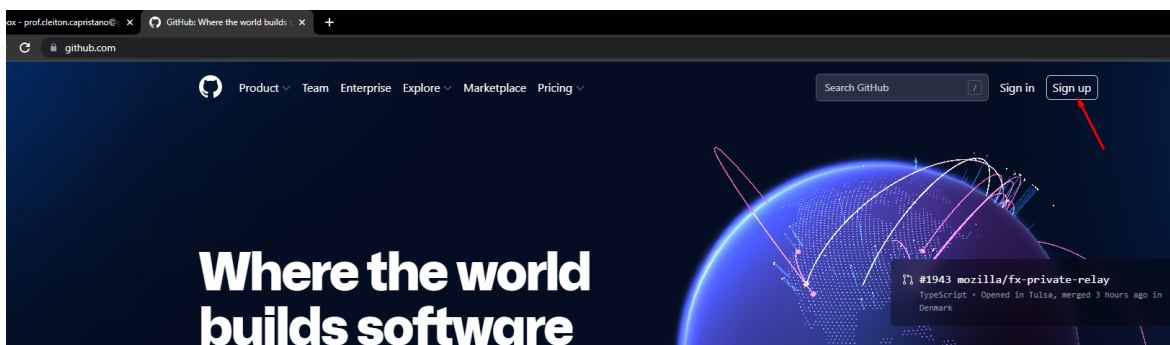


Para se cadastrar no GitHub



Welcome to GitHub!  
Let's begin the adventure

Enter your email

✓ prof.cleiton.capristano@gmail.com

Create a password

✓ ••••••••

Enter a username

✓ profcapristano

Would you like to receive product updates and announcements via email?

