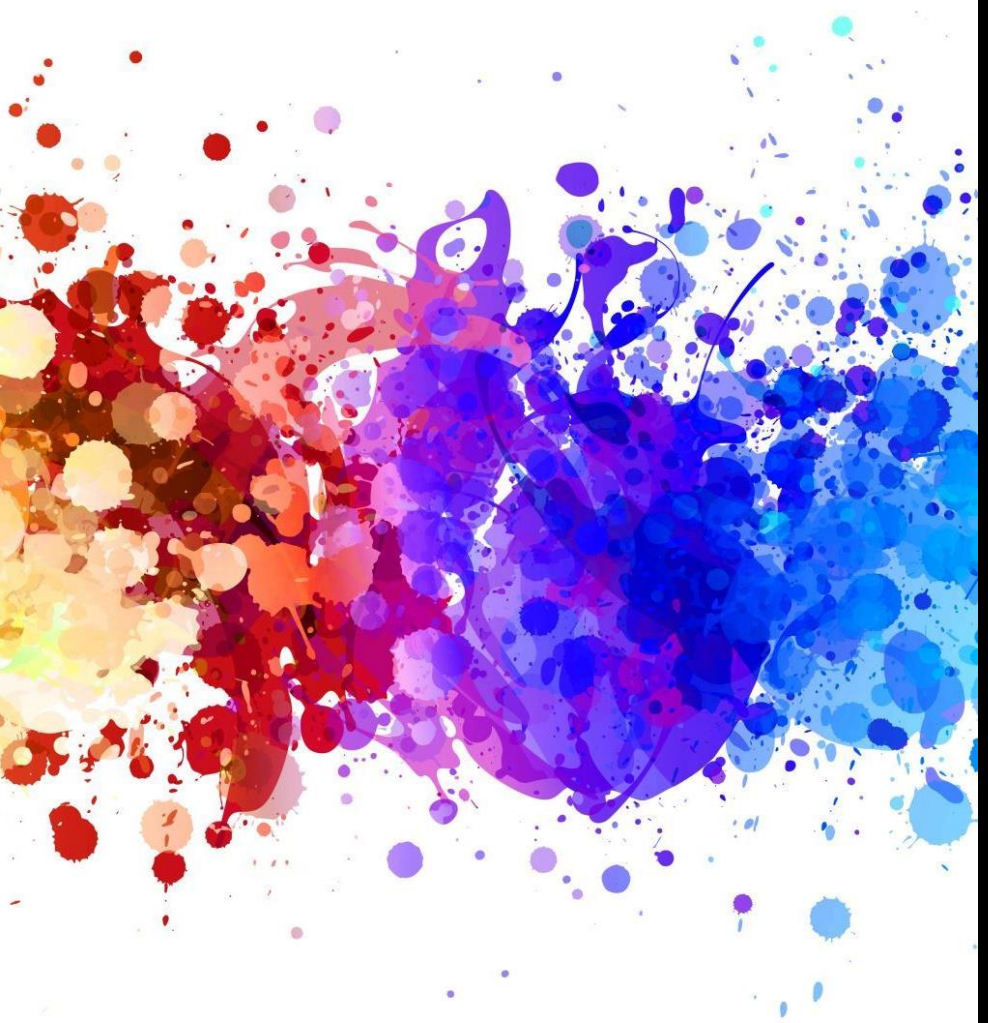


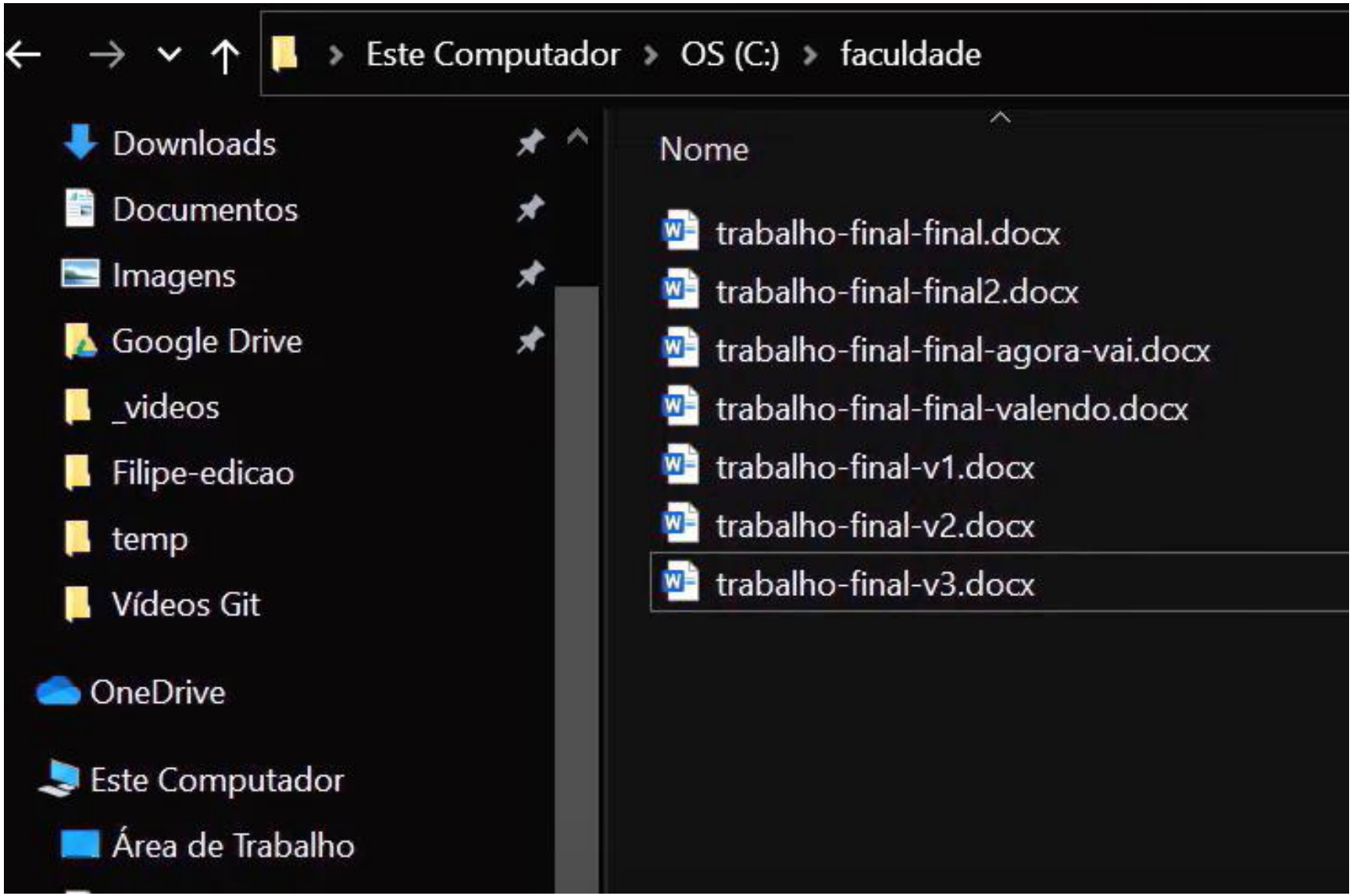
GIT E GITHUB

Na prática

GIT

Git em inglês britânico é um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software, mas pode ser usado para registrar o histórico de edições de qualquer tipo de arquivo.

