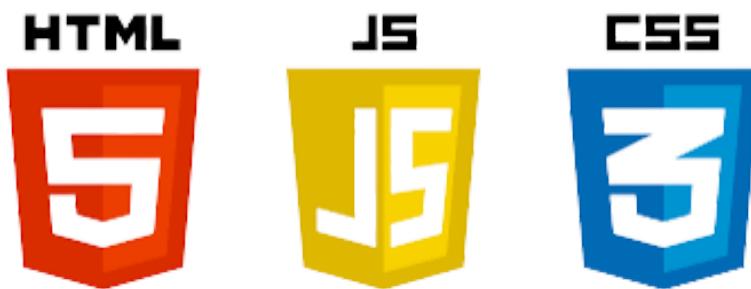


# Curso: Front-End do Bem



## AULA 11

- O que é o Git?
- Instalação do Git
- O que é o GitHub
- Instalação do GitHub
- Exemplo Prático do GitHub
- Sugestões de Prática para essa aula
  - Links das 10 aulas anteriores
- O que vamos aprender na aula 12



/igor-rebolla

# O que é o Git?

O Git é um sistema de versionamento (gratuito) que armazena e grava mudanças feitas em nossos arquivos de forma local(repositório local) ou seja é como se fosse um depósito na sua obra onde você guarda alguns materiais para que a obra possa ser concluída com aquilo que você tem armazenado localmente(repositório local) sem a necessidade de ir buscar qualquer material no depósito externamente.

Agora que fizemos esse paralelo com a obra, vamos entender como isso funciona com os nossos arquivos e pastas no computador.

Na primeira aula desse curso foi sugerido a criação de uma pasta(local) a qual chamamos de (Curso-FrontEnd) e dentro dessa pasta criamos os arquivos (index.html e o style.css) e nas aulas seguintes esses 2 arquivos passariam por mudanças devido aos conteúdos novos que aprendemos.

Na obra ao guardarmos todo o material localmente, nós corremos o risco de perder uma parte desse material, por exemplo: 5 sacos de cimento que estavam armazenados em um canto no terreno (choveu e ventou bastante) e acabou molhando o cimento, o qual resultou na perda do material.

Agora você fez um site para um cliente, deixou todos os arquivos desse site (que estava quase pronto para ser entregue) armazenados localmente no seu computador e por alguma razão o computador "deu pane"... resultando na perda de todos os arquivos e não tinha sido feito backup.

Aliás tinha sido feito backup sim, localmente na máquina e salvo em pastas da seguinte maneira:



# O que é o Git?

1. Versão 01;
2. Versão 02;
3. Versão Agora vai;
4. Versão uhuuuu Falta pouco;
5. Versão Final para o cliente.

E por estarem armazenadas em repositório local (todas essas versões foram perdidas)

Por isso o recomendado é guardar versões anteriores dos seus projetos para caso você precise voltar e fazer uma alteração específica na Versão 02 por exemplo. Só que notem, mesmo em um exemplo com "apenas" 5 pastas diferentes acima, fica um pouco complicado organizar essas versões e para isso podemos instalar no nosso computador um carinha chamado "Gerenciador de Versões", o Gerenciador de Versões possibilita que para cada versão que fizermos (e essas versões sofrem alterações) eu falo: - 'Amigão Gerenciador de Versões, isso aqui é uma nova versão' e ele gentilmente responde: "OK, versão nova guardada/armazenada".

Estou na aula 02 desse curso, criei os exemplos dessa aula respectiva, fui para a aula 03 estava lá todo "empolgadão" criando os exemplos dessa aula 03 e lembrei "Não curti algo que foi feito na aula 02" agora que o Gerenciador de Versões está instalado, eu posso falar para ele "Gerenciador de Versões, eu preciso alterar o arquivo da Aula 02" como ele guarda/armazena essas versões você pode acessar essa versão e fazer as alterações necessárias.

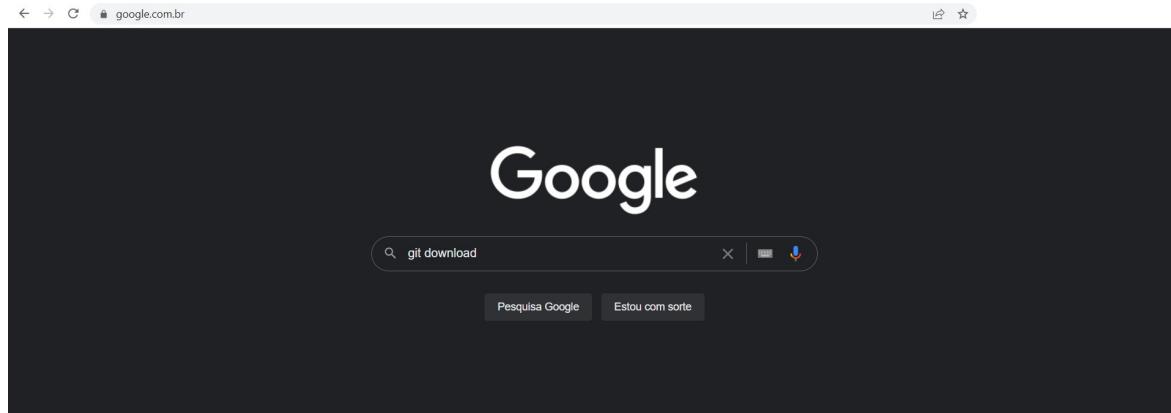
Só reforçando o nome desse Software que faz esse Gerenciamento de Versões é o GIT.

E bora ver como instalamos o GIT.



# Instalação do Git

Passo 1 – Na caixa de pesquisa do Google, vamos digitar git download



Passo 2 – Clicar no primeiro link mostrado com base na pesquisa do Passo 1

Google

git download

Todas Videos Imagens Notícias Livros Mais Ferramentas

Aproximadamente 348.000.000 resultados (0,29 segundos)

<https://git-scm.com/downloads> Traduzir esta página **Passo 2**

**Downloads - Git SCM**

Latest source Release. 2.35.1 Release Notes (2022-01-29) [Download](#) for Linux. GUI Clients.

Git comes with built-in GUI tools (`git-gui`, `gitk`), but there are ...

[Download for Linux and Unix - GUI Clients · Logos](#)

<https://git-scm.com/download/win> Traduzir esta página

**Downloading Package - Git SCM**

Click here to download the latest (2.35.1) 32-bit version of Git for Windows.

<https://git-scm.com/Começando-Instalando-o-Git>

**Instalando o Git**

Para mais opções de instruções de como instalar o **Git** em outros vários sistemas Unix, veja na página do **Git**, em <http://git-scm.com/download/linux>.



/igor-rebolla

# Instalação do Git

Passo 3 - Como eu estou usando o Windows (vou clicar dentro da Caixa: Downloads - Windows)

The screenshot shows the official Git website at [git-scm.com/downloads](https://git-scm.com/downloads). The 'Downloads' section is highlighted with a red box. Inside this box, the 'Windows' option is also highlighted with a red box. Below the box, the text 'Passo 3' is visible. To the right, there's a monitor icon displaying the latest source release version 2.35.1 and a 'Download for Windows' button.

Passo 4 - Aqui eu posso escolher entre as versões do Windows (32-bit e 64-bit) - vou instalar a versão 64-bit

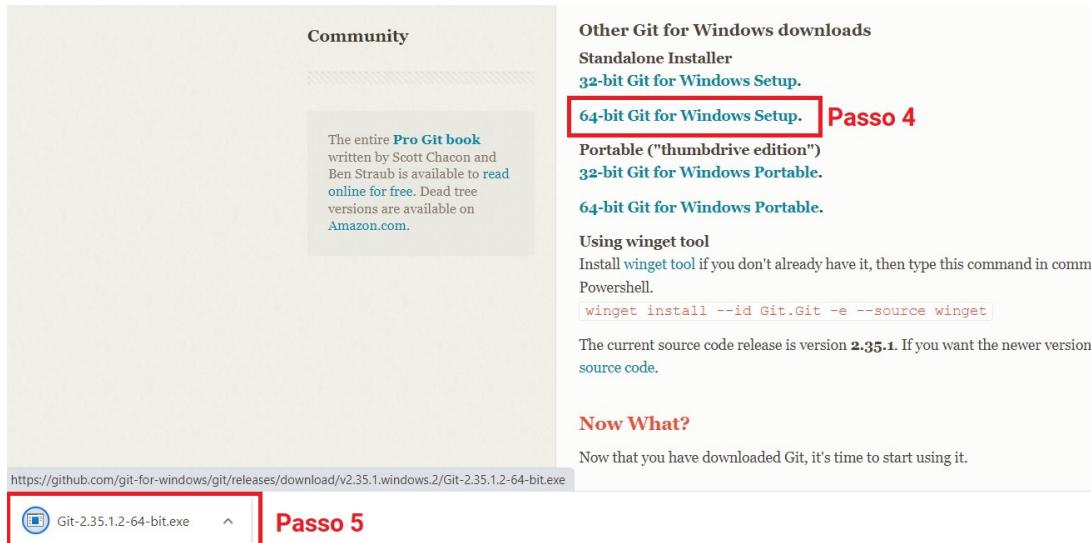
The screenshot shows the 'Download for Windows' section of the Git website. A red box highlights the '64-bit Git for Windows Setup.' link under 'Other Git for Windows downloads'. To the right, a large block of text in red reads: 'Passo 4 - Lembrando que 64-bit é a versão a qual eu estou instalando, caso a sua versão seja 32-bit é só escolher essa versão nas opções ao lado'.



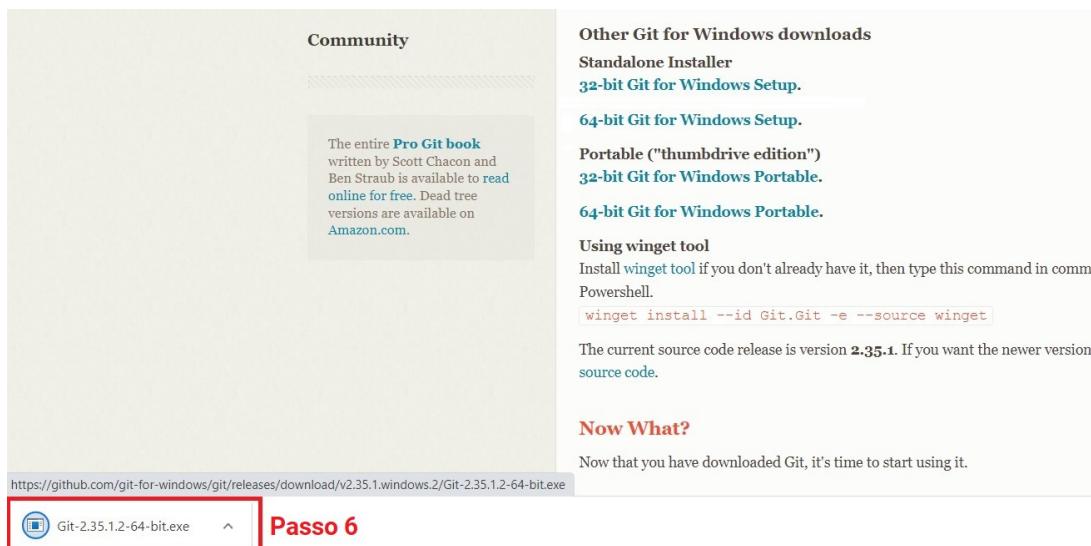
/igor-rebolla

# Instalação do Git

Passo 5 – Ao clicar no passo 4, o download da Versão escolhida do Git vai começar automaticamente.



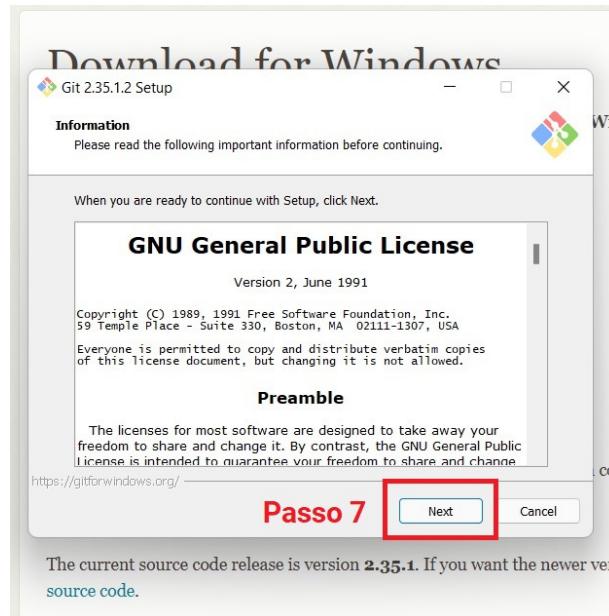
Passo 6 – Download concluído (Clicar 2x sobre o mesmo) e bora seguir os passos da instalação.