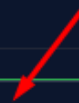
Type "y" for yes or "n" for no

✓ n

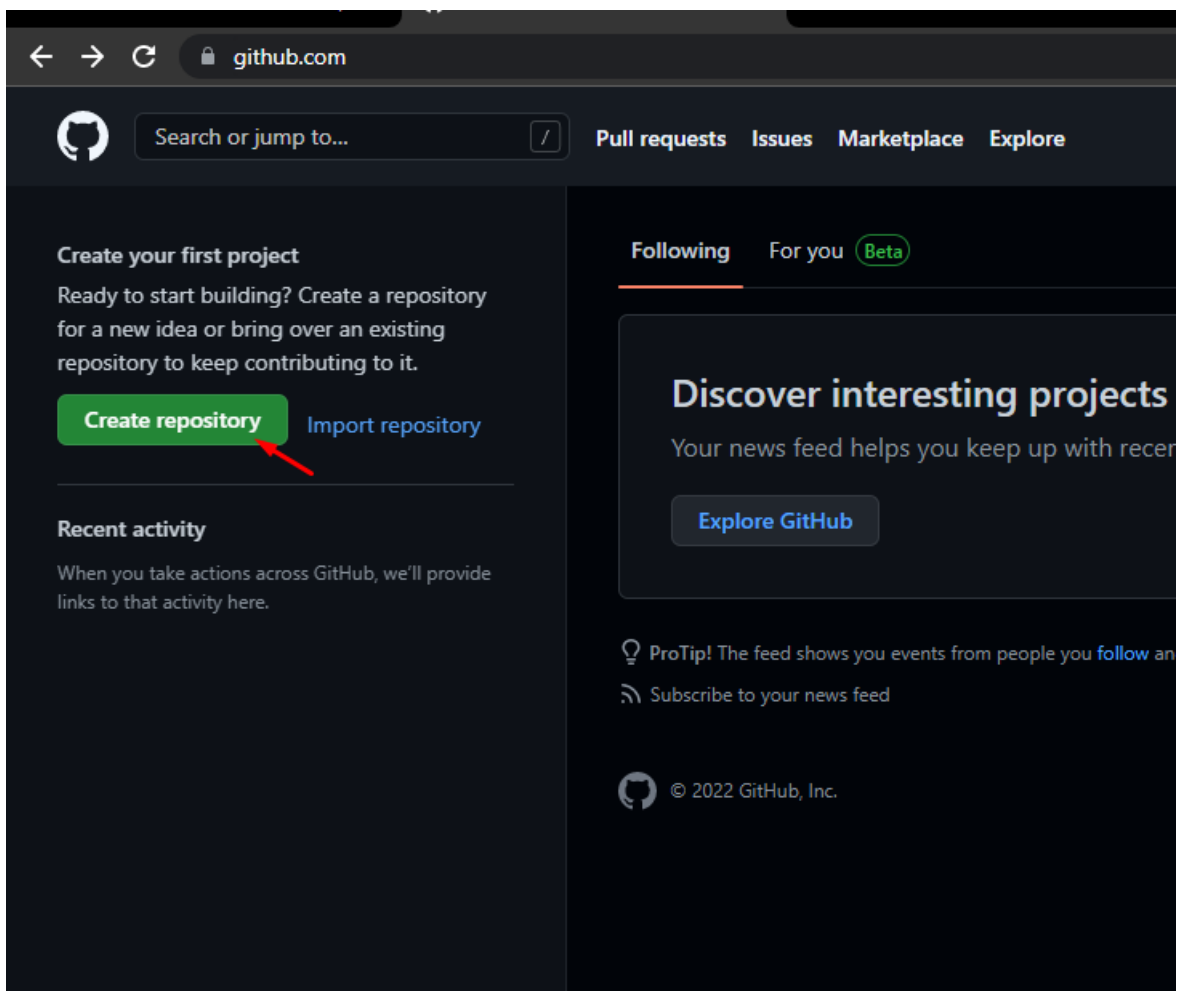
Verify your account



Create account



Para criar seu primeiro repositório:



Utilize essas configurações caso não saiba o que fazer:


## 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 \*


 profcapristano ▾

/


primeiro\_repo ✓

Great repository names are short and memorable. Need inspiration? How about **expert-system**?

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:**

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.](#)

.gitignore template: None ▾

**Choose a license**

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

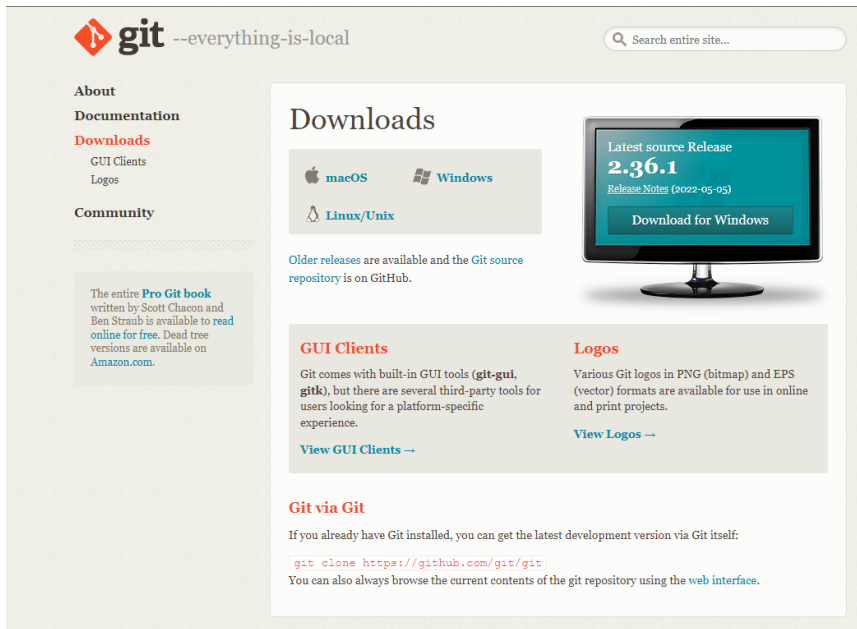
License: None ▾

 You are creating a public repository in your personal account.