https://git-scm.com/

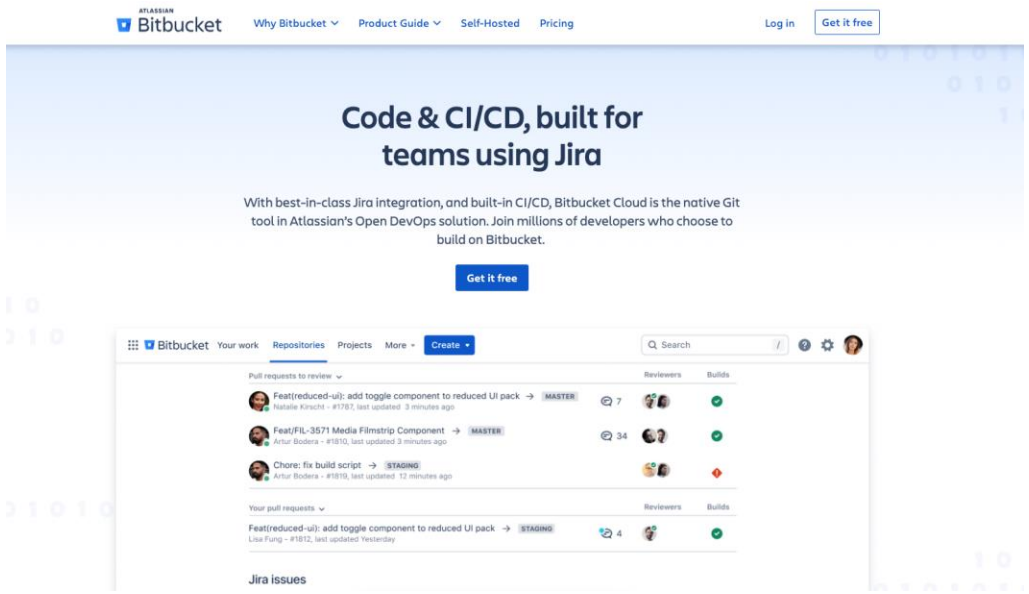

Este Computador
OS (C:)
faculdade
Downloads
Documentos
Imagens
Google Drive
_videos
Filipe-edicao
temp
Vídeos Git
OneDrive
Este Computador
Área de Trabalho
Nome
trabalho-final-final.docx
trabalho-final-final2.docx
trabalho-final-final-agora-vai.docx
trabalho-final-final-valendo.docx
trabalho-final-v1.docx
trabalho-final-v2.docx
trabalho-final-v3.docx

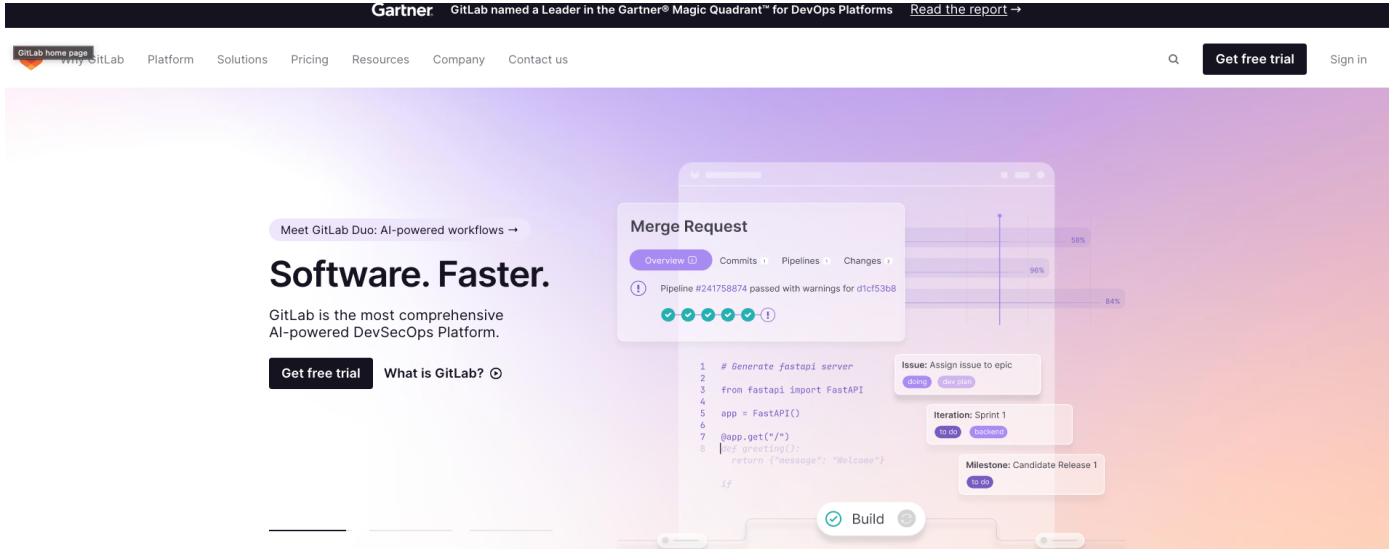
GITHUB

GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo.