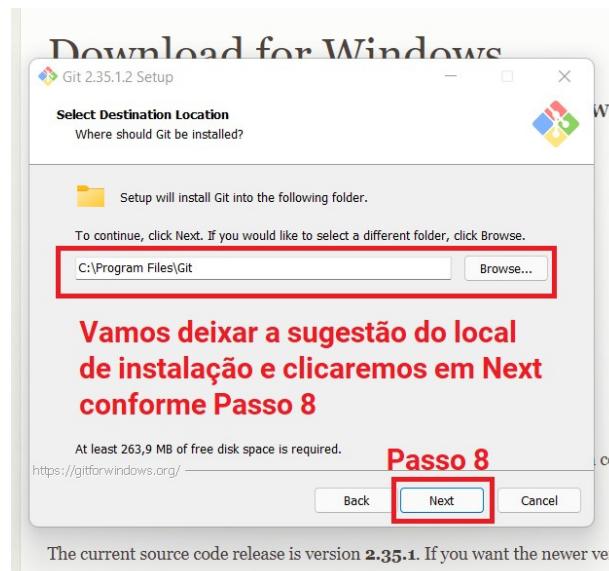
/igor-rebolla

# Instalação do Git

Passo 7 - A primeira caixa que será mostrada é sobre a Licença (Vamos clicar em Next)



Passo 8 - Aqui vamos escolher o local onde o GIT será instalado no Windows (Vamos deixar a sugestão e clicar em next)



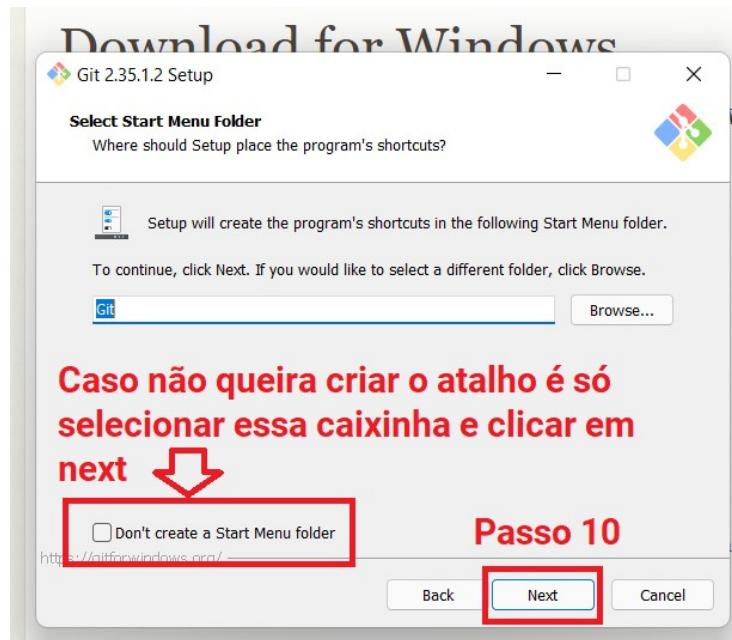
/igor-rebolla

# Instalação do Git

Passo 9 – Essa parte da instalação (permite acrescentar ou tirar alguns componentes), vamos deixar a sugestão e clicar clicar em Next.



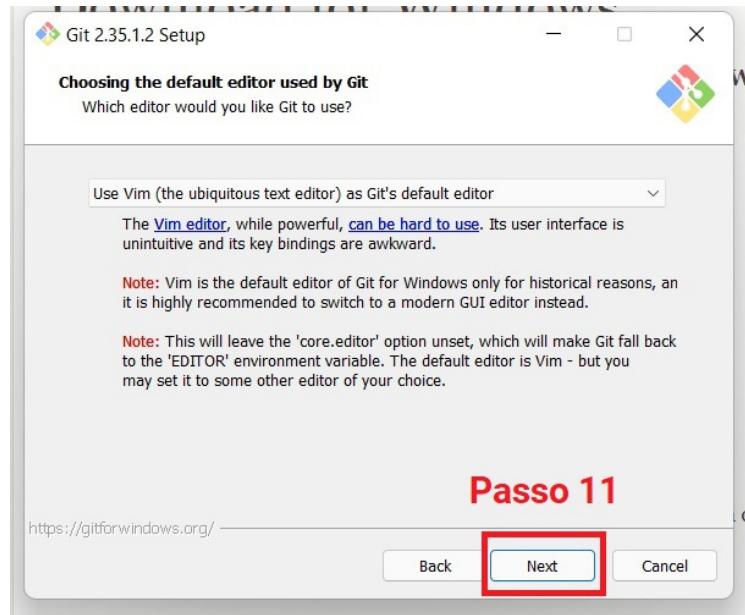
Passo 10 – Permite adicionar ou não um atalho no Menu Iniciar (Vamos adicionar esse atalho)



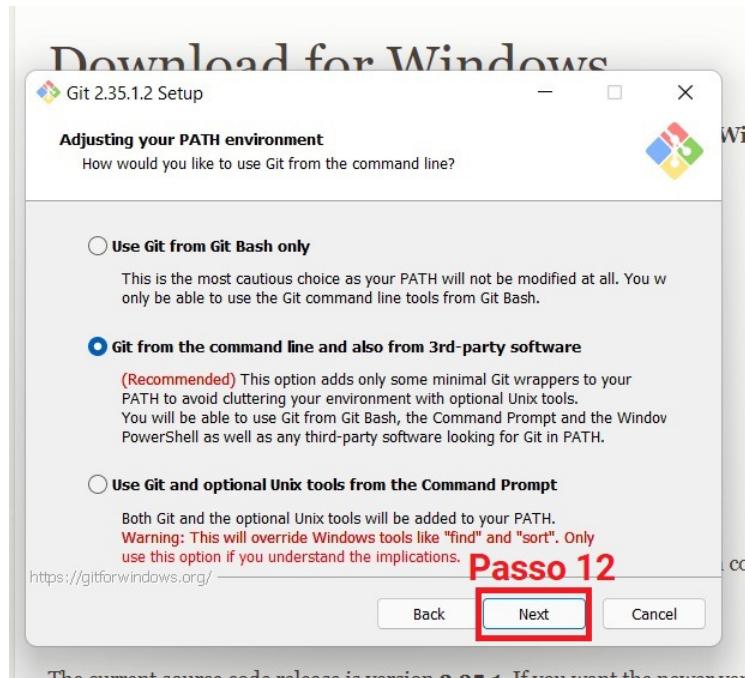
/igor-rebolla

# Instalação do Git

Passo 11 – Está informando qual editor que o Git vai usar (Bora clicar em Next)



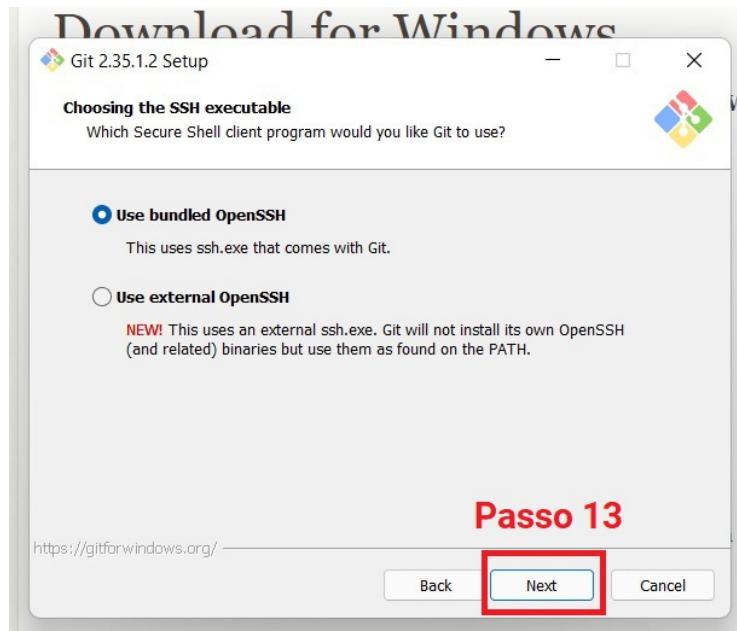
Passo 12 – Vamos seguir a recomendação da instalação e deixaremos selecionada a opção do meio e clicaremos em Next



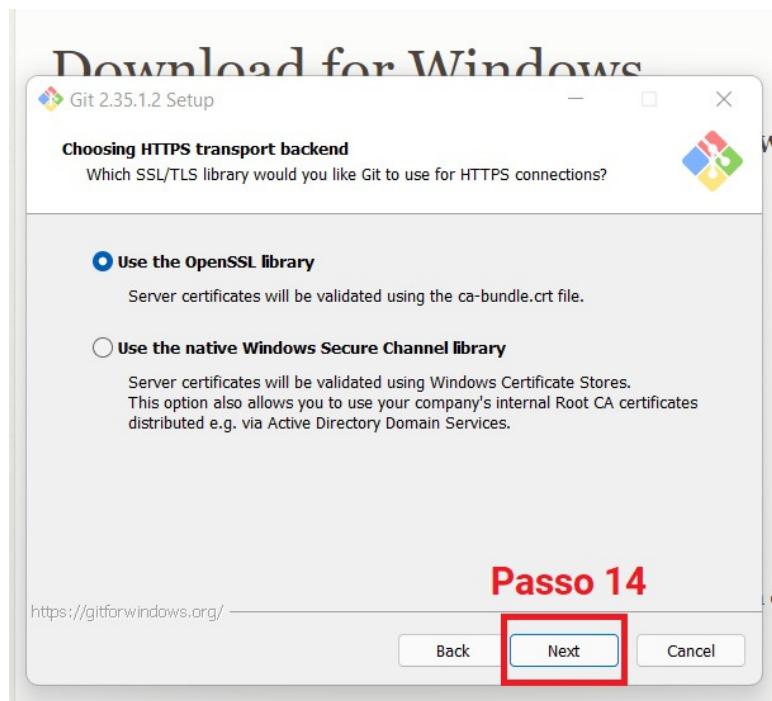
/igor-rebolla

# Instalação do Git

Passo 13 - Vamos seguir a recomendação da instalação e deixaremos selecionada a primeira opção e clicaremos em Next



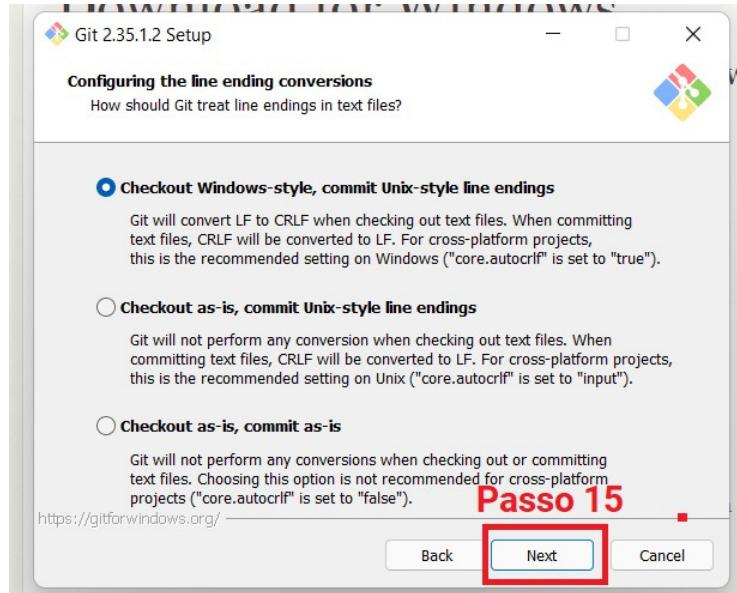
Passo 14 - Vamos seguir a recomendação da instalação e deixaremos selecionada a primeira opção e clicaremos em Next