Create repository

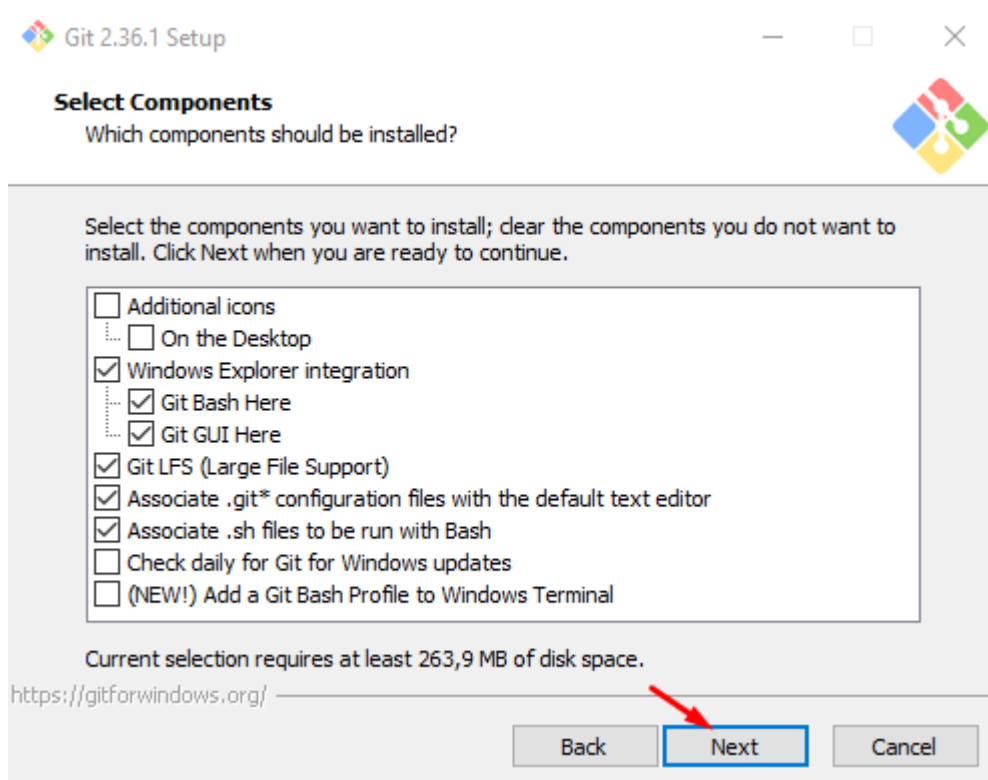
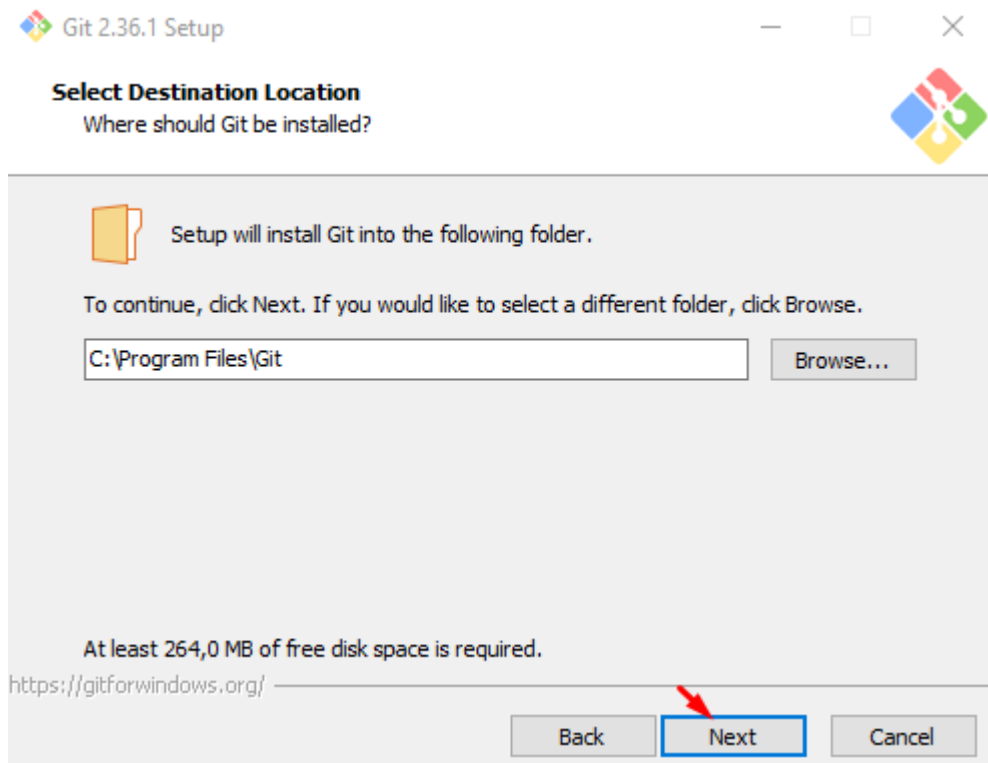
Tento configurado seu repositório remoto, faça o download do Git: <https://git-scm.com/downloads>

