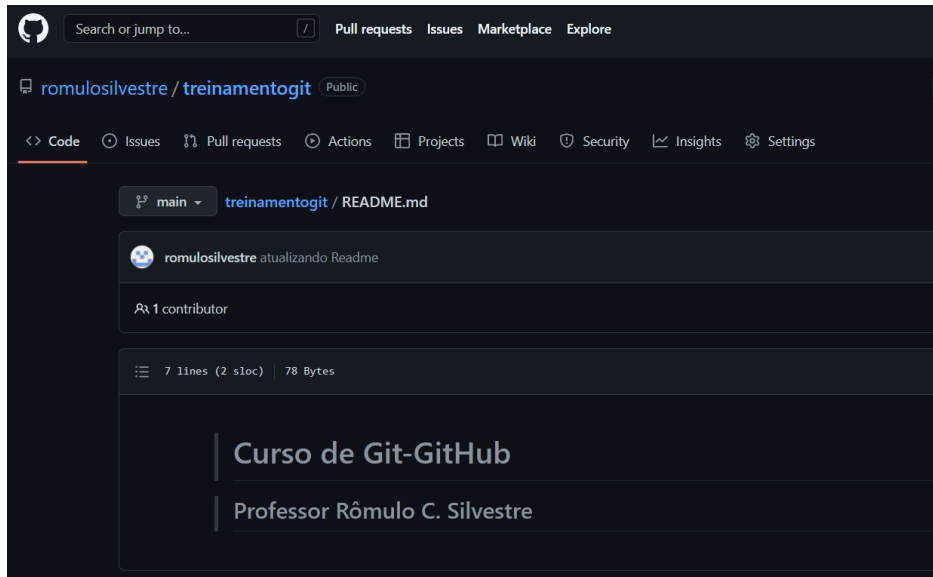


# Lista de Exercícios

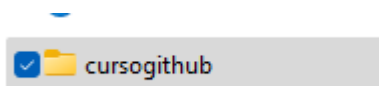
## Questão 01

Você foi solicitado em uma entrevista de emprego para criar um repositório e fornecer o endereço desse repositório. O repositório deve conter um arquivo README.md com uma apresentação bem básica conforme exemplo:

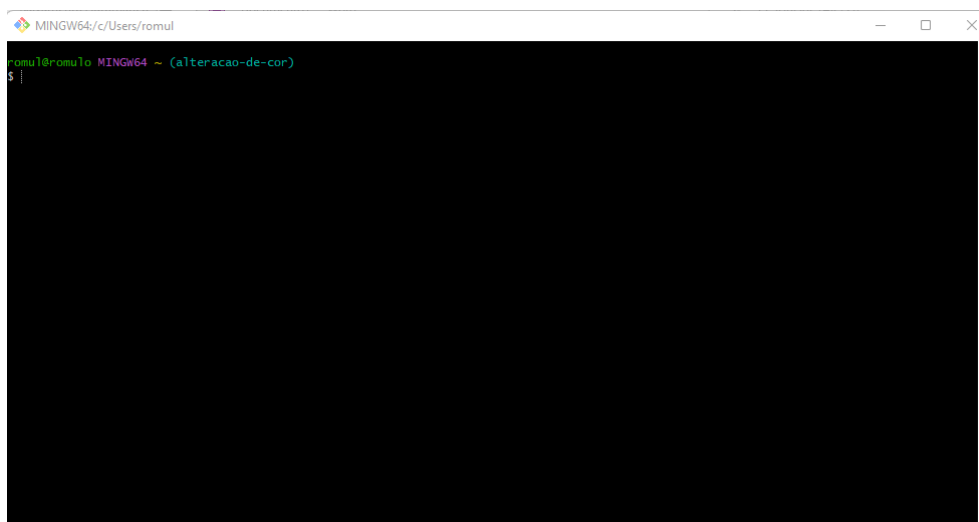


## Questão 02

Crie uma pasta em seu local preferido com o seguinte nome:



Abra o gitbash



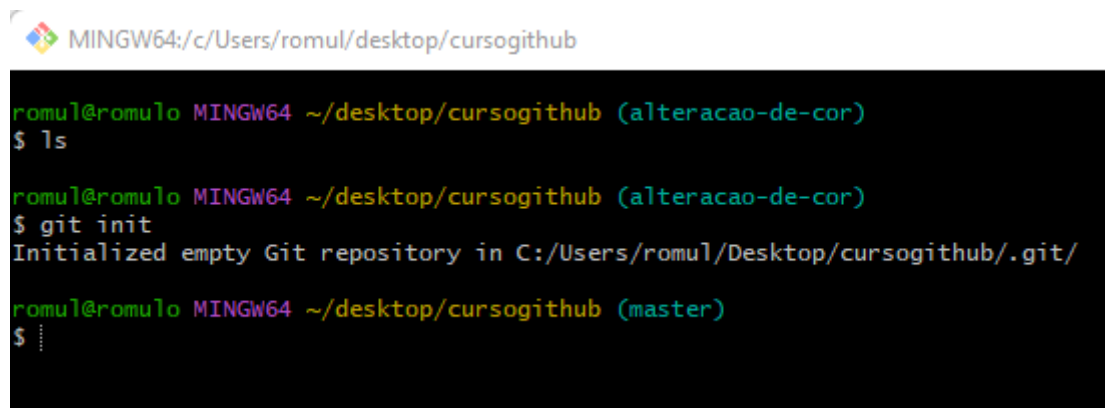
Utilize comandos de **cd** , **cd ..** ou **ls** para listar, navegar e ou voltar nas pastas.

Seu objetivo é entrar na pasta que você criou no seu lugar preferido da questão 1.

romul@romulo MINGW64 ~/desktop/cursogithub (alteracao-de-cor)

\$ ls

**Inicialize o repositório usando o comando git init**

A terminal window with a black background and green text. The title bar shows the Windows logo and the path 'MINGW64:/c/Users/romul/desktop/cursogithub'. The prompt is 'romul@romulo MINGW64 ~/desktop/cursogithub (alteracao-de-cor)'. The user enters '\$ ls'. The prompt changes to 'romul@romulo MINGW64 ~/desktop/cursogithub (alteracao-de-cor)'. The user enters '\$ git init'. The output is 'Initialized empty Git repository in C:/Users/romul/Desktop/cursogithub/.git/'. The prompt changes to 'romul@romulo MINGW64 ~/desktop/cursogithub (master)'. The user enters '\$' followed by a cursor.

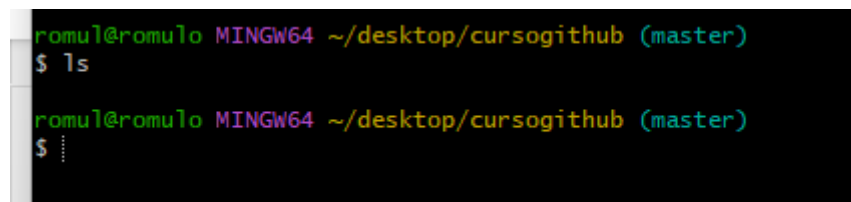
```
romul@romulo MINGW64 ~/desktop/cursogithub (alteracao-de-cor)
$ ls

romul@romulo MINGW64 ~/desktop/cursogithub (alteracao-de-cor)
$ git init
Initialized empty Git repository in C:/Users/romul/Desktop/cursogithub/.git/

romul@romulo MINGW64 ~/desktop/cursogithub (master)
$
```

Isso significa agora que esta pasta vai comportar um repositório do git.