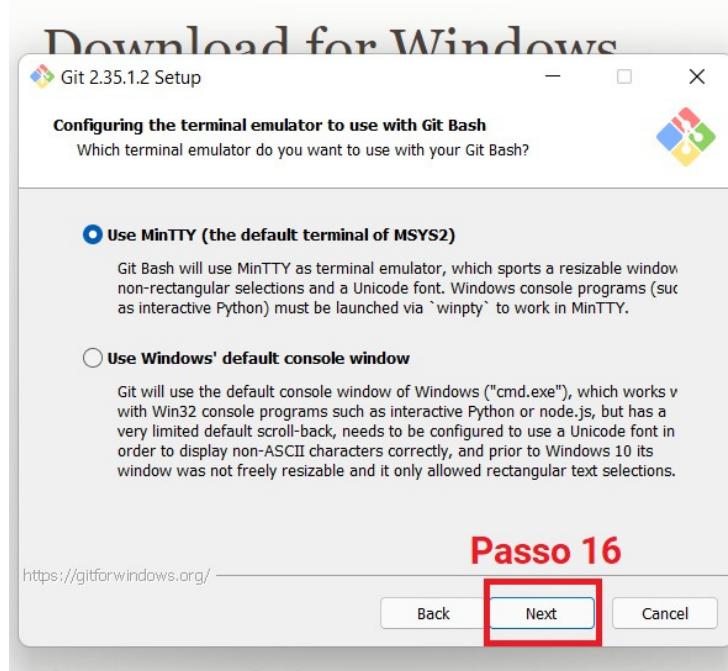
/igor-rebolla

# Instalação do Git

Passo 15 – Vamos seguir a recomendação da instalação e deixaremos selecionada a primeira opção e clicaremos em Next



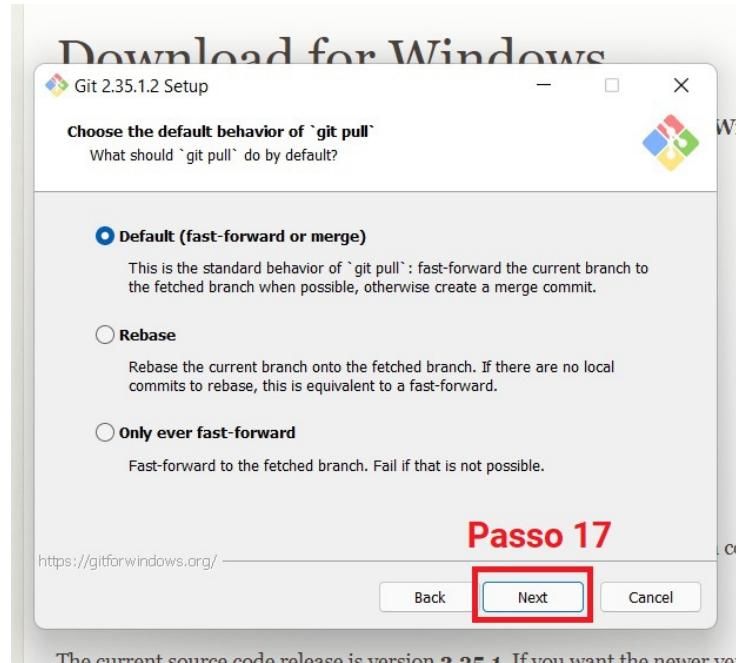
Passo 16 – Vamos seguir a recomendação da instalação e deixaremos selecionada a primeira opção e clicaremos em Next



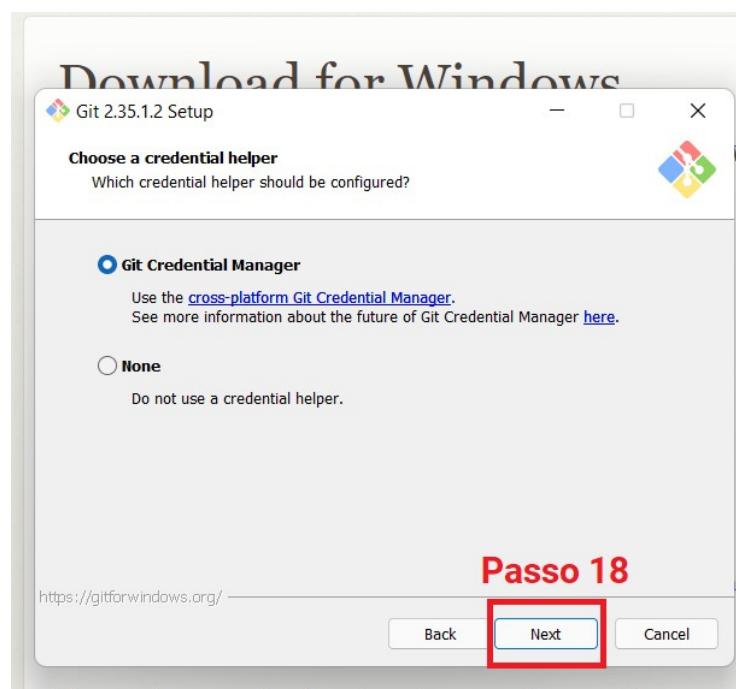
/igor-rebolla

# Instalação do Git

Passo 17 – Vamos seguir a recomendação da instalação e deixaremos selecionada a primeira opção e clicaremos em next



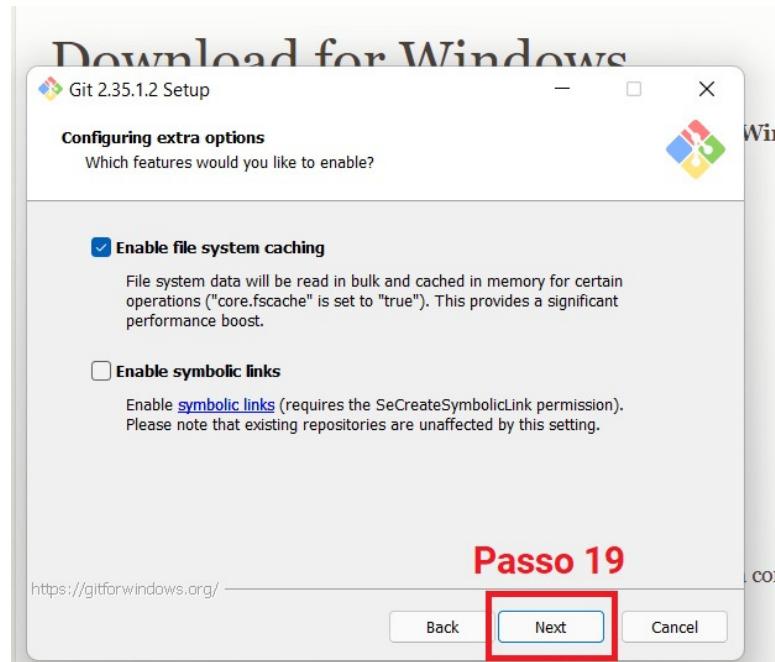
Passo 18 – Vamos seguir a recomendação da instalação e deixaremos selecionada a primeira opção e clicaremos em Next



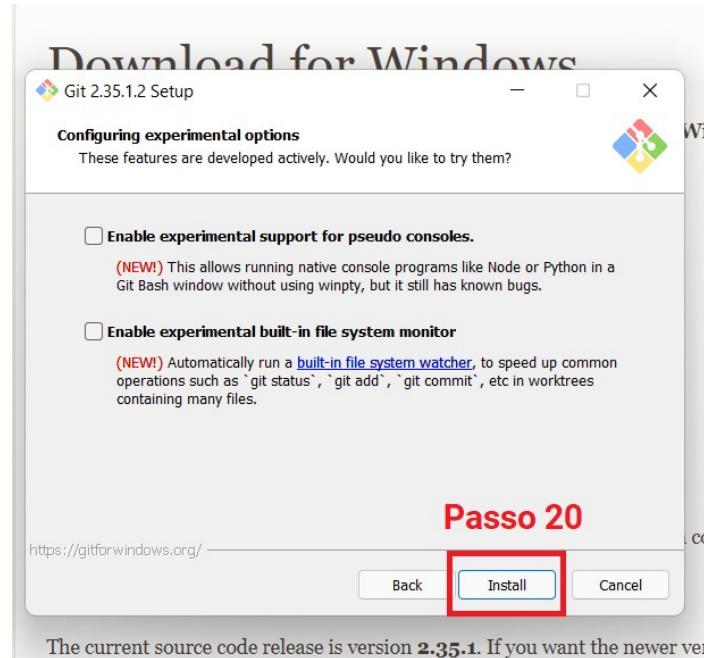
/igor-rebolla

# Instalação do Git

Passo 19 – Vamos seguir a recomendação da instalação e deixaremos selecionada a primeira opção e clicaremos em Next



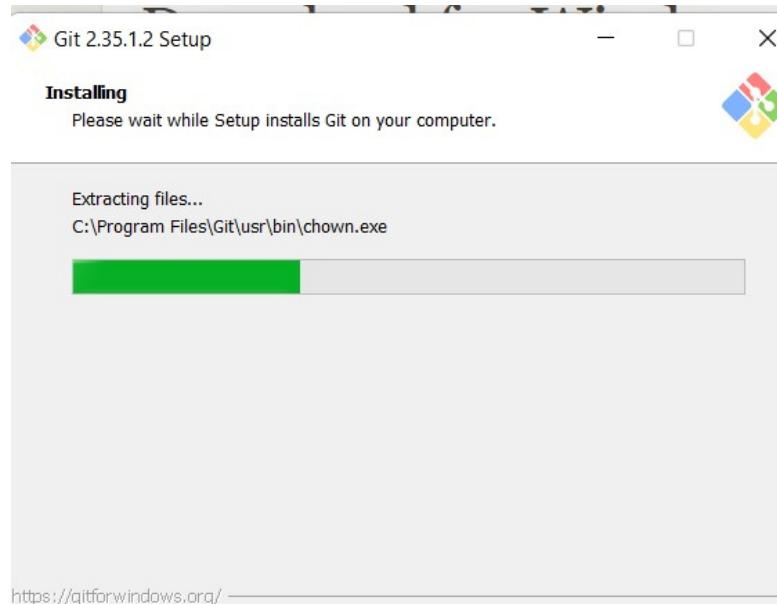
Passo 20 – Aqui não vou selecionar nenhuma opção e vou clicar em Install.



