-se um commit. Como podemos reverter isso? Basta utilizar o `git revert`. Segue o exemplo:

```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$ git log
commit ae0489bc4d58ddde0ff198a3534a679f606374e8 (HEAD -> main)
commit ae0489bc4d58ddde0ff198a3534a679f606374e8 (HEAD -> main)
Author: lufecrx <luizfelipecrx@gmail.com>
Date: Thu Aug 31 02:57:47 2023 -0300

    simulando alteracao indesejada

commit c8431aa6435d154e13e5c0a49ae6958d08d99b00
Author: lufecrx <luizfelipecrx@gmail.com>
Date: Thu Aug 31 02:55:26 2023 -0300

    arquivo de texto atualizado

commit c03fd06a7089daf66f37eeadb088e3fcf6d30821 (origin/main)
Author: lufecrx <luizfelipecrx@gmail.com>
Date: Thu Aug 31 01:07:09 2023 -0300

    apagando pasta

commit a06b33e044745f9d1f74d21fdc293fe8aa7ec33b
Author: lufecrx <luizfelipecrx@gmail.com>
Date: Thu Aug 31 01:04:02 2023 -0300
```

Queremos voltar para o commit com mensagem “arquivo de texto atualizado”, para isso usamos a chave desse commit, que pode ser representada por c8431a, dessa forma:

- No terminal e dentro do seu repositório local, insira o comando
git revert “chave”

```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
...balho/Workspace/RESTIC-18/Mod01/Atividades/P1-002/.git/COMMIT_EDITMSG
Revert "arquivo de texto atualizado"

This reverts commit c8431aa6435d154e13e5c0a49ae6958d08d99b00.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# No ramo main
# Seu ramo está à frente de 'origin/main' por 2 submissões.
# (use "git push" to publish your local commits)
#
# Mudanças a serem submetidas:
#     modified:   P1-2.odt
#
# Arquivos não monitorados:
#     .~lock.P1-2.odt#
#     da
#

^G Ajuda      ^O Gravar     ^W Onde está? ^K Recortar   ^T Executar   ^C Local
^X Sair       ^R Ler o arq ^\ Substituir ^U Colar     ^J Justificar ^_ Ir p/ linha
```

Essa mensagem vai aparecer, você vai poder alterar a mensagem do commit entre aspas e gravar as alterações com Ctrl + O. Isso vai gerar um novo commit que vai retornar para o estado do commit que foi revertido. Se você der um git log, depois de gravar o revert, será possível visualizar isso melhor:

```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P
1-002$ git log
commit 48af2deb97535faae9ddb4ab672da2ab034f0f5 (HEAD -> main)
Author: lufecrx <luizfelipecrx@gmail.com>
Date: Thu Aug 31 03:03:57 2023 -0300

    Revert "arquivo de texto atualizado - commit corrigido"

    This reverts commit c8431aa6435d154e13e5c0a49ae6958d08d99b00.

commit ae0489bc4d58ddde0ff198a3534a679f606374e8
Author: lufecrx <luizfelipecrx@gmail.com>
Date: Thu Aug 31 02:57:47 2023 -0300

    simulando alteracao indesejada

commit c8431aa6435d154e13e5c0a49ae6958d08d99b00
Author: lufecrx <luizfelipecrx@gmail.com>
Date: Thu Aug 31 02:55:26 2023 -0300

    arquivo de texto atualizado
```

Exercício 10 – Trabalhando com Tags

Criando uma tag no meu repositório para marcar a versão 1.0 do projeto. Para fazer isso, vamos abrir o terminal em nosso repositório e inserir o seguinte comando:

```
git tag -a v1.4 -m "my version x.x"
```

Exemplo:

```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$ git tag -a v1.0 -m "version 1.0"
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$
```

Se quisermos ver mais informações sobre a tag criada, podemos usar o seguinte comando em nosso terminal:

`git show v1.0`

```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$ git tag -a v1.0 -m "version 1.0"
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$ git show v1.0
tag v1.0
Tagger: lufecrx <luizfelipecrx@gmail.com>
Date: Thu Aug 31 03:26:49 2023 -0300

version 1.0

commit 48af2deb97535faae9dd0b4ab672da2ab034f0f5 (HEAD -> main, tag: v1.0)
Author: lufecrx <luizfelipecrx@gmail.com>
Date: Thu Aug 31 03:03:57 2023 -0300

    Revert "arquivo de texto atualizado - commit corrigido"

    This reverts commit c8431aa6435d154e13e5c0a49ae6958d08d99b00.

diff --git a/P1-2.odt b/P1-2.odt
index 6b88568..154e992 100644
Binary files a/P1-2.odt and b/P1-2.odt differ
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$
```


Vamos fazer algumas alterações no arquivo “introdução.txt” e criar uma nova tag para marcar a versão 2.0.

```
lufecrx@Luiz-H110M-S2V: ~/Área de Trabalho/Workspace/RE...
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$ git commit -m "alterar para versao 2.0"
[main 86582fc] alterar para versao 2.0
2 files changed, 2 insertions(+), 1 deletion(-)
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$ git tag -a v2.0 -m "version 2.0"
lufecrx@Luiz-H110M-S2V:~/Área de Trabalho/Workspace/RESTIC-18/Mod01/Atividades/P1-002$ git show v2.0
tag v2.0
Tagger: lufecrx <luizfelipecrx@gmail.com>
Date: Thu Aug 31 03:36:40 2023 -0300

version 2.0

commit 86582fc25c31ed4d3ba66a0a611064534e0d8aad (HEAD -> main, tag: v2.0)
Author: lufecrx <luizfelipecrx@gmail.com>
Date: Thu Aug 31 03:36:19 2023 -0300

    alterar para versao 2.0

diff --git a/P1-2.odt b/P1-2.odt
index 154e992..915a6ba 100644
Binary files a/P1-2.odt and b/P1-2.odt differ
diff --git a/Projeto/introducao.txt b/Projeto/introducao.txt
```