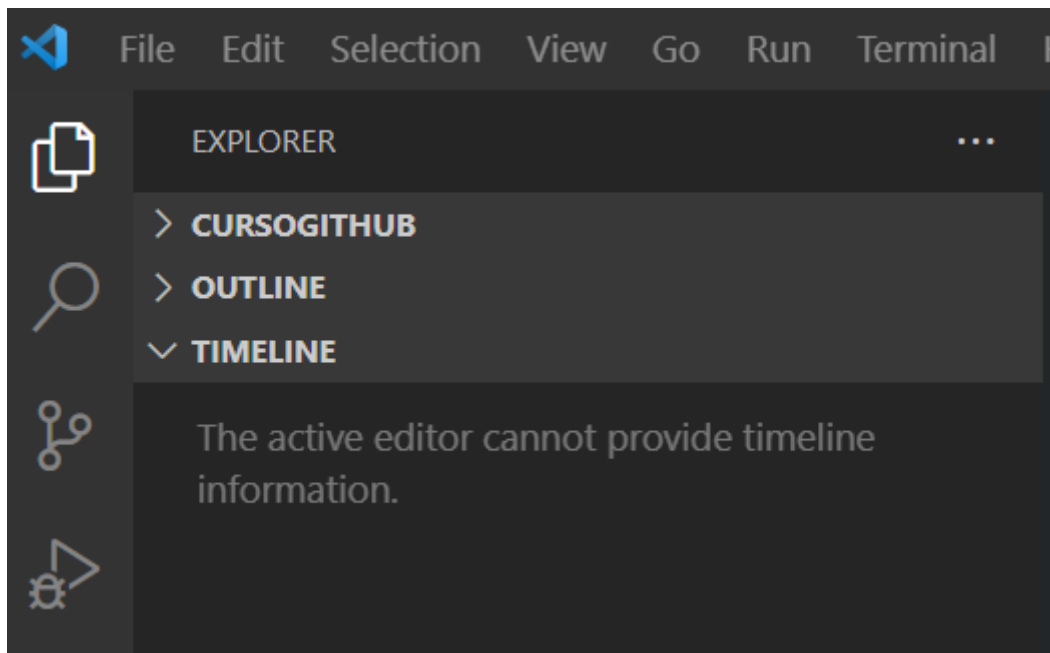
O repositório deve estar vazio.

A terminal window with a black background and green text. The title bar shows the Windows logo and the path 'MINGW64:/c/Users/romul/desktop/cursogithub'. The prompt is 'romul@romulo MINGW64 ~/desktop/cursogithub (master)'. The user enters '\$ ls'. The prompt changes to 'romul@romulo MINGW64 ~/desktop/cursogithub (master)'. The user enters '\$' followed by a cursor.

```
romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ ls

romul@romulo MINGW64 ~/desktop/cursogithub (master)
$
```

Agora abra o seu Visual Studio Code e nele abra a pasta do repositório.



Ele cria um diretório chamado `.git/`, porém, não é visualizado pois é um arquivo oculto.

Agora volte no gitbash e abra o nano.

```
romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ nano index.html
```

Coloque uma tag h1

```
MINGW64:/c/Users/romul/desktop/cursogithub
GNU nano 6.2
<h1>Testando Repositório</h1>
```

Control X para Salvar

Agora vamos usar o git status

Nele ele vai dizer que estamos na **branch master**

Não houve **nenhum** commit por enquanto

Temos alguns arquivos não rastreados

O git ele funciona assim:

- Ao adicionar um arquivo precisamos informar que ele vai ser gerenciado pelo git.
- A partir daí ele vai controlar todo o histórico desse arquivo.
  - Tudo que você fizer com esse arquivo vai ser rastreado
- Você pode usar `git add file` para adicionar
- Ele fala com arquivo não está rastreado.
- Não há nenhuma informação para ser adicionada no commit
- Lembre-se para iniciar é `git init` e para visualizar informações sobre o repositório é o `git status`.

```
romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ git status
On branch master

No commits yet

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

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

romul@romulo MINGW64 ~/desktop/cursogithub (master)
$
```

## Questionário



Questão 1 de 2

Qual comando responsável por inicializar um repositório?

- ☐ `git init.`
- ☐ `git status.`
- ☐ `init git.`
- ☐ `status git.`
- ☐ `git new.`



### Questão 2 de 2

Qual comando responsável por obter o status de um repositório?