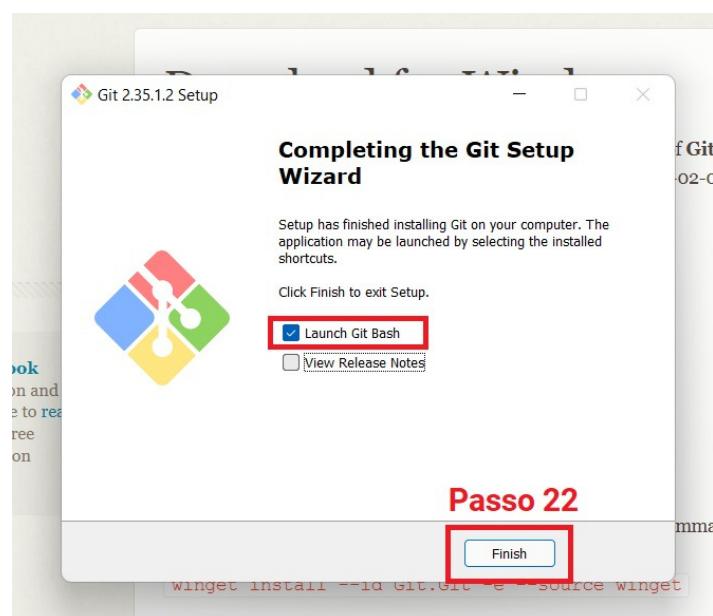
/igor-rebolla

# Instalação do Git

Passo 21 – Vamos aguardar o processo de instalação do GIT



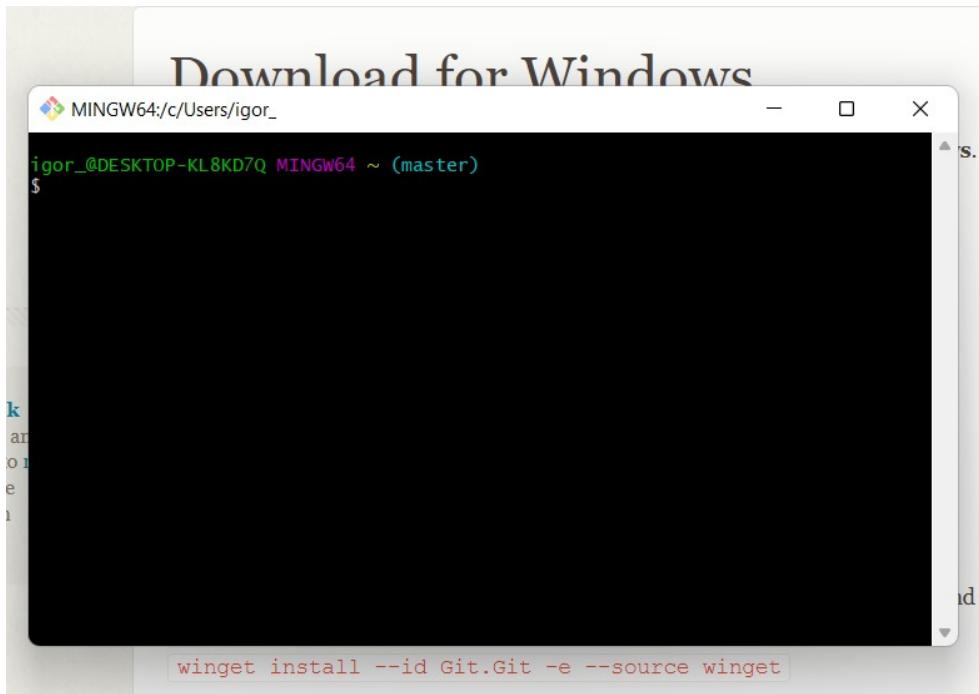
Passo 22 – Instalação Concluida vamos selecionar a opção "Launch Git Bash" e clicaremos em Finish. O Git Bash é um terminal (o qual iremos utilizar aqui nessa aula) do Git. Ele é semelhante ao bom e velho prompt de comando do Windows.



/igor-rebolla

# Instalação do Git

Passo 23 - Quando selecionamos a opção "Launch Git Bash" no passo 22  
(Irá abrir uma tela como essa abaixo)



A screenshot of a Windows terminal window titled "Download for Windows". The window has a title bar and a close button. Inside, there's a black terminal window with a white border. The terminal shows a command-line interface with the following text:

```
MINGW64:/c/Users/igor_
igor_@DESKTOP-KL8KD7Q MINGW64 ~ (master)
$ k
an
on
e
1
winget install --id Git.Git -e --source winget
```

The terminal window is centered on the screen.



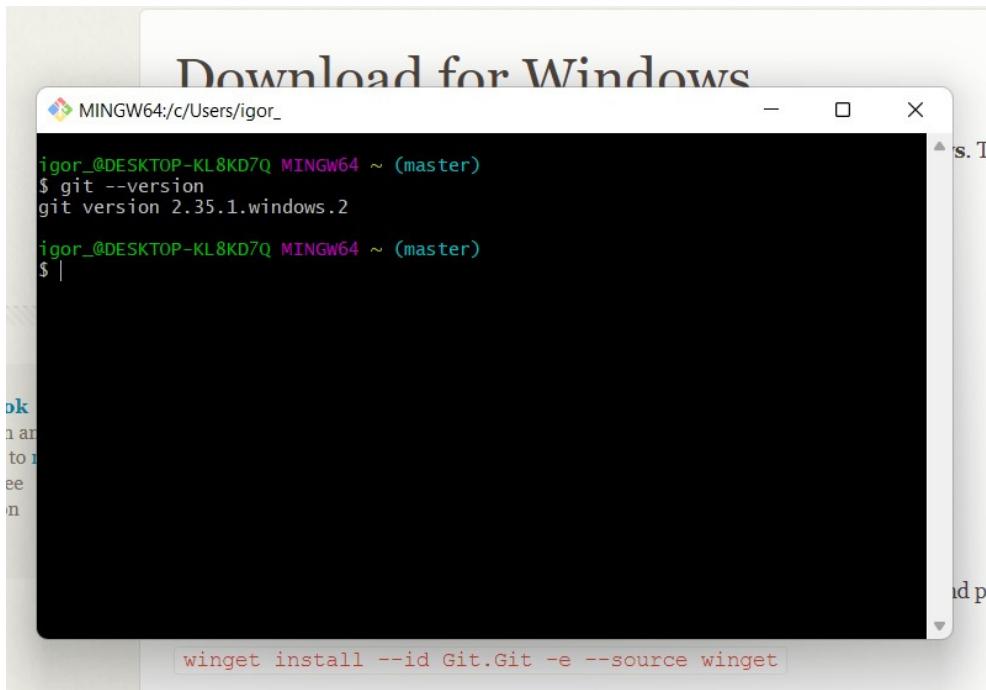
/igor-rebolla

# Instalação do Git

Passo 24 – Depois do primeiro contato com o Git Bash. O primeiro comando que iremos executar até para começarmos a nos acostumar com esses comandos é o:

```
git --version
```

Esse comando mostra qual é a versão do GIT que está instalada. Aqui a versão instalada é a: 2.35.1.windows.2 (Isso significa que nossa instalação está funcionando corretamente).



```
MINGW64:/c/Users/igor_
igor_@DESKTOP-KL8KD7Q MINGW64 ~ (master)
$ git --version
git version 2.35.1.windows.2

igor_@DESKTOP-KL8KD7Q MINGW64 ~ (master)
$ |
```

The screenshot shows a terminal window titled "Download for Windows". The command \$ git --version is run, and the output "git version 2.35.1.windows.2" is displayed. The terminal window has a light gray background and a black text area. The title bar includes the path "MINGW64:/c/Users/igor\_" and the user name "igor\_". The bottom of the window shows a red status message: "winget install --id Git.Git -e --source winget".



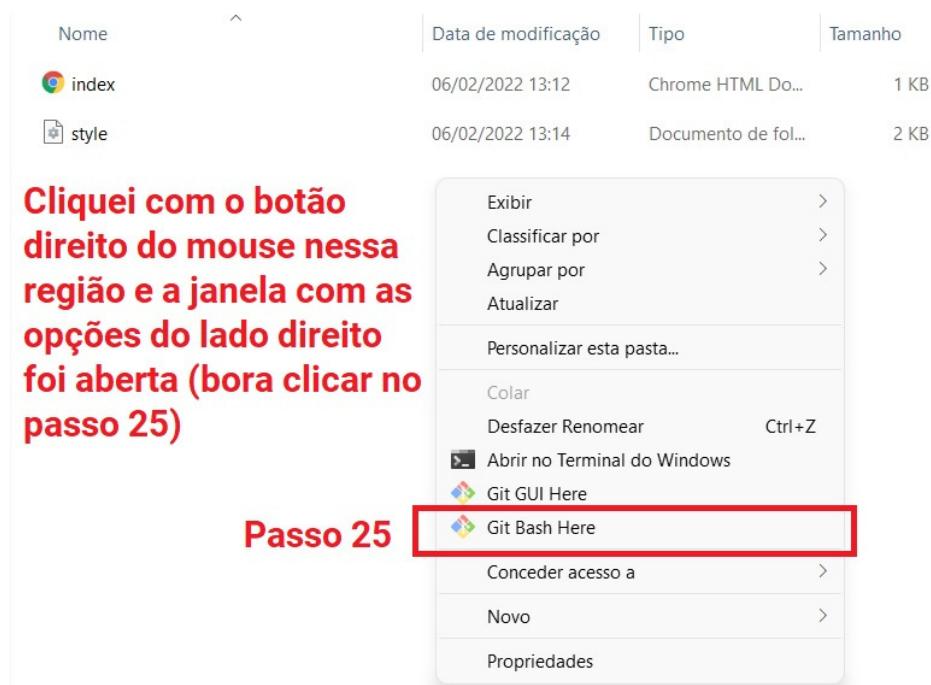
/igor-rebolla

# Instalação do Git

Passo 25 – Vamos procurar o local onde a pasta Curso-FronEnd foi criada.

A minha foi criada no D:\Curso-FrontEnd (vou clicar 2x nessa pasta e aqui é esperado que 2 arquivos tenham sido criados: 1. index.html e 2. style.css).

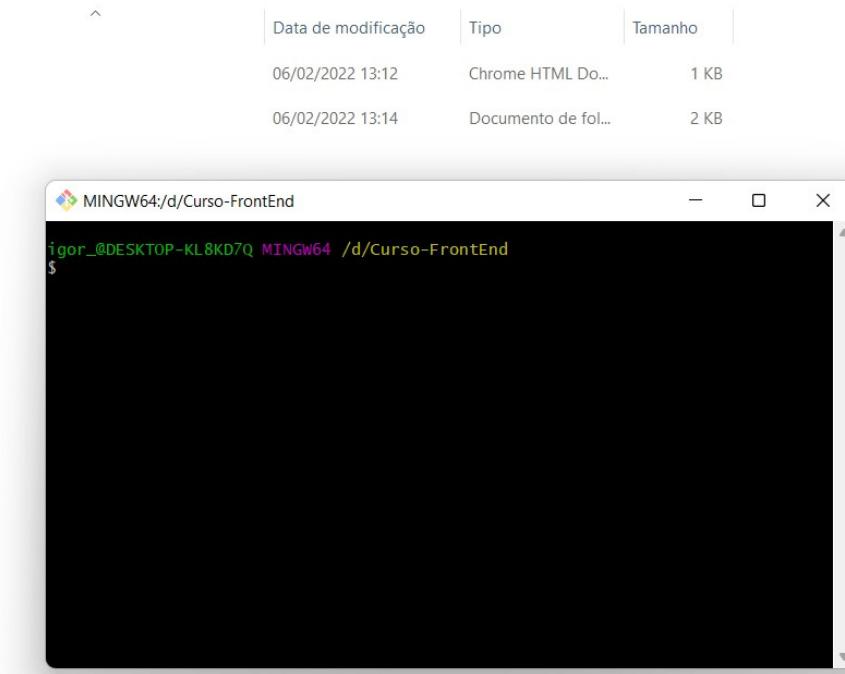
Clicaremos com o botão direito do mouse e podemos ver o Git Bash Here (bora clicar nele)



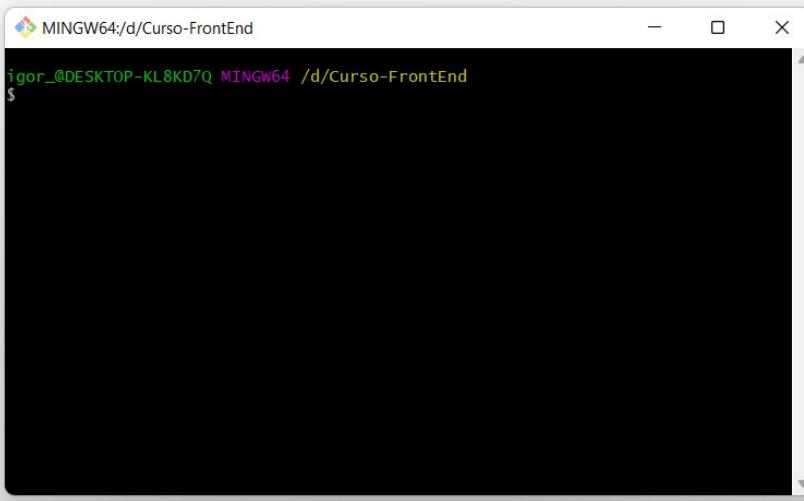
/igor-rebolla

# Instalação do Git

Passo 26 - Com o Git Bash aberto dentro da pasta Curso-FrontEnd essa tela irá surgir e notem que agora está mostrando:  
/d/Curso-FrontEnd pra gente (está mostrando exatamente onde estamos) que é dentro da pasta Curso-FrontEnd.