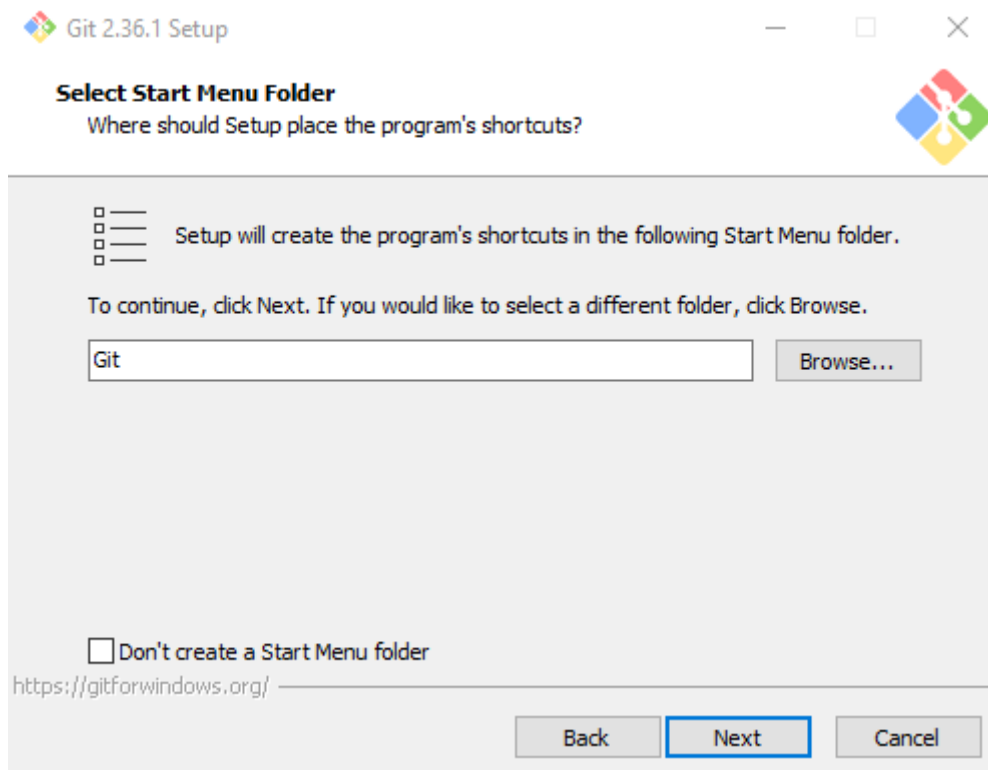
Escolha a instalação adequada para você



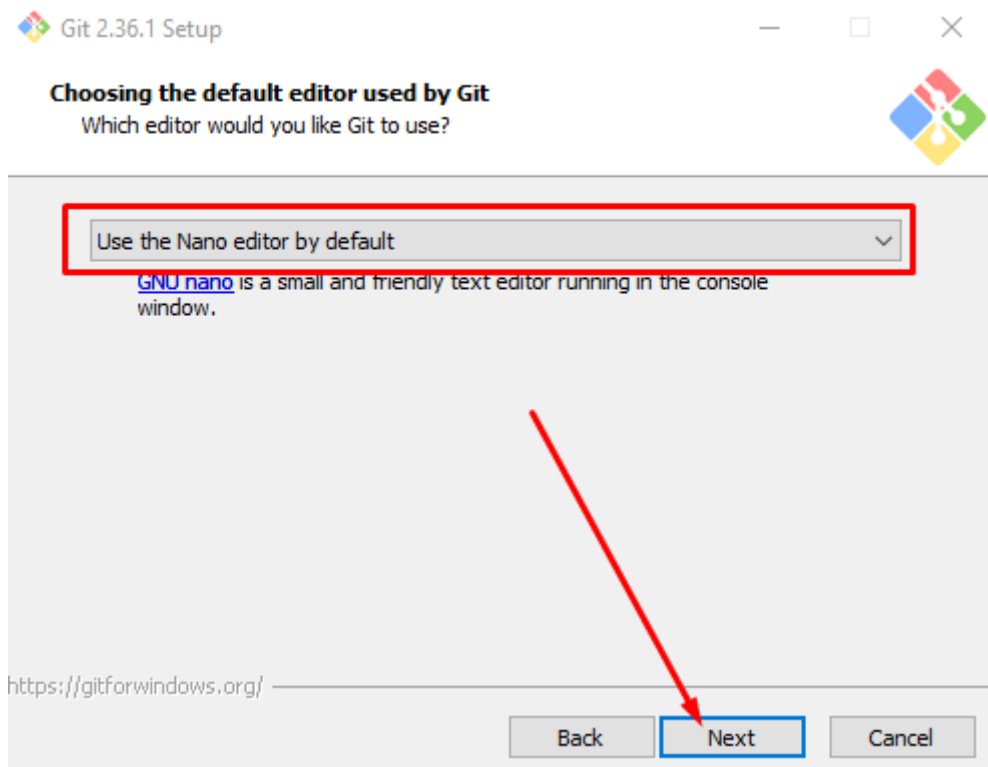
Faça a instalação, quase todos os passos da instalação são padrões, exceto um deles.