- ☐ `git init.`
- ☐ `git status.`
- ☐ `git log.`
- ☐ `init git.`
- ☐ `status git.`

### Questão 3

Um arquivo tem três estados a saber:

**Modificado:** quando há alguma alteração, conteúdo adicionado ou removido.

**Preparado:** quando ele é adicionado ao versionamento usando o comando **git add**.

**Consolidado:** após serem modificados (adicionados, removidos ou alterados) e preparados (adicionados ao versionamento), o último passo de um arquivo é sua consolidação. Neste passo, quando você coloca **git commit**, estamos salvando as alterações do arquivo e o versionamento, mantendo um histórico de suas alterações.



O que é um commit então?

Pelo que vimos até aqui no curso o commit serve para salvar as alterações realizadas em um arquivo.

Em uma determinada situação problema você está finalizando as alterações no arquivo da nossa loja virtual e você quer salvar o arquivo.

Então você deve “commitar” colocando uma mensagem para identificar essa alteração.

O commit é um dos recursos principais do git. Com ele podemos criar o histórico de alterações. Podemos salvar ou desfazer alterações (rollback).

Lembre-se que aprendemos no curso configurar o usuário do git:

**git config --global user.name "SEU NOME"**

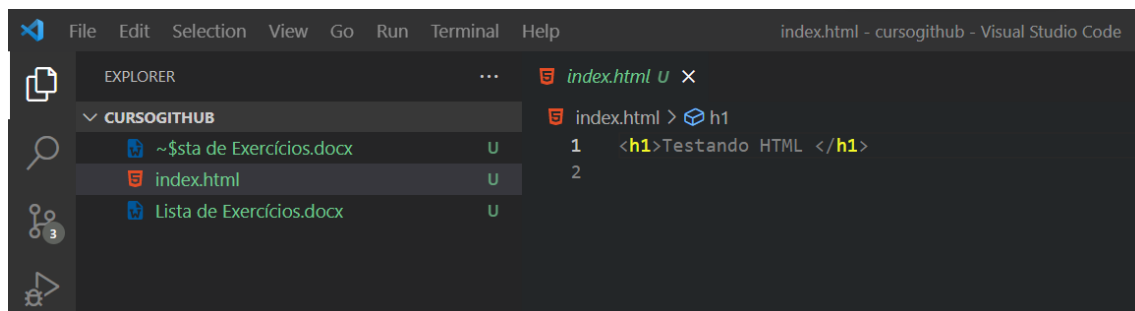
**git config --global user.email "SEU EMAIL"**

```
romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ git config --global user.name "romulosilvestre"

romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ git config --global user.email "romulo.silvestre@df.estudante.senai.br"
```

Vamos abrir o novo Visual Studio Code para visualizar os novos arquivos adicionados.

Adicionei também esses exercícios.



#### Questão 04

Agora você irá demonstrar o que aprendeu sobre os **commits**.

É com o commit que salvamos (tipo um CTRL +S)

Antes de commit devemos adicionar o arquivo para que ele seja trackeado pelo git.

Lembra que nas questões anteriores eles não estava

Então vamos usar o comando git add.

Adicione o arquivo index.html

Adicione também esse arquivo do word. Essa lista de exercícios dentro do repositório.

```

romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html

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

romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ git add listadeexercicios.docx

romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html
        new file:   listadeexercicios.docx

romul@romulo MINGW64 ~/desktop/cursogithub (master)
$


```

Utilize git status até ele mostrar que foi adicionado novos arquivos.

Ele mostra a mensagem dizendo que algumas alterações não foram “comitadas”.

git commit -m “insira aqui o comentário”

Então abra o index.html no nano e adicione mais um <h1> ao arquivo.