Nome	Data de modificação	Tipo	Tamanho
index	06/02/2022 13:12	Chrome HTML Do...	1 KB
style	06/02/2022 13:14	Documento de fol...	2 KB

```
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd$
```



# Instalação do Git

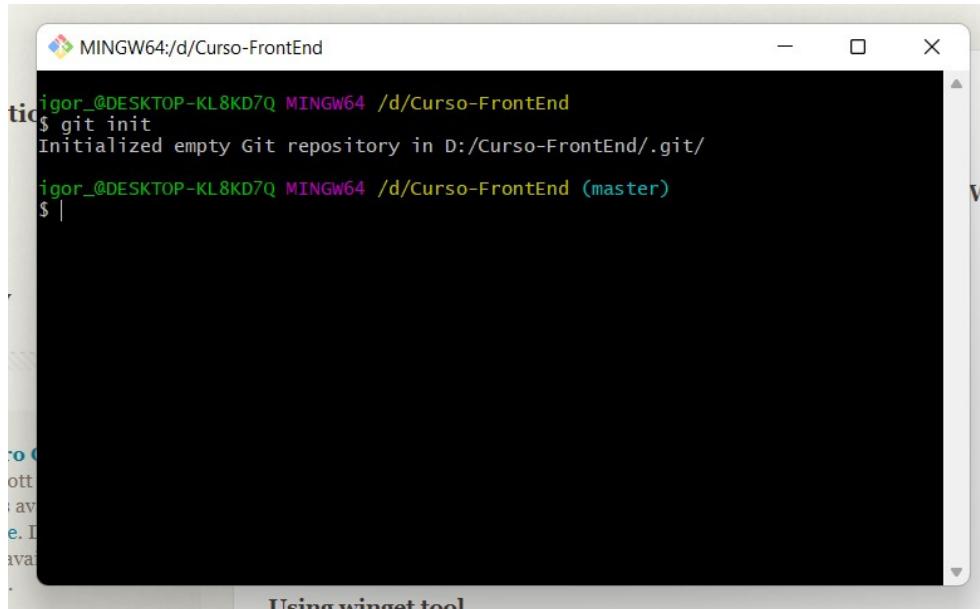
Passo 27 - Agora vamos ver outro comando que é o git init (Esse comando mostra pra gente que iniciamos um repositório git local vazio (empty)).

git init

Ah já ia quase me esquecendo notem que depois que o comando git init foi dado, apareceu um tal de (master).

E qual é a função desse carinha?

É nos mostrar que estamos dentro da (branch master) ou seja da raiz master que quando criamos o repositório ela é criada de forma automática.



```
MINGW64:/d/Curso-FrontEnd
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd
$ git init
Initialized empty Git repository in D:/Curso-FrontEnd/.git/
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (master)
$ |
```



/igor-rebolla

# Instalação do Git

Passo 28 - O git init (inicializou pra gente um repositório vazio) só que na nossa pasta Curso-FrontEnd (temos 2 arquivos dentro dela) o index.html e o style.css. Vamos ver agora como adicionamos esses arquivos dentro do nosso repositório. Esse comando é o:

```
git add
```

O git add ele vai mandar os nossos 2 arquivos (index.html e o style.css) para uma especie de "bastidor"... "Bastidor Igor, o que é isso?"

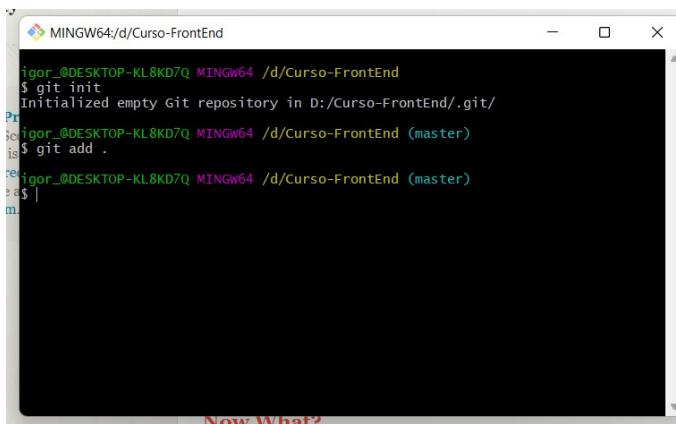
Os arquivos vão ficar aguardando sua vez de serem chamados para entrar em cena (assim como em um espetáculo).

O git add vai convocar esses arquivos que estavam lá aguardando e colocará nessa área de bastidor (o qual você vai ficar aguardando até ser "empurrado" vai fazer sentido já já essa palavra "empurrado") para a parte principal.

Aqui como eu vou adicionar 2 arquivos, vou usar o:

```
git add .
```

O .(ponto) aqui permite que eu adicione todos os arquivos ao mesmo tempo. Caso eu queira adicionar só um arquivo é só usar: git add index.html e só o index.html será adicionado o style.css não.



```
MINGW64/d/Curso-FrontEnd
$ git init
Initialized empty Git repository in D:/Curso-FrontEnd/.git/
$ git add .
$ git commit -m "Initial commit"
[master (root-commit) 123456] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 index.html
$
```



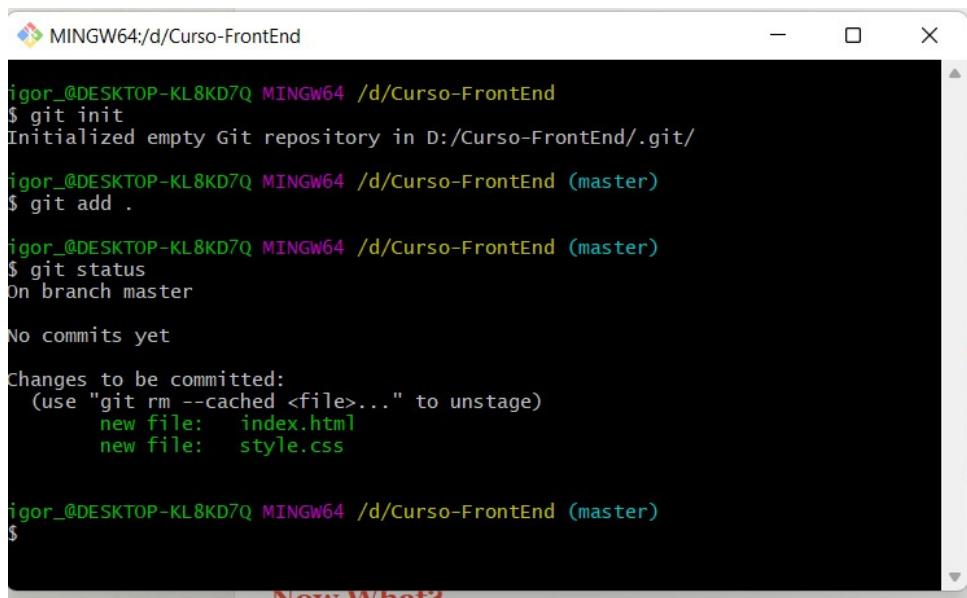
/igor-rebolla

# Instalação do Git

Passo 29 - Notem que parece que nada aconteceu, foi dado o git add . e ele simplesmente foi para a linha da nossa pasta. Mas aconteceu sim e aqui vamos usar um novo comando que é o:

```
git status
```

Esse comando mostra os arquivos que foram adicionados naquele "bastidor" pelo git add (e os quais ainda não foram comitados).



```
MINGW64:/d/Curso-FrontEnd
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd
$ git init
Initialized empty Git repository in D:/Curso-FrontEnd/.git/
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (master)
$ git add .

igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (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:   style.css

igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (master)
$
```



/igor-rebolla

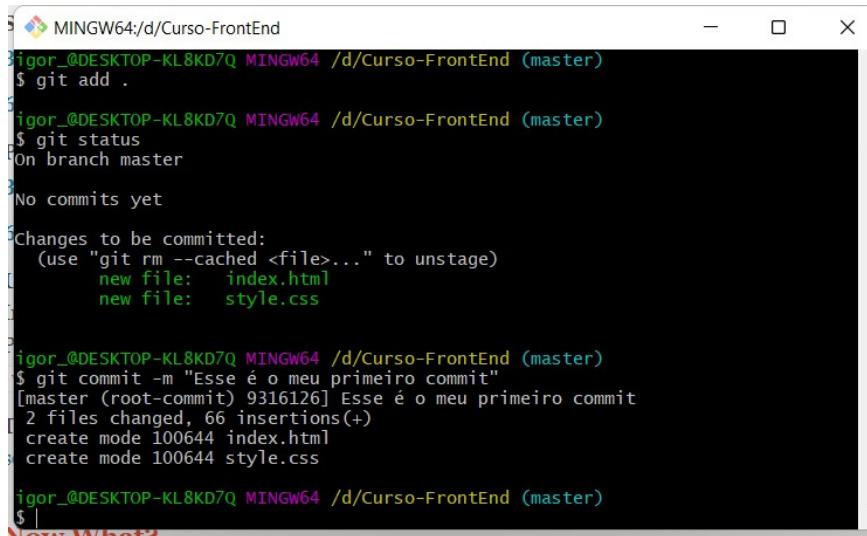
# Instalação do Git

passo 30 - O comando que usaremos para fazer o commit é o:

```
git commit -m "Esse é o meu primeiro commit"
```

Os commits são as versões dos nossos arquivos.

Dado esse comando e conforme imagem abaixo que nos mostra que conseguimos fazer o primeiro commit (primeira versão) em nosso repositório.



A screenshot of a terminal window titled 'MINGW64:/d/Curso-FrontEnd'. The session shows the following commands and output:

```
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (master)
$ git add .
.
.
$ 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:   style.css
.
.
.
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (master)
$ git commit -m "Esse é o meu primeiro commit"
[master (root-commit) 9316126] Esse é o meu primeiro commit
  2 files changed, 66 insertions(+)
  create mode 100644 index.html
  create mode 100644 style.css
.
.
.
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (master)
$ | Now What?
```

Obs.: O git alterou a nomenclatura da branch principal que vimos em alguns passos anteriores que é chamada de master para main.

"Então porque você não mostrou o main direto aqui?"

Algumas empresas ainda estão utilizando o master... e foi por isso que eu mostrei o main antes. Como esse curso é a base... o ideal é mostrar tudo gradativo (com base no que acontece no mercado de trabalho).

O comando para alterar a branch de master para main é o:

```
git branch -M "main"
```



# Instalação do Git

Vamos ver como fica essa alteração dentro do Git Bash:

```
MINGW64:/d/Curso-FrontEnd
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (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:  style.css

igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (master)
$ git commit -m "Esse é o meu primeiro commit"
[master (root-commit) 9316126] Esse é o meu primeiro commit
 2 files changed, 66 insertions(+)
 create mode 100644 index.html
 create mode 100644 style.css

igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (master)
$ git branch -M "main"
$
```

Now What?

Agora mesmo com o Git instalado na máquina (estou com as minhas versões todas "gerenciadas") e por mais que os arquivos estejam todos versionados (eles continuam armazenados em um repositório local). E se eu quiser:

- mostrar esses arquivos que eu estou estudando para mais pessoas;
- entregar todos os arquivos ou ir mostrando as versões durante o seu desenvolvimento para o meu cliente;
- deixar salvo em algum lugar(repositório remoto) para caso meu computador tenha dado problema (eu consiga buscar essas versões)



# O que é o GitHub?

Para isso tem o Github que é um repositório remoto que permite armazenar seus arquivos e assim (salvar, compartilhar....) tudo de forma segura.

O GitHub permite também que conforme vamos armazenando nossos projetos, seja de (estudo, clientes...) vamos gerando um portfólio para quando formos nos candidatar para alguma vaga de emprego (o recrutador vai pedir o seu link do github) para verificar o que você desenvolveu até aquele momento.

Outra coisa importante sobre o GitHub é que ele permite que trabalhemos em equipe (Vou deixar a sementinha plantada na cabeça de vocês) sobre esse tema.