**Preste atenção**, esta é a única parte que não é padrão na instalação. Essa escolha facilitará nosso trabalho em alguns momentos.



**Adjusting the name of the initial branch in new repositories**

What would you like Git to name the initial branch after "git init"?

☒ **Let Git decide**

Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project [intends](#) to change this default to a more inclusive name in the near future.

☐ **Override the default branch name for new repositories**

**NEW!** Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

This setting does not affect existing repositories.

<https://gitforwindows.org/>

Back

Next

Cancel

**Adjusting your PATH environment**

How would you like to use Git from the command line?

☐ **Use Git from Git Bash only**

This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

☒ **Git from the command line and also from 3rd-party software**

**(Recommended)** This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

☐ **Use Git and optional Unix tools from the Command Prompt**

Both Git and the optional Unix tools will be added to your PATH.

**Warning:** This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

<https://gitforwindows.org/>

Back

Next

Cancel



### Choosing the SSH executable

Which Secure Shell client program would you like Git to use?



☒ **Use bundled OpenSSH**

This uses ssh.exe that comes with Git.

☐ **Use external OpenSSH**

**NEW!** This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.

<https://gitforwindows.org/>

Back

Next

Cancel

### Choosing HTTPS transport backend

Which SSL/TLS library would you like Git to use for HTTPS connections?



☒ **Use the OpenSSL library**

Server certificates will be validated using the ca-bundle.crt file.

☐ **Use the native Windows Secure Channel library**

Server certificates will be validated using Windows Certificate Stores.  
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

<https://gitforwindows.org/>

Back

Next

Cancel

**Configuring the line ending conversions**

How should Git treat line endings in text files?

☒ **Checkout Windows-style, commit Unix-style line endings**

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

☐ **Checkout as-is, commit Unix-style line endings**

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

☐ **Checkout as-is, commit as-is**

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

<https://gitforwindows.org/>

Back

Next

Cancel

**Configuring the terminal emulator to use with Git Bash**

Which terminal emulator do you want to use with your Git Bash?

☒ **Use MinTTY (the default terminal of MSYS2)**

Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.

☐ **Use Windows' default console window**

Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.

<https://gitforwindows.org/>

Back

Next

Cancel

**Choose the default behavior of `git pull`**

What should `git pull` do by default?

☒ **Default (fast-forward or merge)**

This is the standard behavior of `git pull`: fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.

☐ **Rebase**

Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.

☐ **Only ever fast-forward**

Fast-forward to the fetched branch. Fail if that is not possible.

<https://gitforwindows.org/>

Back

Next

Cancel

**Choose a credential helper**

Which credential helper should be configured?

☒ **Git Credential Manager**

Use the [cross-platform Git Credential Manager](#).  
See more information about the future of Git Credential Manager [here](#).