 MINGW64:/c/Users/romul/desktop/cursogithub

```

GNU nano 6.2
<h1>Testando HTML </h1>
<h1>Testando De Novo uma Tag HTML</h1>

```

CTRL+X para salvar e voltar para o GitBash

```

romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ git commit -m "adicionado uma tag h1 ao arquivo index.html usando o nano"
[master (root-commit) 3e26466] adicionado uma tag h1 ao arquivo index.html usando o nano
 2 files changed, 1 insertion(+)
 create mode 100644 index.html
 create mode 100644 listadeexercicios.docx

```

Dois arquivos alterados e um inserido

Veja o comentário.

Como faço para excluir um arquivo?

```
romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ ls
index.html  listadeexercicios.docx  '~$stadeexercicios.docx'

romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ git rm --cached listadeexercicios.docx
rm 'listadeexercicios.docx'

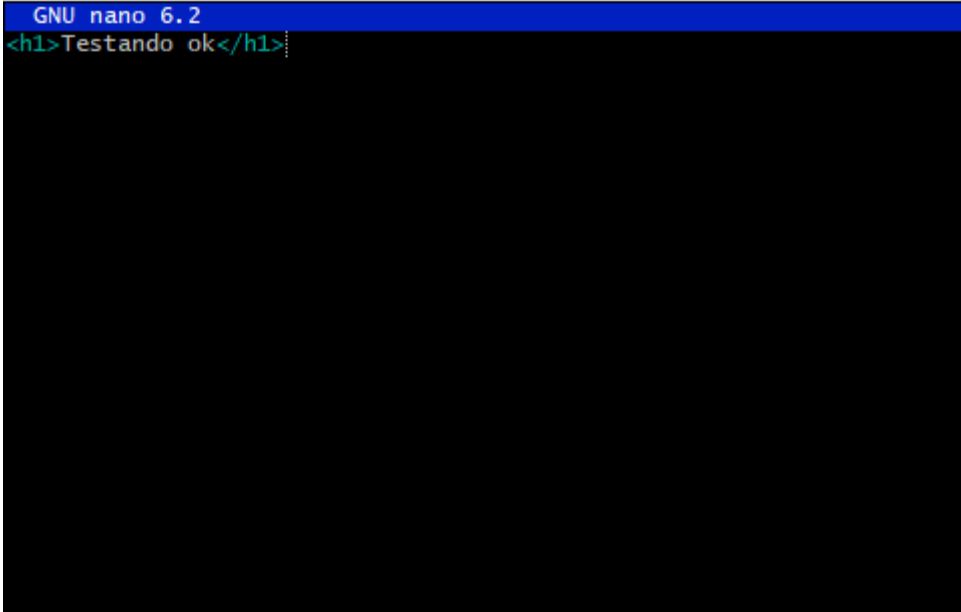
romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ git commit -m "exclui o arquivo listadeexercicios.docx"
[master 50a3983] exclui o arquivo listadeexercicios.docx
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 listadeexercicios.docx

romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ ls
index.html  listadeexercicios.docx  '~$stadeexercicios.docx'

romul@romulo MINGW64 ~/desktop/cursogithub (master)
$
```

Questão 05- Como desfazer um commit?

Crie um novo arquivo



```
GNU nano 6.2
<h1>Testando ok</h1>
```



```
romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ ls
index.html  index2.html

romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ ..
```

Abra o arquivo index.html e faça uma modificação

```
GNU nano 6.2
<h1>Testando HTML </h1>
<h2>Criando um novo título</h2>...
```

No git status ele diz:

```
romul@romulo MINGW64 ~/desktop/cursogithub (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:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index2.html

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

romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ ..
```

Index.html – trackeado e modificado

Index2.html – não trackeado

```
romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ git add index.html index2.html

romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ ..
```

Faça o commit para alterar os dois arquivos

```
romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ git commit -m "alterando dois arquivos"
[master 2126c1d] alterando dois arquivos
2 files changed, 2 insertions(+)
create mode 100644 index2.html
```

```
romul@romulo MINGW64 ~/desktop/cursogithub (master)
$ .....
```