https://github.com/

CONCORRENTES





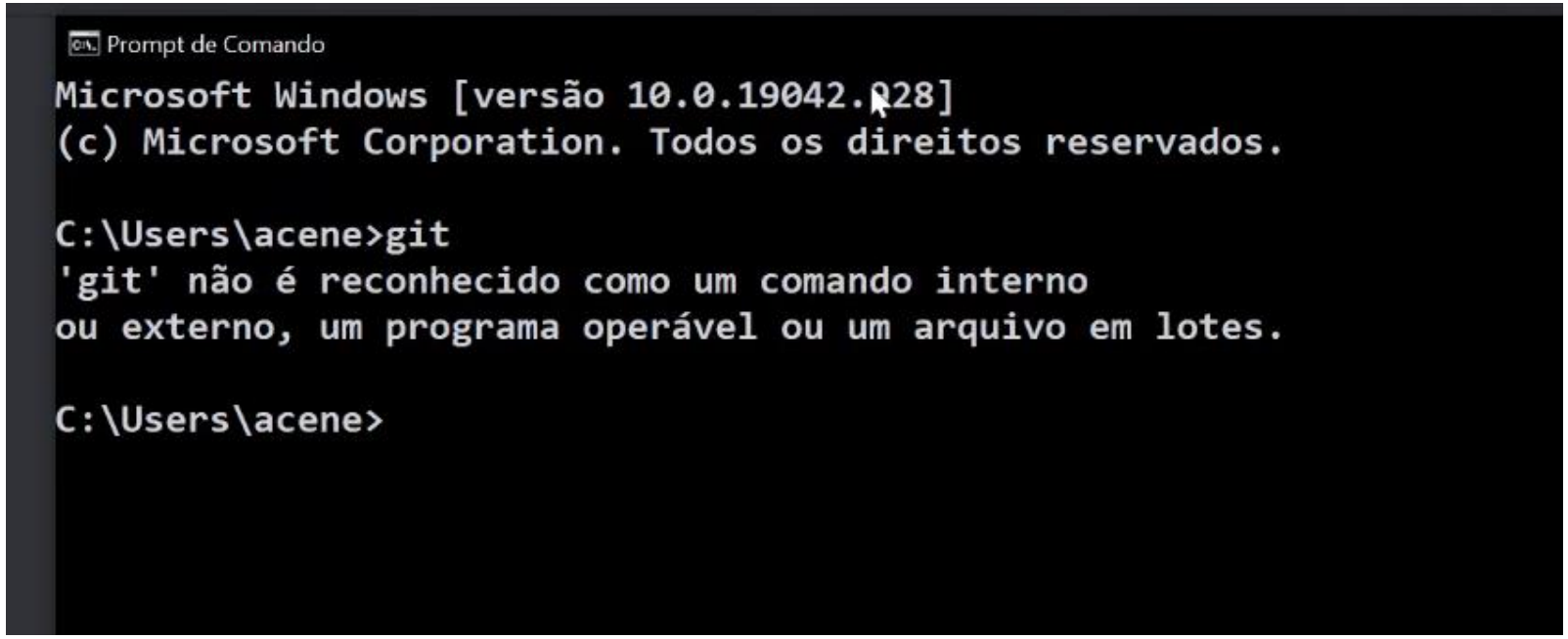
https://bitbucket.org/product/

https://about.gitlab.com/

INSTALANDO O GIT

PARA SABER SE JÁ ESTÁ NA MAQUINA

Abra o cmd (terminal) e digite git.



git --local-branching-on-the-cheap

Search entire site...

About

Documentation

Downloads

GUI Clients

Logos

Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Downloads

macOS Windows Linux/Unix

Older releases are available and the Git source repository is on GitHub.

GUI Clients

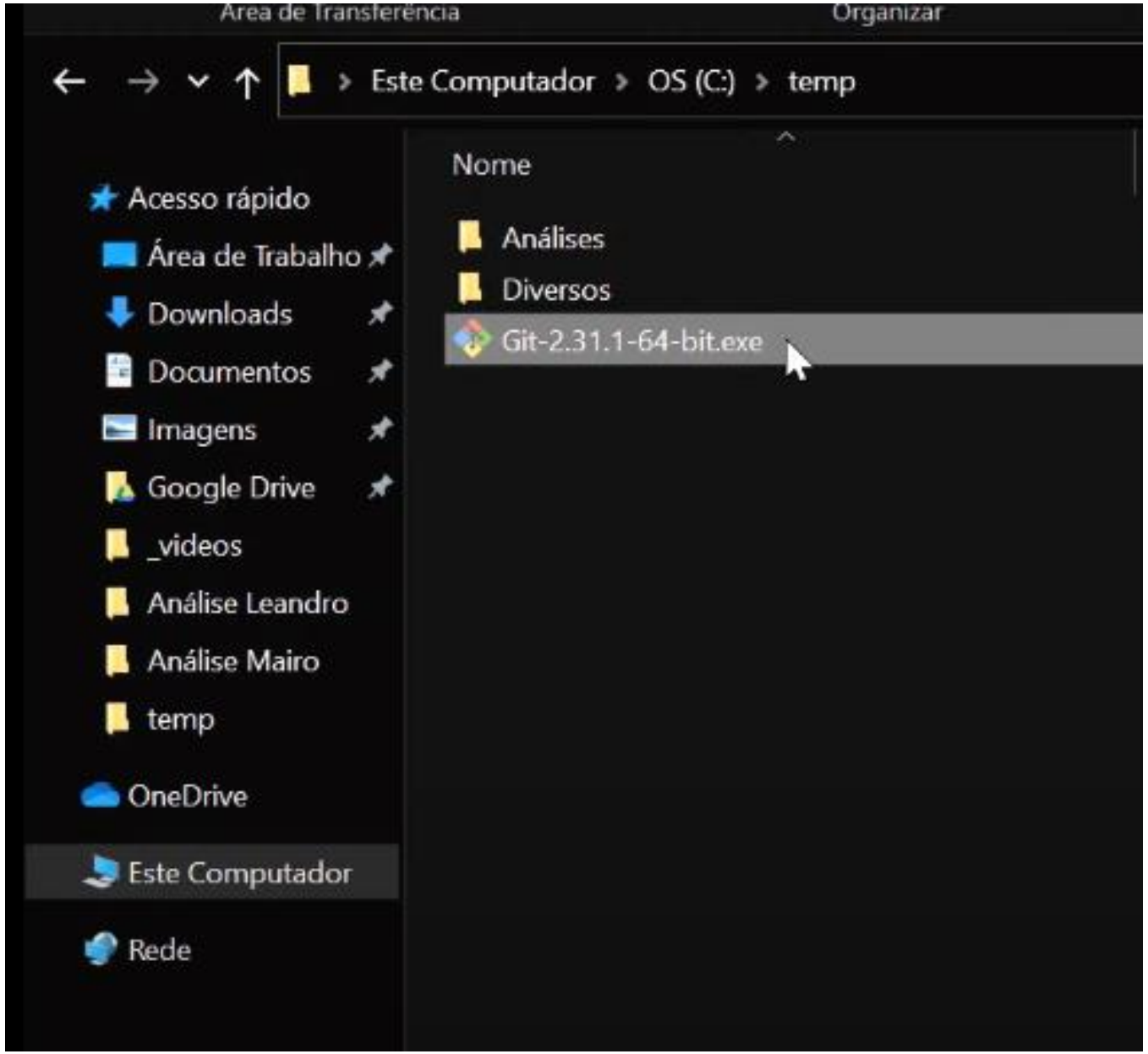
Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