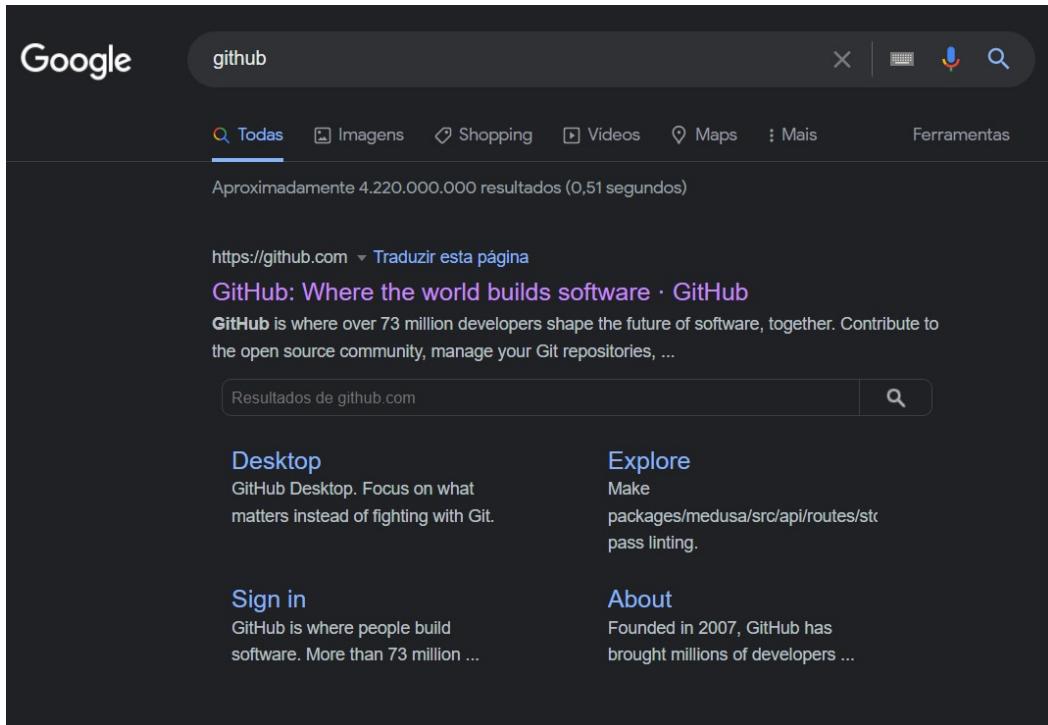
**Resumindo: O GIT vai armazenar versões dos arquivos dentro do computador (repositório local) e o GitHub vai fazer uma cópia desses arquivos/pastas na nuvem (repositório remoto).**

Bora ver como é feito a instalação do GitHub...



# Instalação do GitHub

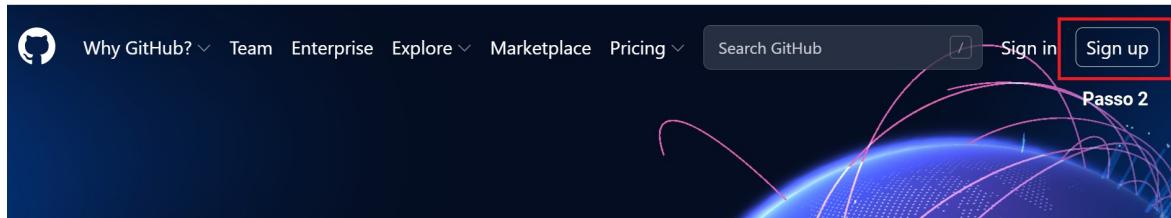
Passo 1 - Dentro do buscador do Google, vamos digitar github e enter para confirmar. E logo em seguida vamos clicar no primeiro link da busca - <https://github.com>



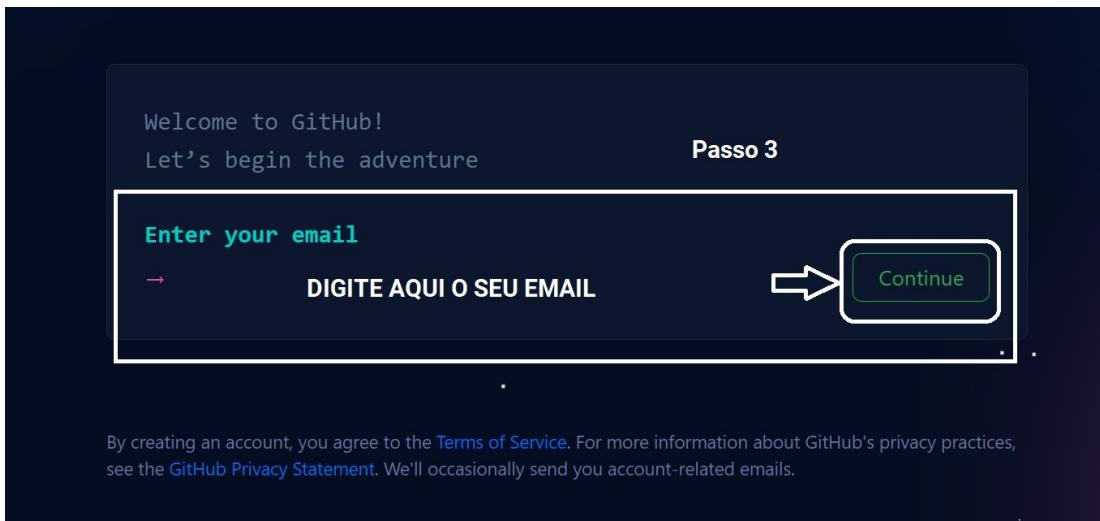
/igor-rebolla

# Instalação do GitHub

Passo 2 - Ao clicarmos em <https://github.com> – o site do github será aberto e vamos clicar em Sign up



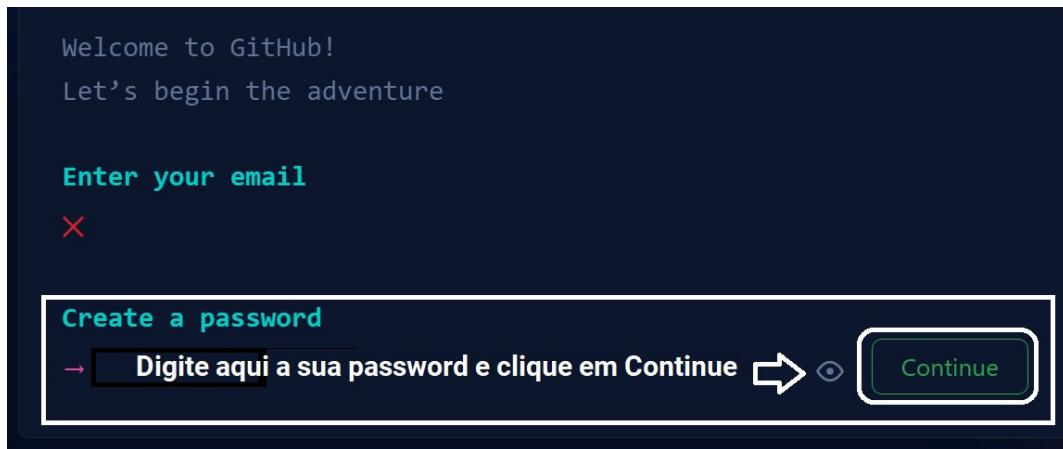
Passo 3 – Vamos digitar um email válido e clicaremos em Continue:



/igor-rebolla

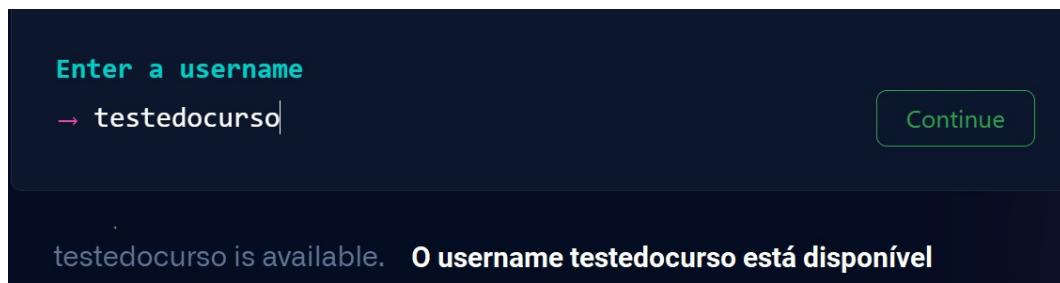
# Instalação do GitHub

Passo 4 - Vamos digitar uma senha (password) válida e clicaremos em Continue



Passo 5 - Vamos digitar um username válido para sermos identificados dentro do GitHub e clicaremos em Continue.

Aqui eu escolhi o username (testedocurso) e ele está disponível(available).



# Instalação do GitHub

Passo 6 - Aqui ele está perguntando "Se gostaríamos de receber atualizações e anúncios por e-mail?" - Eu optei por Não (Digitei a letra n) e cliquei em Continue.



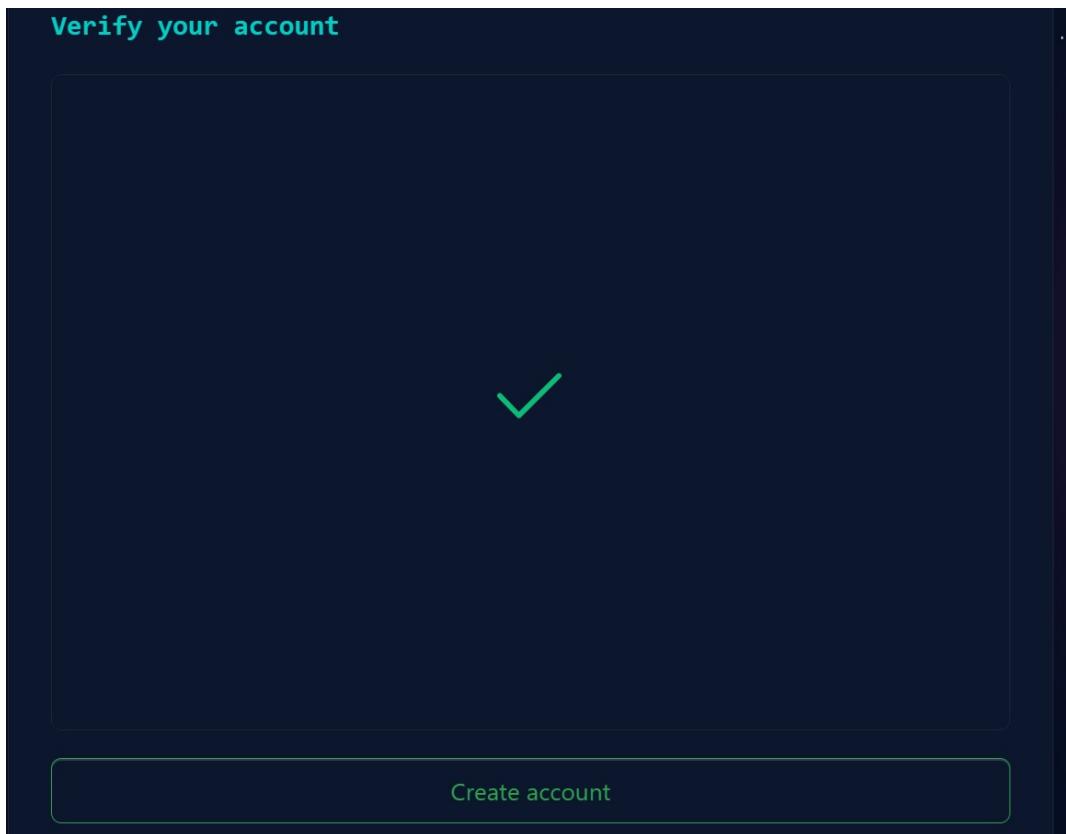
Passo 7 - Aqui ele vai pedir para você resolver um desafio (Escolha a galáxia espiral) - 5 imagens da galáxia espiral apareceram aqui para que eu resolvesse.



/igor-rebolla

# Instalação do GitHub

Passo 8 – Depois que as 5 imagens foram resolvidas, aparece um: Selo de validação verde no meio da tela e o botão Create account fica habilitado e é nesse botão que iremos clicar para criarmos nossa conta com as informações passadas até o momento dentro do GitHub.