☐ **None**

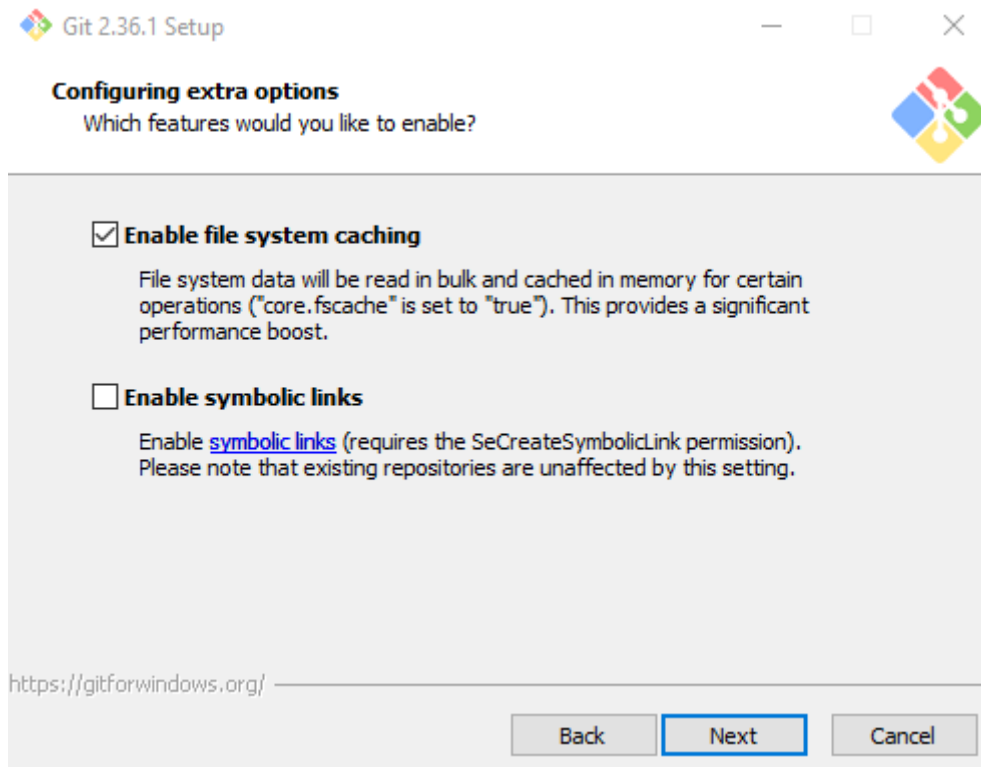
Do not use a credential helper.

<https://gitforwindows.org/>

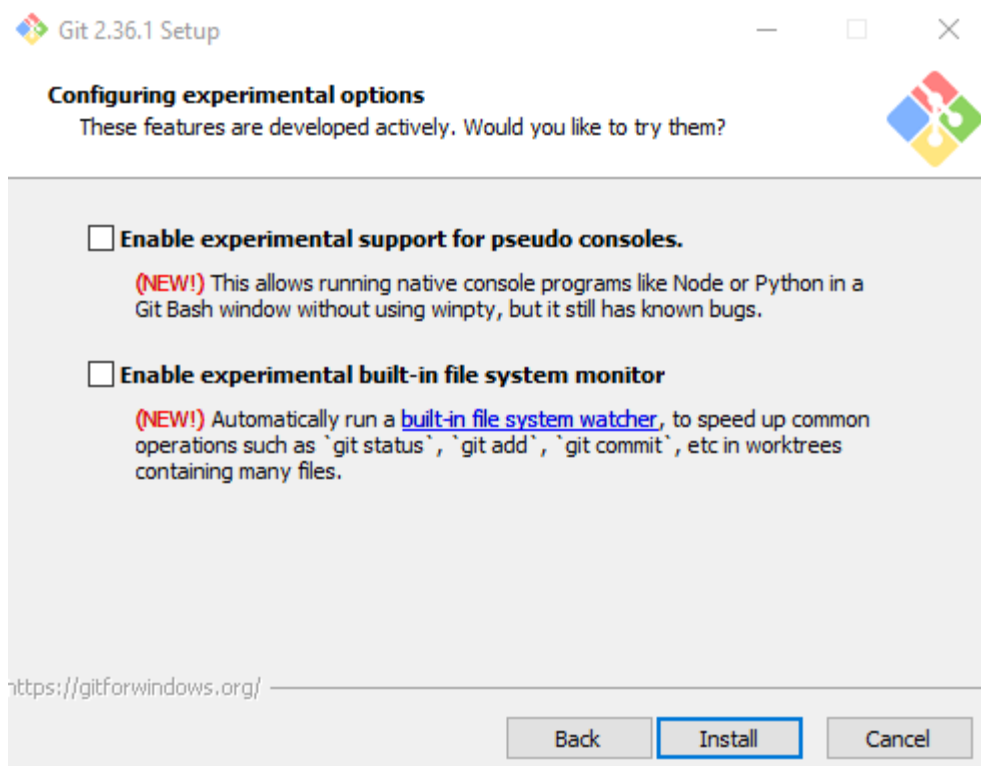
Back

Next

Cancel



Finalmente, clique para instalar.