View GUI Clients →

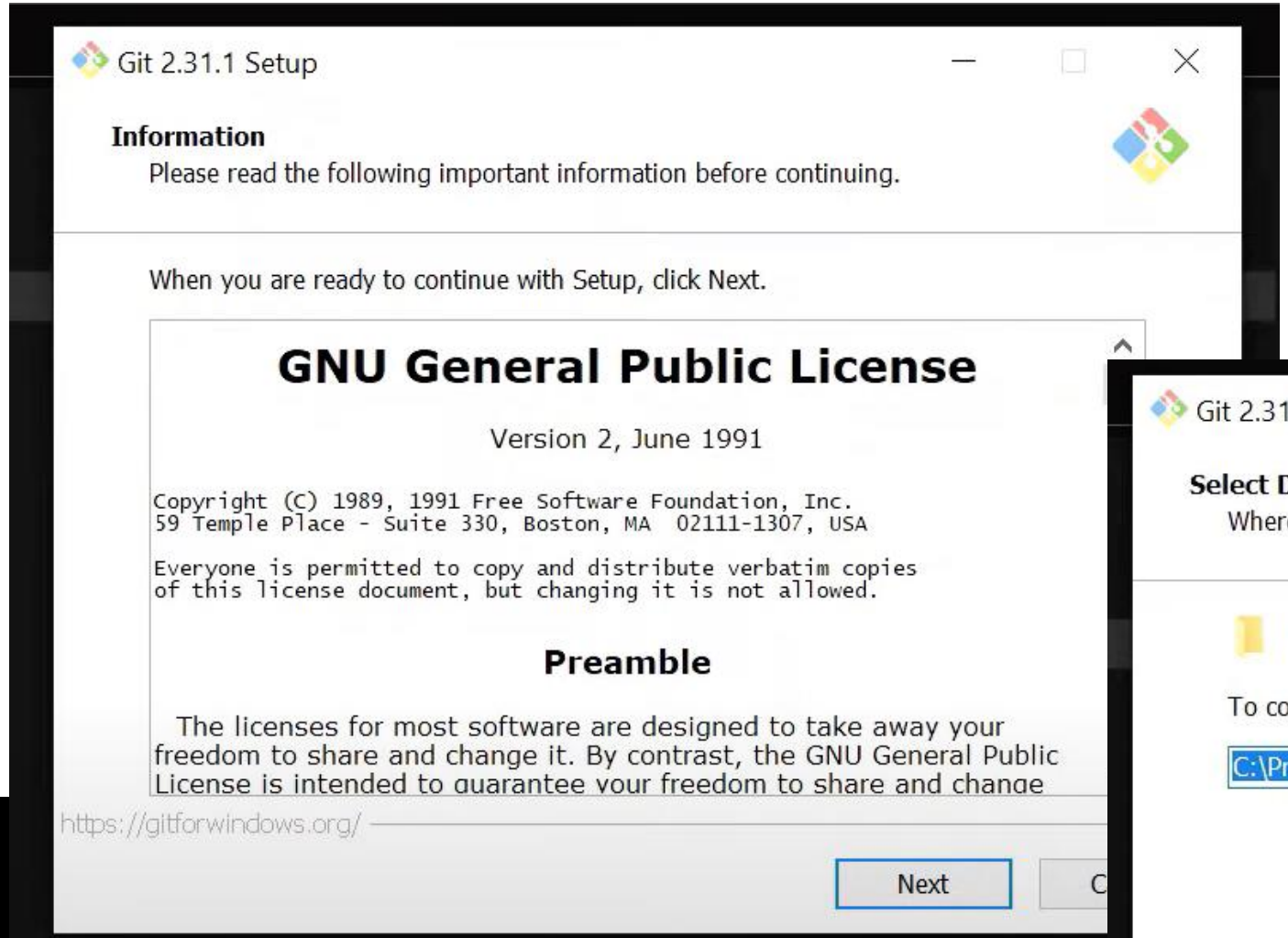
Logos

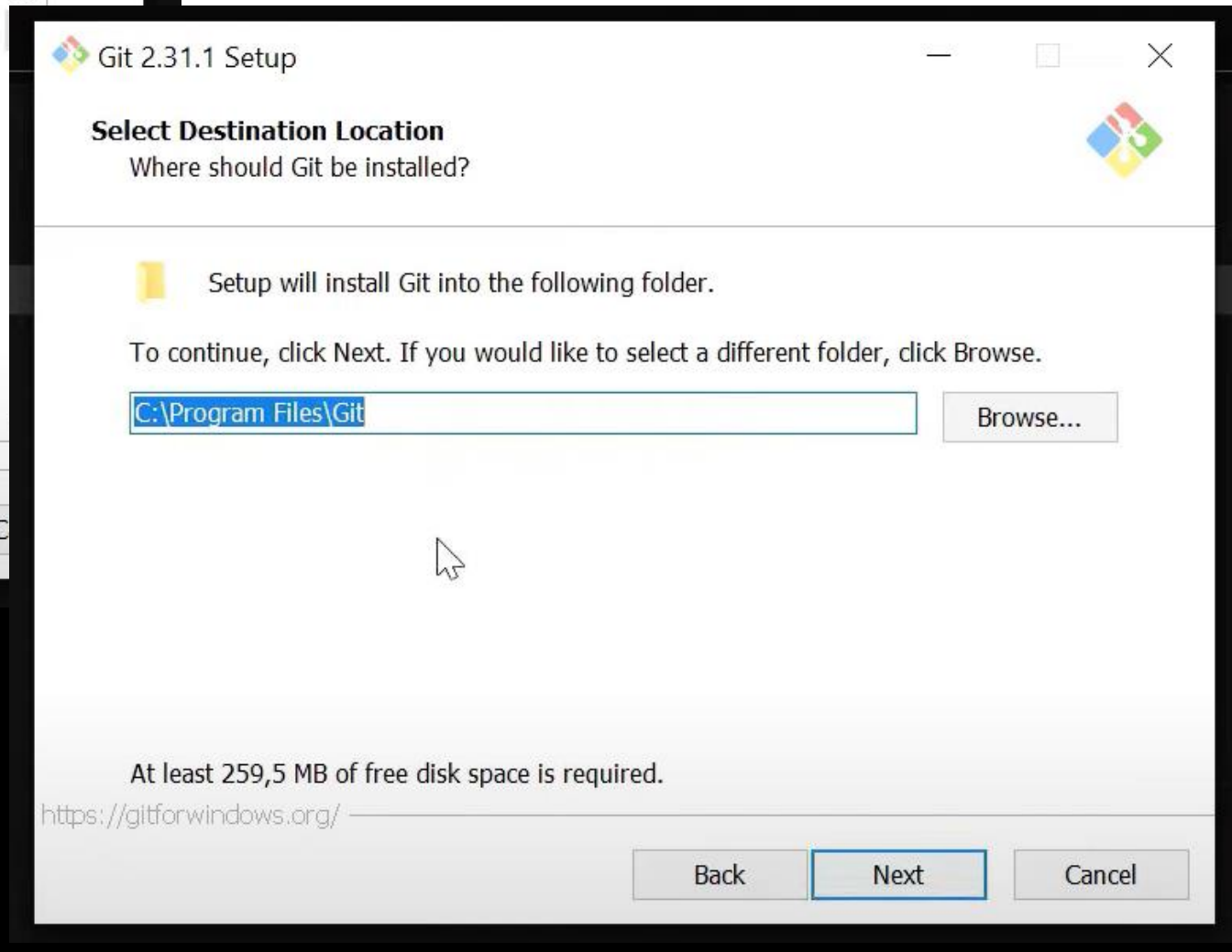
Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