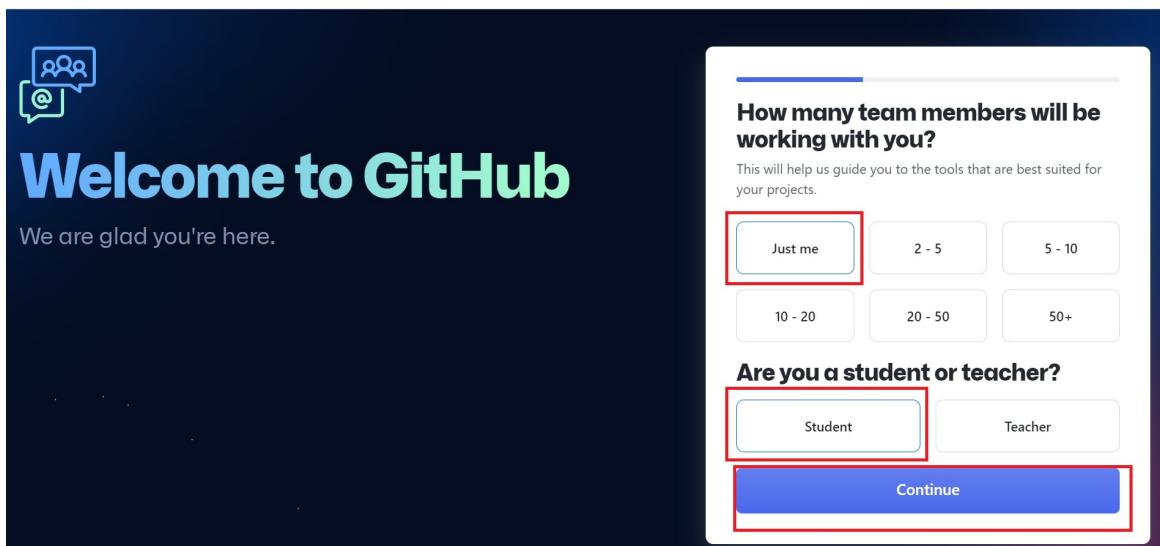
# Instalação do GitHub

Passo 9 – O GitHub enviou um código de segurança para o email cadastrado no passo 3 da sua instalação. Basta ir até esse email pegar o código e inseri-lo aqui. Peço desculpas, não dei o print da tela nesse passo.

Ao digitar os 8 dígitos do código ele automaticamente vai para a tela com 2 perguntas:

- Quantas membros do time vão trabalhar contigo?
- Você é um estudante ou professor?

Vou selecionar (Just me) e (Student) e clicarei em Continue

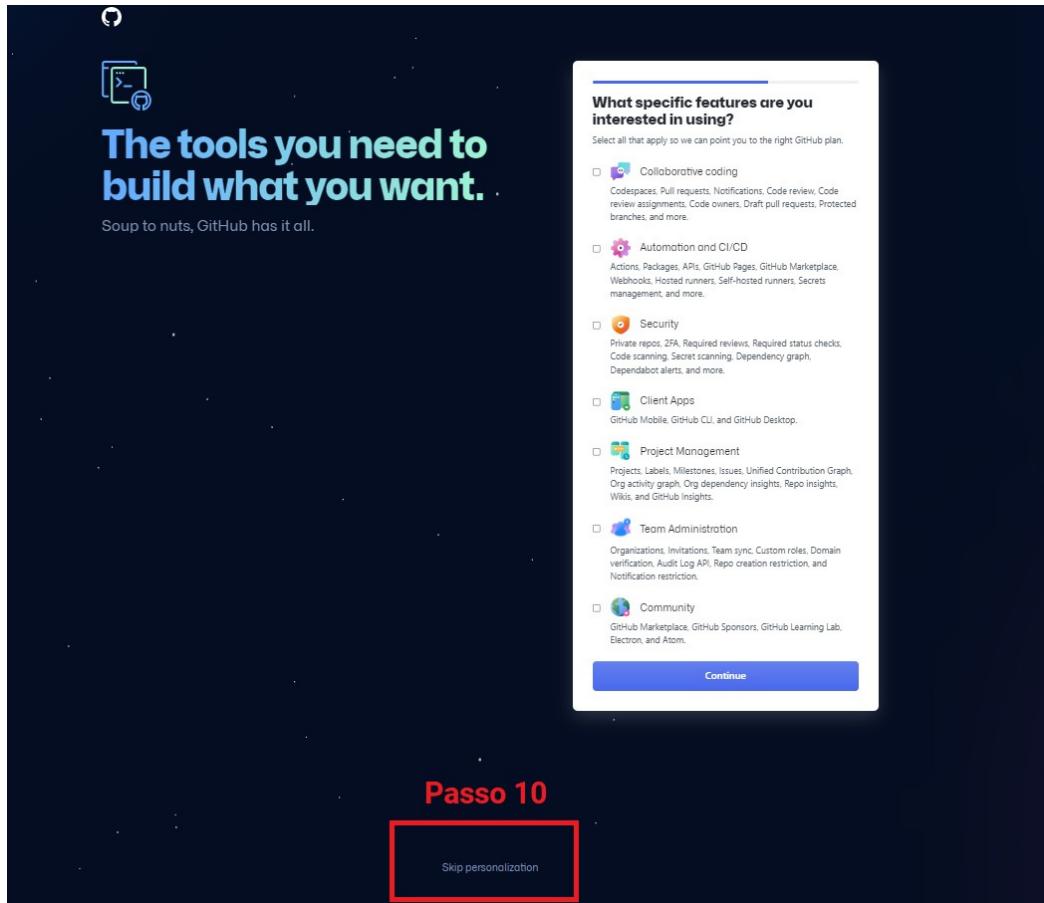


/igor-rebolla

# Instalação do GitHub

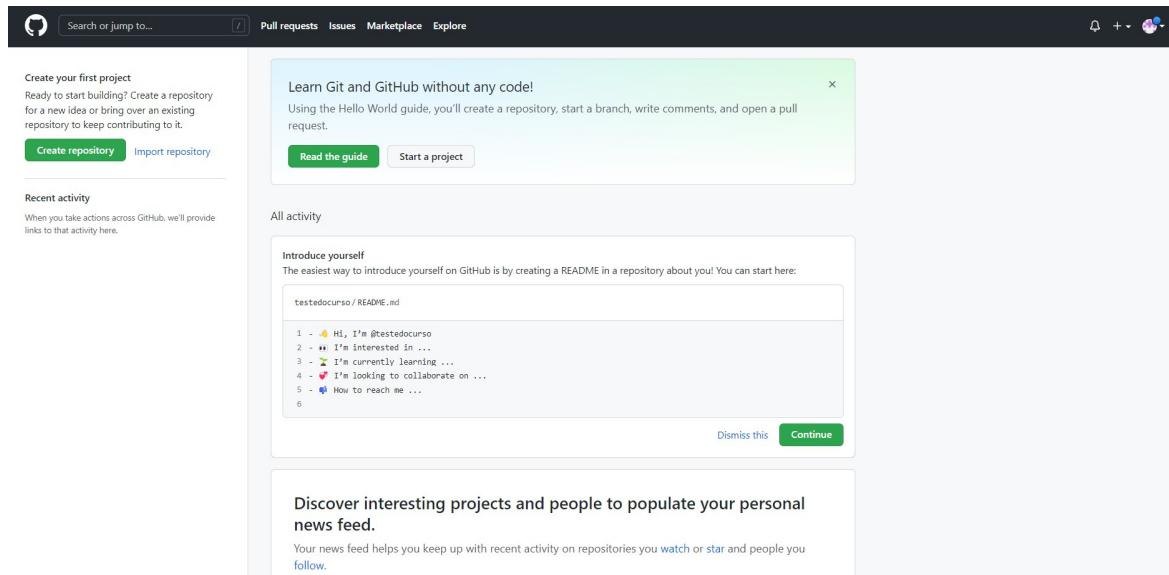
Passo 10 - Aqui ele está perguntando: "Quais features específicas você está interessado em usar?"

Eu vou clicar em Skip Personalization (Pular personalização)



# Instalação do GitHub

Passo 11 - Deu tudo certo com a nossa instalação. Agora já temos o nosso Dashboard do GitHub criado.

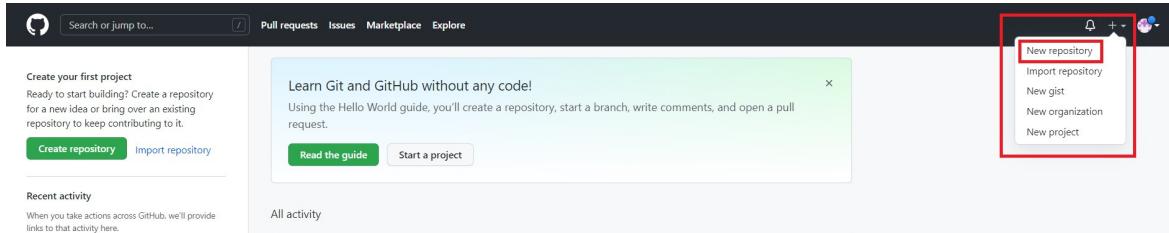


/igor-rebolla

# Exemplo Prático do GitHub

Passo 12 - Agora que o GitHub já está instalado, vamos criar o nosso primeiro repositório (para entender o seu funcionamento).

- Clicar no sinal de + no canto superior direito da tela;
- Clicar em New Repository (Novo Repositório)



/igor-rebolla

# Exemplo Prático do GitHub

Passo 13 – Aqui vamos definir o nome do nosso repositório. Notem que antes de definirmos o nome... em Owner (ele vem o nome do usuário) esse foi o nome o qual configuramos em uma das etapas da instalação do GitHub.

- Eu vou definir o nome desse repositório como Curso-FrontEnd;
- Aqui vamos deixar como Public (público) o acesso a esse repositório;
- Vou criar em Create repository

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 \*

 **testedocurso** /   **Nome do Repositório**

Great repository names are short and memorable. Need inspiration? How about refactored-train?

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.

Vamos deixar como Public mesmo.

Initialize this repository with:

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

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

**Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)

**Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

**Create repository**  E vamos clicar em Create Repository



/igor-rebolla

# Exemplo Prático do GitHub

Passo 14 - Aqui ele mostra o nosso repositório criado e alguns comandos que já foram utilizados quando vimos o GIT (nessa mesma aula).

## git init

iniciamos um repositório vazio(empty)

## git add README.md

aqui a diferença é que ele está adicionando apenas o README.md e no nosso exemplo usamos o git add . (pois adicionamos todos os arquivos disponíveis - index.html e o style.css)

## git commit -m "first commit"

fizemos o nosso primeiro commit

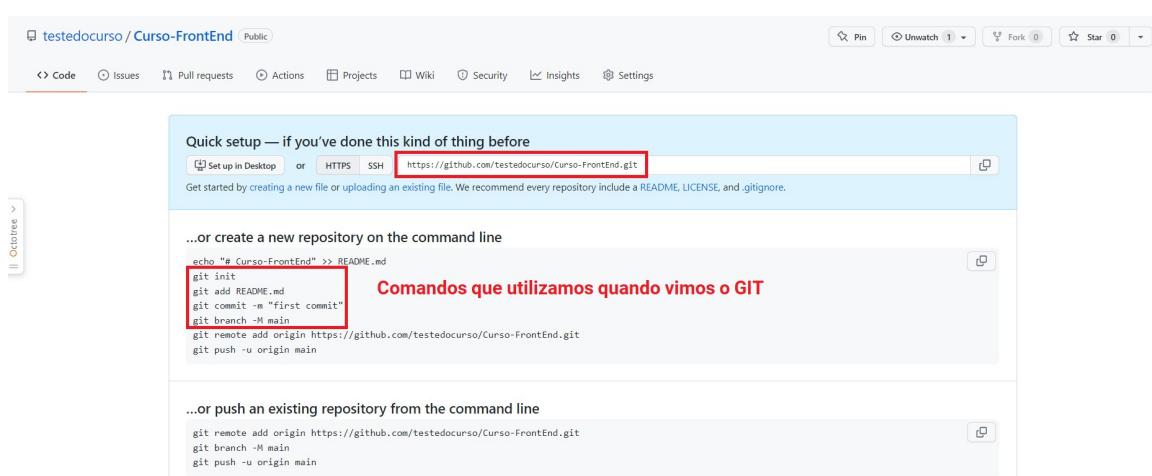
## git branch -M main

aqui mudamos o nome da nossa branch de master para main

Ainda não vimos esses 2 comandos abaixo:

git remote add origin https://github.com/testedocurso/Curso-FrontEnd.git

git push -u origin main



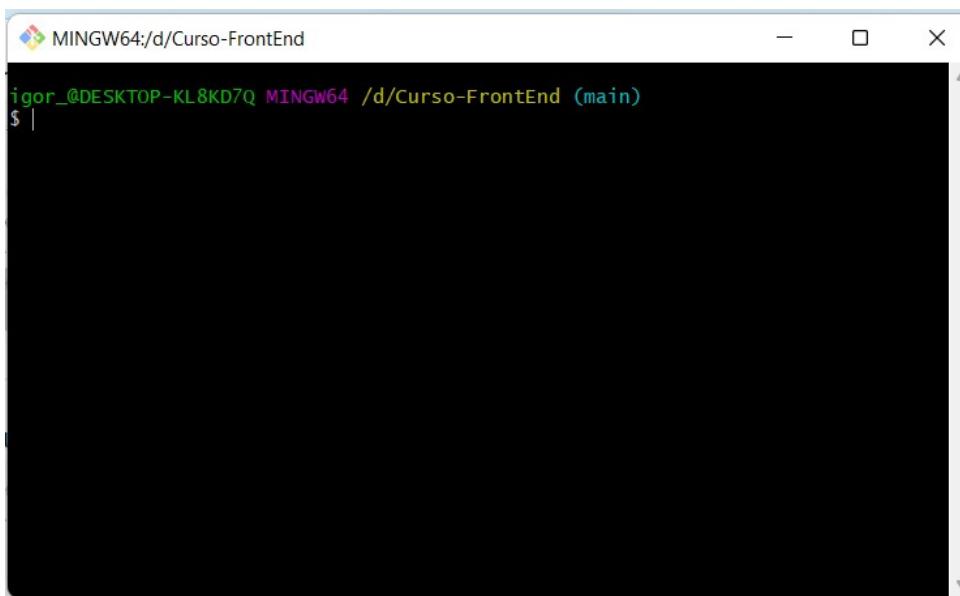
/igor-rebolla

# Exemplo Prático do GitHub

Passo 15 – Vamos abrir o Git Bash, só relembrando os passos:

1. Localizar onde a pasta Curso-FrontEnd foi salva dentro do computador;
2. Clicar 2x para abrir essa pasta;
3. Clicar com o botão direito do mouse e escolher a opção Git Bash Here

Essa é a tela que será mostrada (Já com o main) alterado no lugar do master:



```
MINGW64:/d/Curso-FrontEnd
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (main)
$ |
```



# Exemplo Prático do GitHub

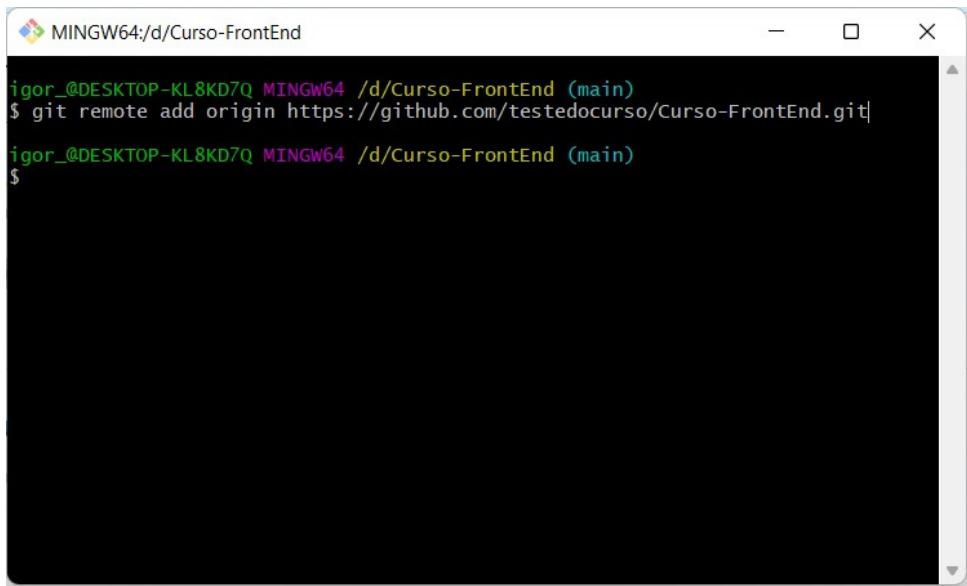
Passo 16 – Com a tela do Git Bash aberta vamos seguir com os 2 comandos que ficaram faltando fazer quando vimos na parte do Git. E o primeiro desses comandos será o:

```
git remote add origin https://github.com/testedocurso/Curso-FrontEnd.git
```

Notem que ao pressionarmos o enter parece que nada aconteceu.

E qual é a função desse comando?

É criar a conexão entre o repositório local(GIT) que está em nosso computador com o repositório remoto(GitHub).



```
MINGW64:/d/Curso-FrontEnd
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (main)
$ git remote add origin https://github.com/testedocurso/Curso-FrontEnd.git
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (main)
$
```

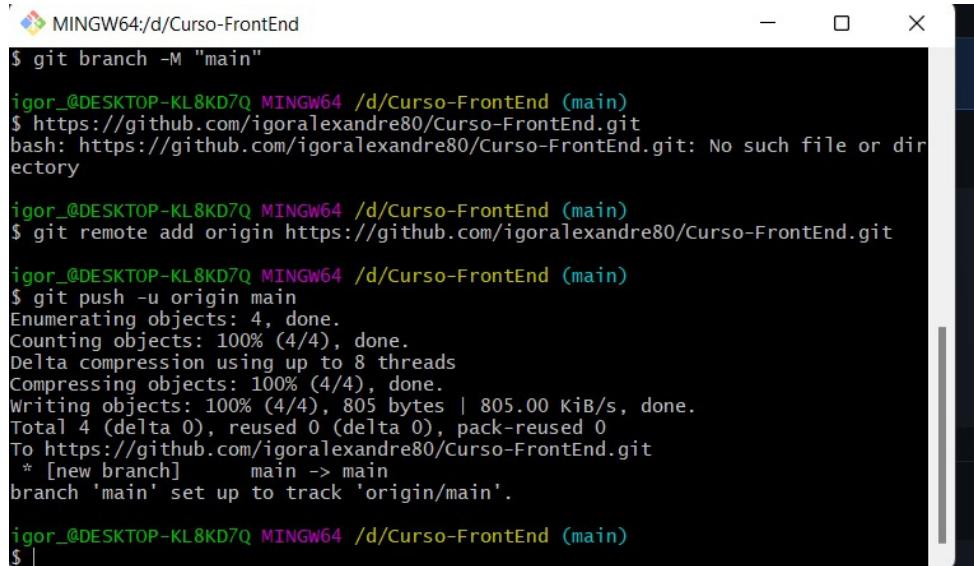


# Exemplo Prático do GitHub

Passo 17 – Só que ainda não enviamos nenhum arquivo para o nosso repositório remoto (github). Para enviarmos esses arquivos vamos usar o comando:

```
git push -u origin main
```

Esse comando praticamente empurra(push) os commits que estão no nosso repositório local(GIT) para o repositório remoto(GitHub).



```
MINGW64:/d/Curso-FrontEnd
$ git branch -M "main"
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (main)
$ https://github.com/igoralexandre80/Curso-FrontEnd.git
bash: https://github.com/igoralexandre80/Curso-FrontEnd.git: No such file or directory
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (main)
$ git remote add origin https://github.com/igoralexandre80/Curso-FrontEnd.git
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 805 bytes | 805.00 Kib/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/igoralexandre80/Curso-FrontEnd.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
igor_@DESKTOP-KL8KD7Q MINGW64 /d/Curso-FrontEnd (main)
$ |
```

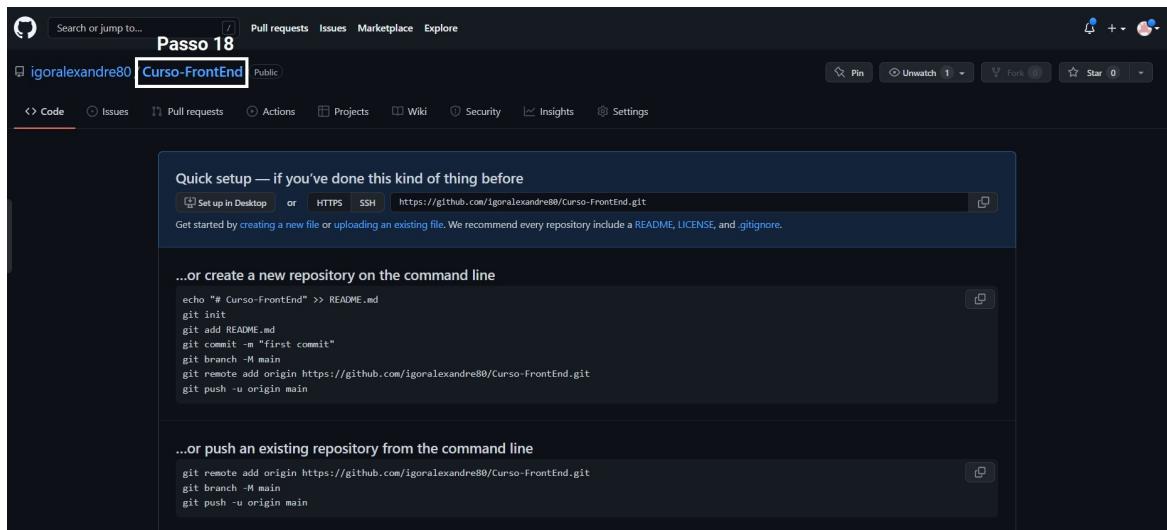


/igor-rebolla

# Exemplo Prático do GitHub

Passo 18 – Arquivos enviados com sucesso do repositório local para o repositório remoto. Vamos visualizar como esses arquivos ficaram dentro do GitHub.

Para isso só clicar em Curso-FrontEnd no canto superior esquerdo da tela conforme imagem abaixo:



# Exemplo Prático do GitHub

Passo 19 – Podemos verificar que a pasta Curso-FrontEnd foi adicionada no repositório remoto (Github) com seus 2 arquivos (index.html e o style.css). Isso permite:

- Acessar remotamente;
- Informar o portfólio para o recrutador;
- Compartilhar os arquivos com o cliente;
- Trabalhar com mais pessoas no mesmo projeto;
- Etc...

Basta acessar por esse endereço aqui:

<https://github.com/igoralexandre80/Curso-FrontEnd>

The screenshot shows a GitHub repository page. At the top, the URL 'github.com/igoralexandre80/Curso-FrontEnd' is in the address bar. The repository name 'igoralexandre80 / Curso-FrontEnd' is displayed with a 'Public' badge. Below the header, there are navigation links: Code (which is underlined), Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Under the 'Code' tab, it shows 'main' branch, '1 branch', and '0 tags'. There is a 'Go to file' button, an 'Add file' button, and a 'Code' dropdown menu. The main content area displays a single commit from 'igoralexandre80' made 13 minutes ago. The commit message is 'Esse é o meu primeiro commit'. Below the commit, there are two files listed: 'index.html' and 'style.css', both also added 13 minutes ago with the same commit message. At the bottom of the page, there is a dark blue footer bar with the text 'Help people interested in this repository understand your project by adding a README.' and a green 'Add a README' button.



/igor-rebolla

# Links das 10 aulas anteriores

Para acessar cada aula, basta clicar na aula de interesse abaixo que será direcionado diretamente para o post com a respectiva aula no LinkedIn:

[Aula 01](#)

[Aula 02](#)

[Aula 03](#)

[Aula 04](#)

[Aula 05](#)

[Aula 06](#)

[Aula 07](#)

[Aula 08](#)

[Aula 09](#)

[Aula 10](#)



# **Sugestões de prática para essa aula:**

1. Instalar o Git (Utilizando o passo a passo dessa aula);
2. Instalar o GitHub (Utilizando o passo a passo dessa aula);
3. Criar o seu primeiro repositório no Github e passar os arquivos do repositório local (GIT) para o repositório remoto (GitHub)

Programação é prática, curiosidade e repetição!!!!



# O que vamos aprender na aula 12:

Na décima segunda e próxima aula, vamos ter um exemplo "lúdico" para fixar e praticar o que foi visto até agora.

**Feedbacks dessa décima primeira aula são bem-vindos.**

Nos vemos na décima segunda aula que será disponibilizada no dia 17/02/2022 às 8h30 (Horário de Brasília) - Dentro do meu perfil no LinkedIn.