Sucesso!

Git 2.36.1 Setup

## Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

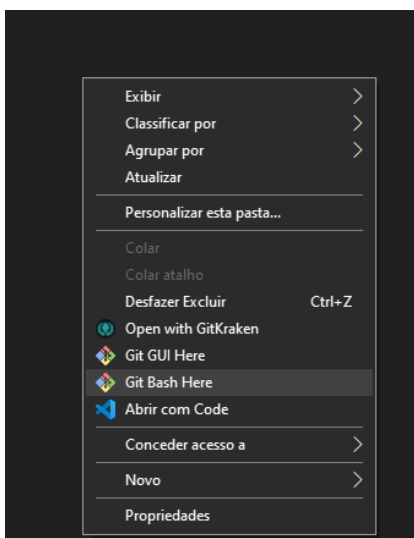


Click Finish to exit Setup.

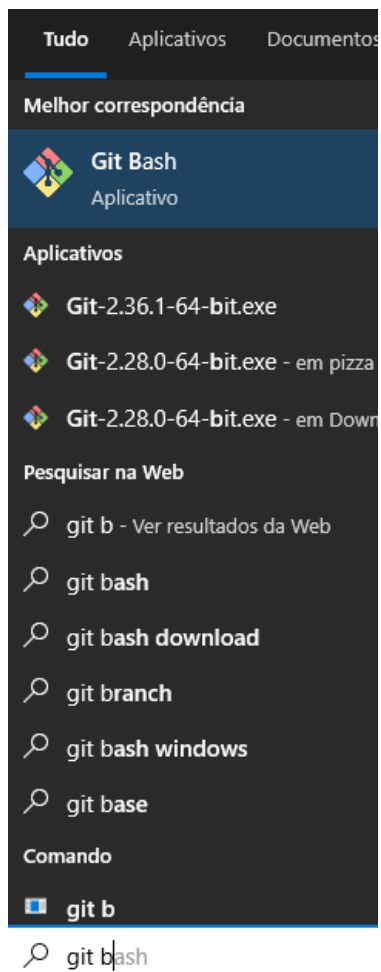
- ☐ Launch Git Bash
- ☒ View Release Notes

Finish

Você agora deve conseguir abrir seu Git Bash no menu de contexto.



Ou o pesquisando pelo Windows



Verifique suas configurações do Git com este comando:

```
git config --list
```

```
MINGW64:/c/src/Fontes/Aulas

Cleiton@DESKTOP-PAQEDE7 MINGW64 /c/src/Fontes/Aulas
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
core.editor=nano.exe
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=cleiton.capristano@gmail.com
user.name=Cleiton Capristano
core.longpaths=true

Cleiton@DESKTOP-PAQEDE7 MINGW64 /c/src/Fontes/Aulas
$
```

Altere a maneira como as pessoas verão suas alterações com os seguintes comandos.

```
git config --global user.name "Your Name"
```

```
git config --global user.email "youremail@yourdomain.com"
```