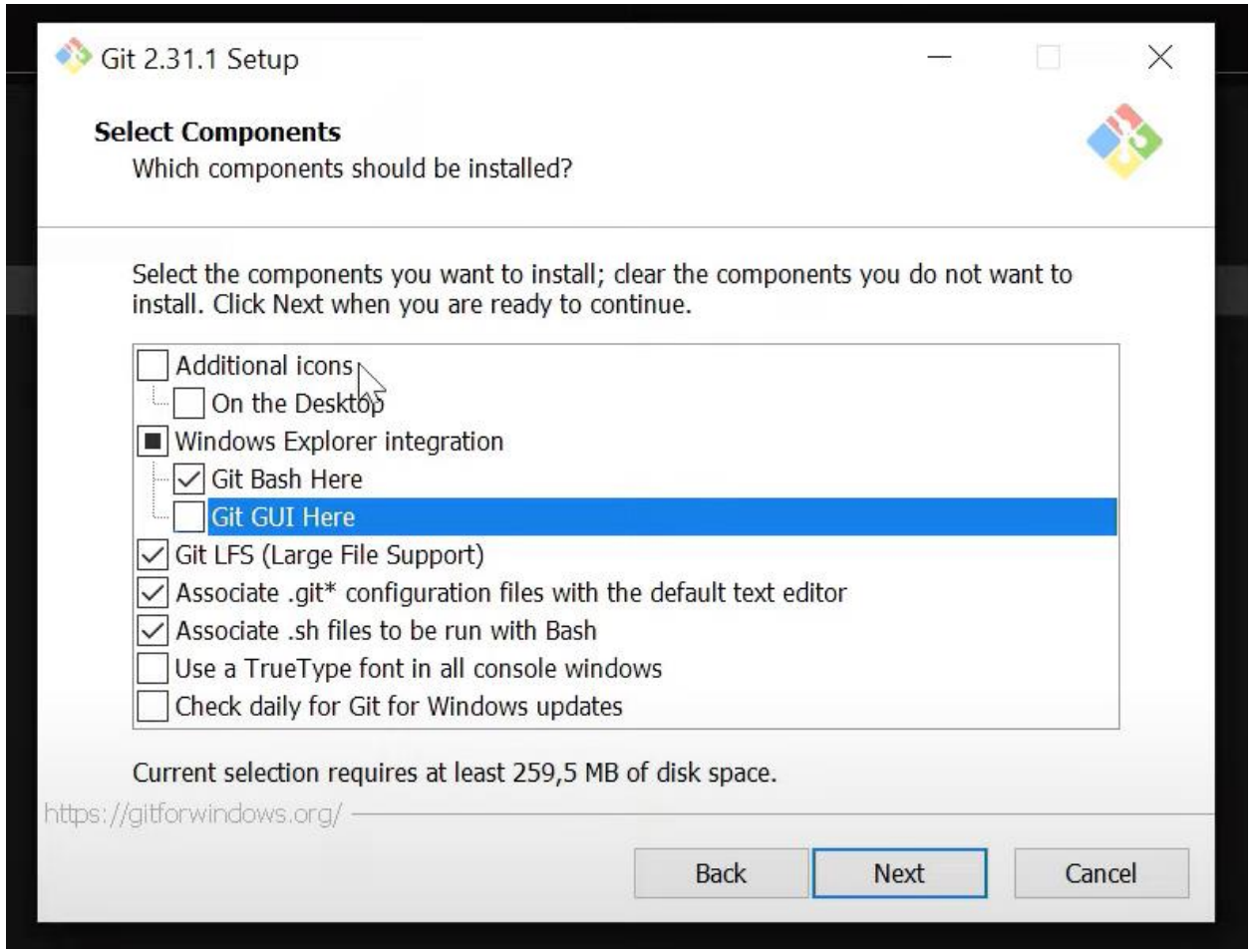
View Logos →

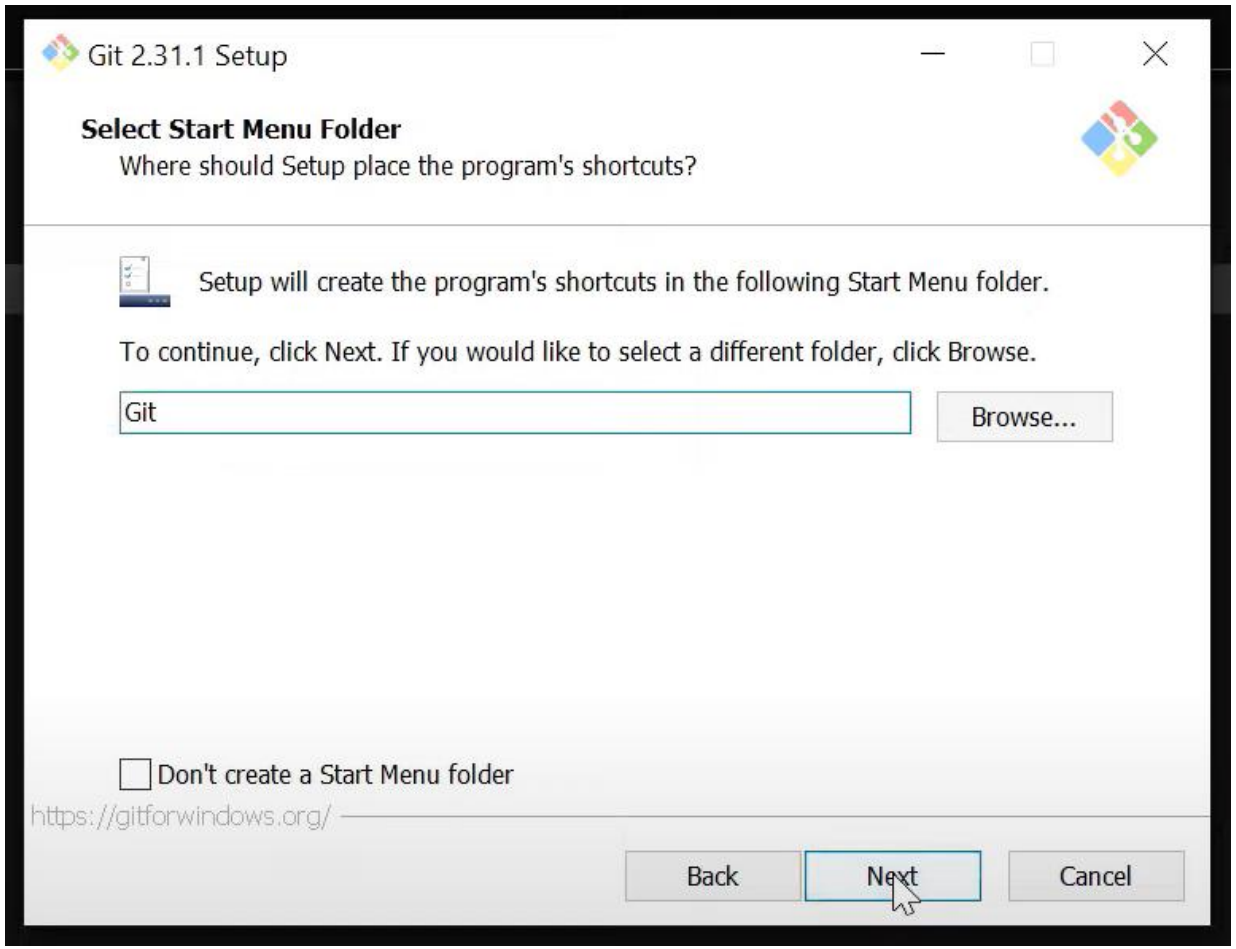


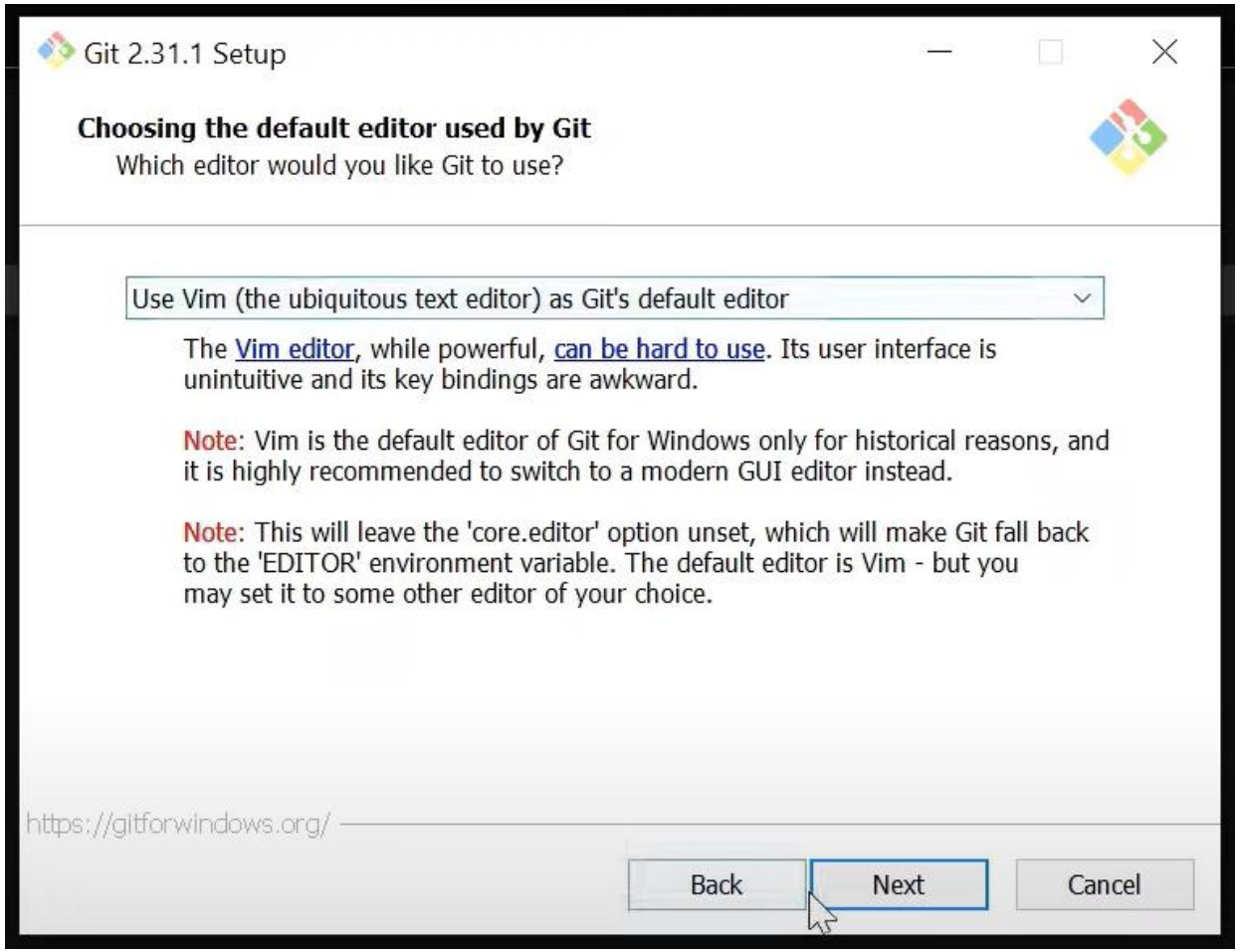
Execute o código baixado

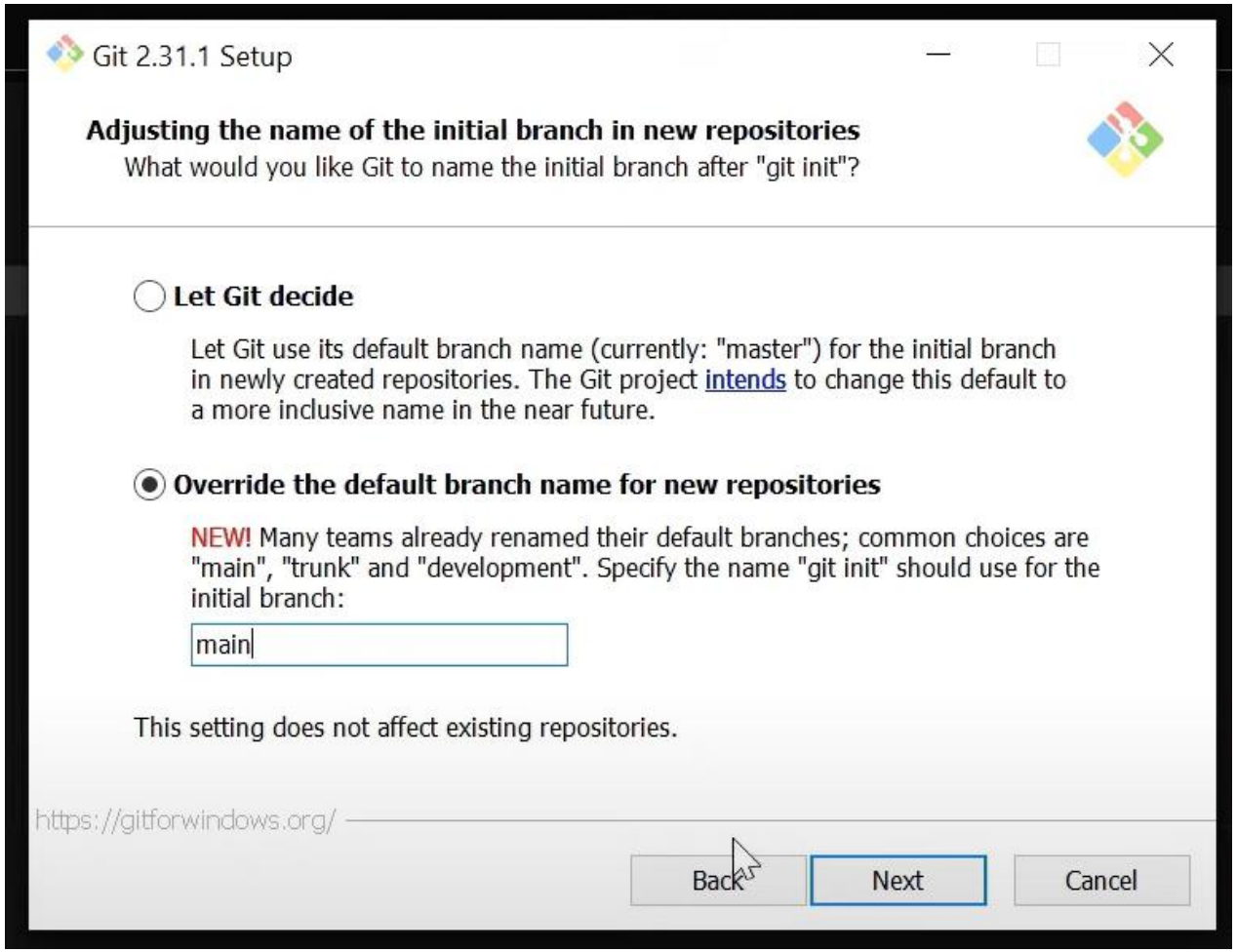

Git 2.31.1 Setup
Information
Please read the following important information before continuing.
When you are ready to continue with Setup, click Next.
GNU General Public License
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
Preamble
The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change
https://gitforwindows.org/
Next
C

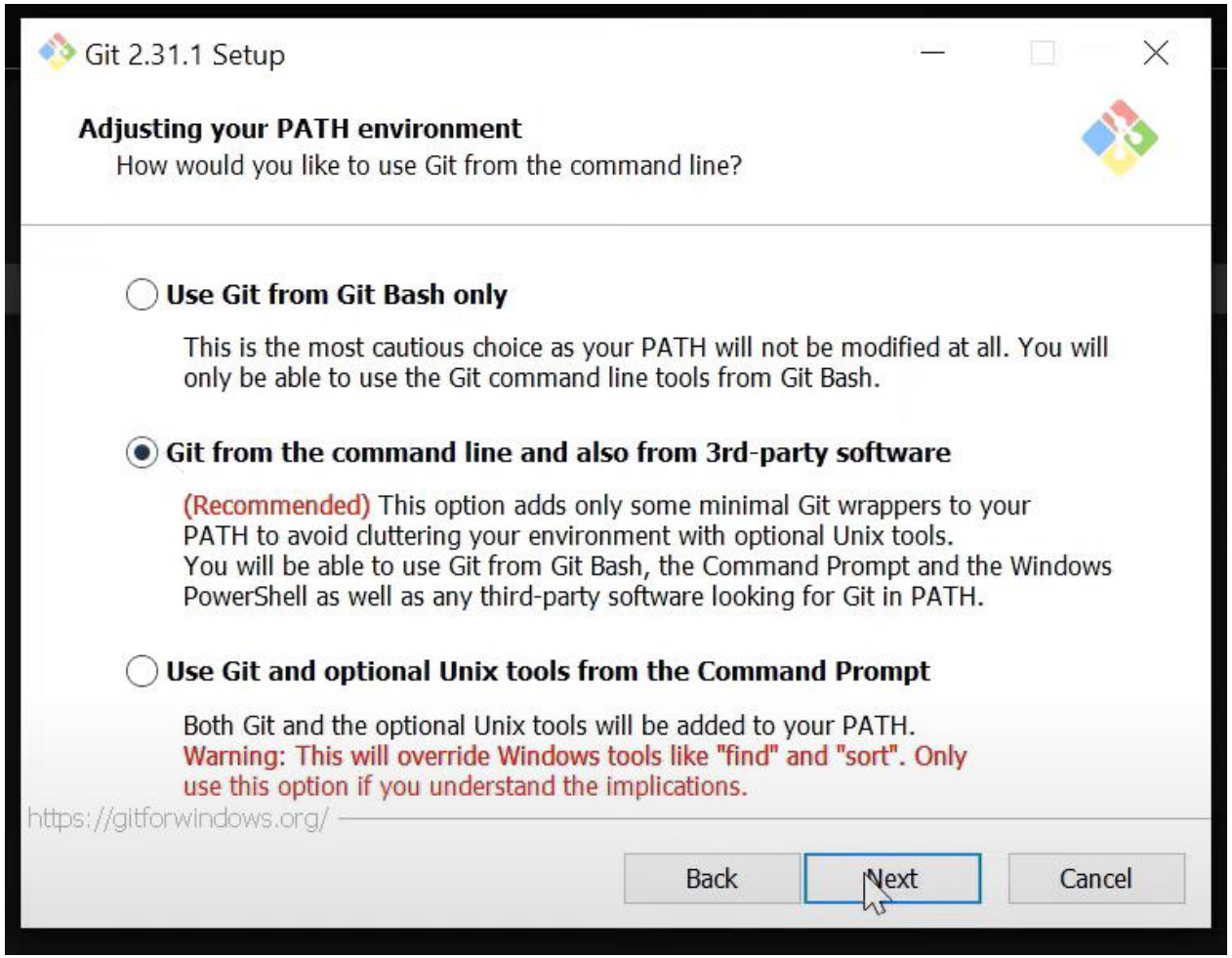

Git 2.31.1 Setup
Select Destination Location
Where should Git be installed?
Setup will install Git into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Program Files\Git
Browse...
At least 259,5 MB of free disk space is required.
https://gitforwindows.org/
Back
Next
Cancel