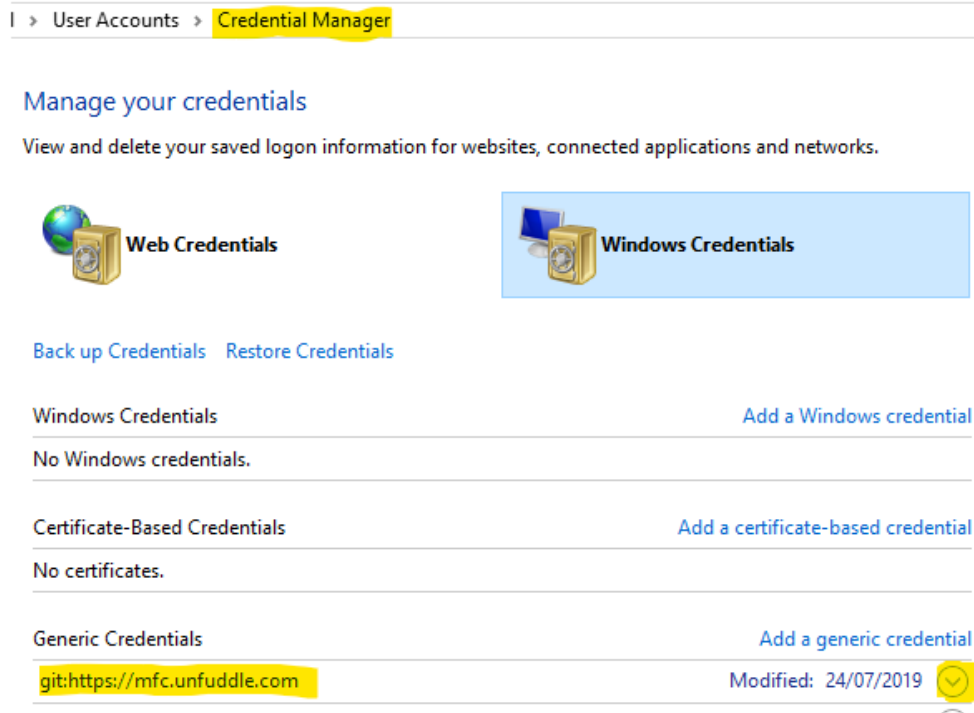
Digite novamente o comando abaixo para testar as alterações:

```
git config --list
```

Caso já possuam um usuário configurado no Git local, acessar as credenciais do windows (por exemplo) e excluir as credenciais.

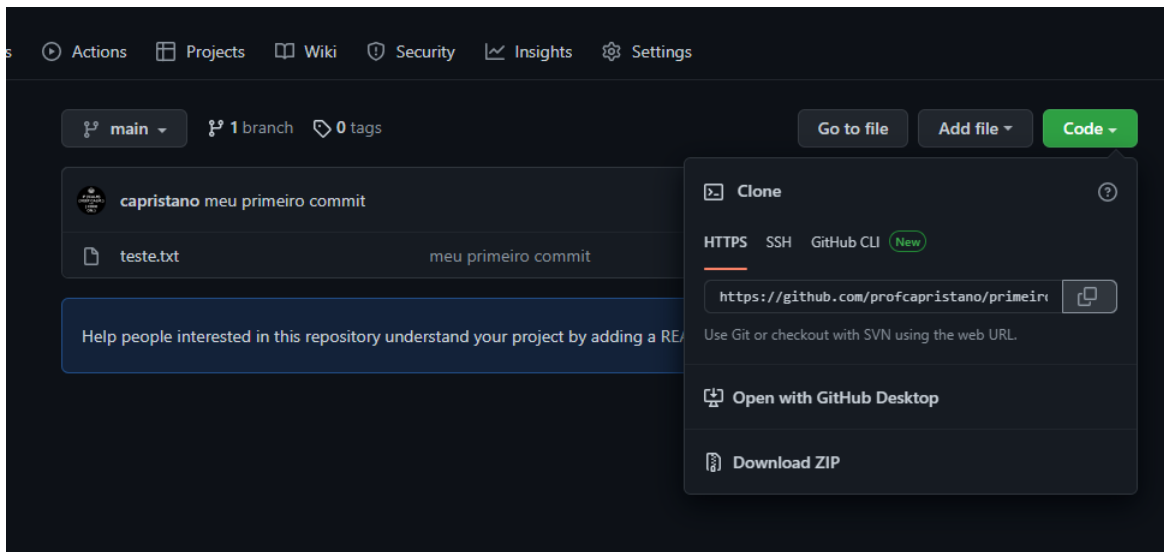
When performing **step 2** if you see `credential.helper=manager` you need to open the credential manager of your computer (Win or Mac) and update the credentials there

Here is how it look on windows



Por fim, copiem o caminho de seu repositório, para conseguir clona-lo para seu ambiente local.





Para clonar um repositório:

`git clone https://github.com/profcapristano/primeiro\_repo.git`

Na primeira alteração que necessite um acesso privilegiado a um repositório vocês poderão se autenticar no Git.