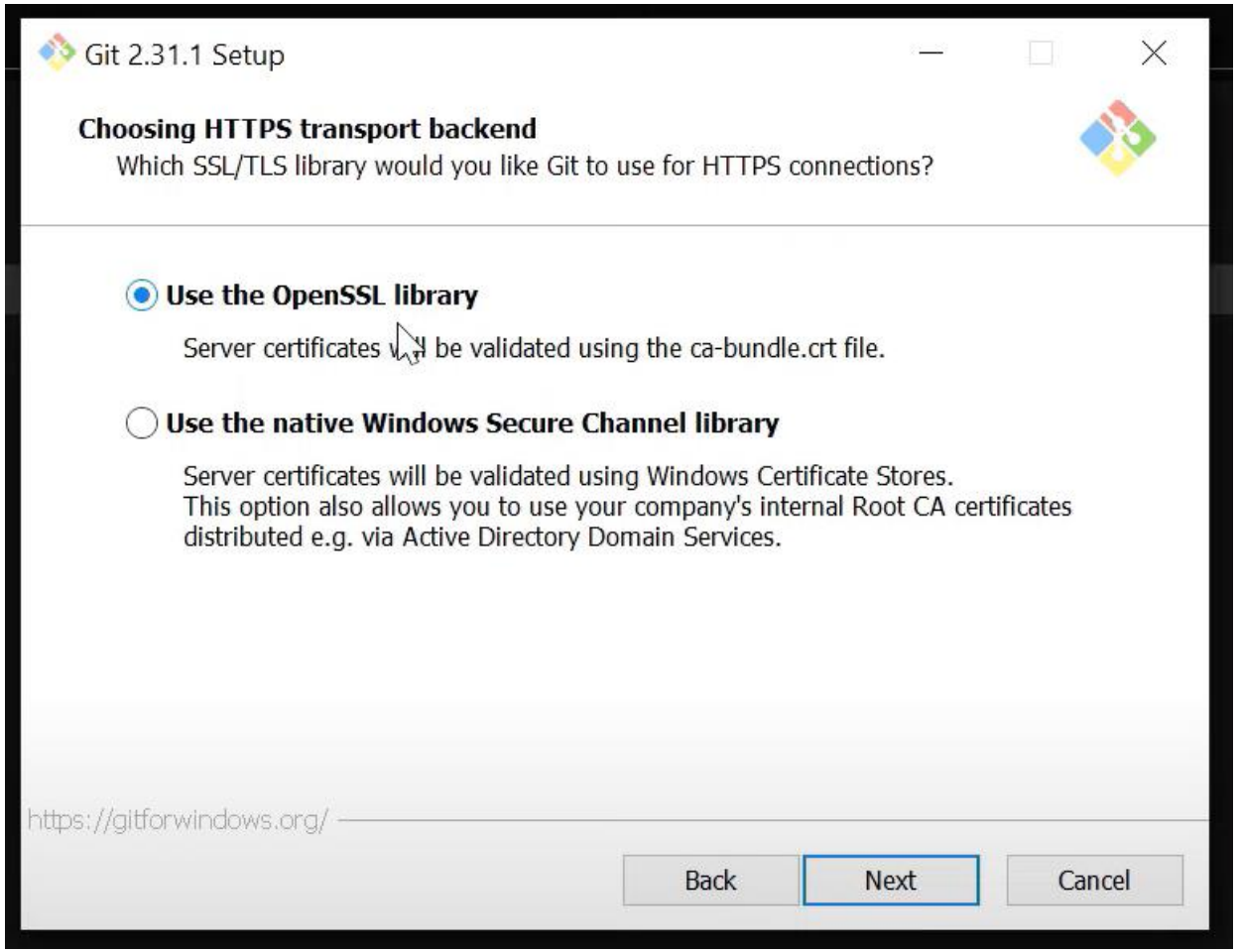

Git 2.31.1 Setup
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Additional icons
On the Desktop
Windows Explorer integration
Git Bash Here
Git GUI Here
Git LFS (Large File Support)
Associate .git* configuration files with the default text editor
Associate .sh files to be run with Bash
Use a TrueType font in all console windows
Check daily for Git for Windows updates
Current selection requires at least 259,5 MB of disk space.
https://gitforwindows.org/
Back
Next
Cancel

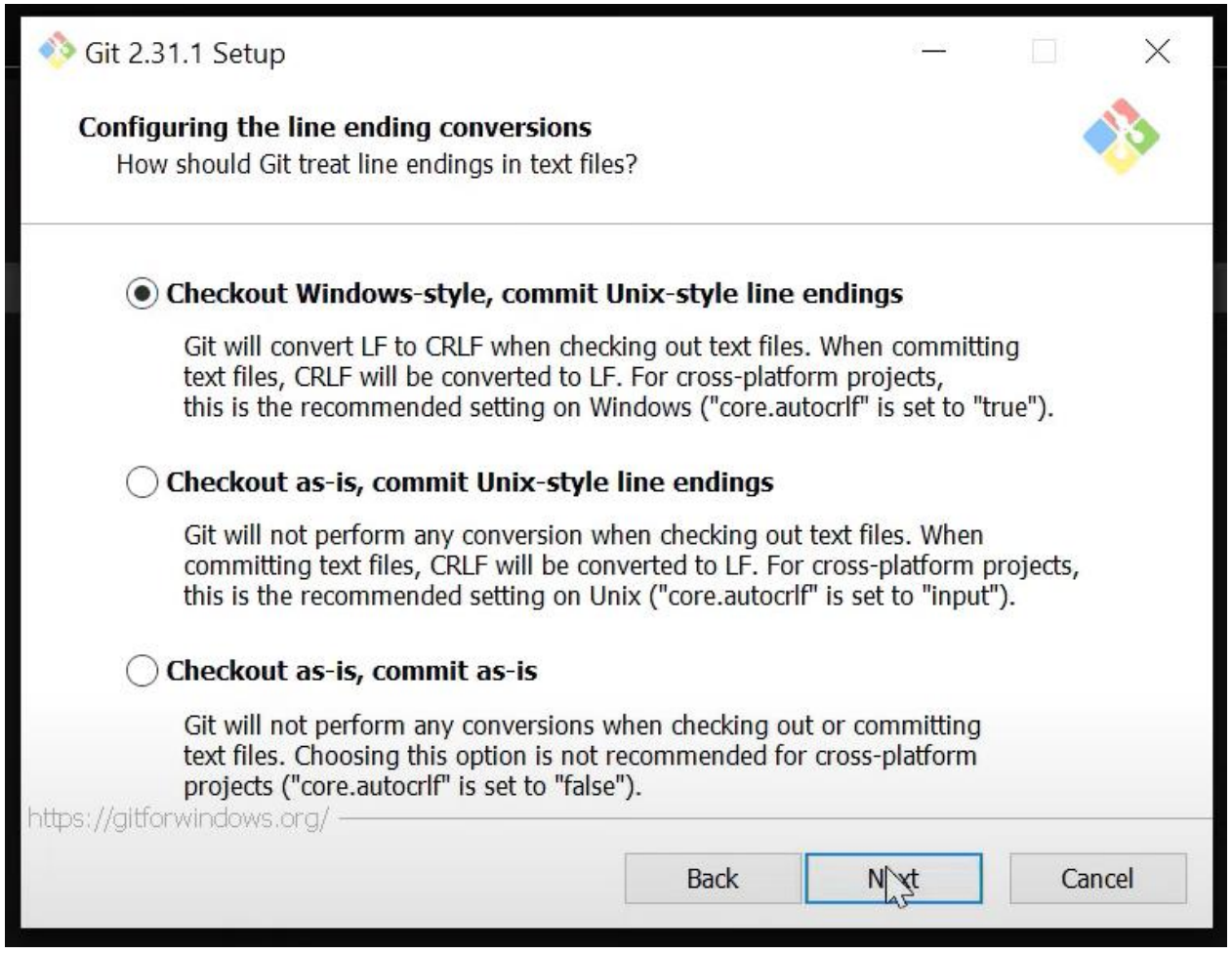

Git 2.31.1 Setup
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
Git
Browse...
Don't create a Start Menu folder
https://gitforwindows.org/
Back
Next
Cancel

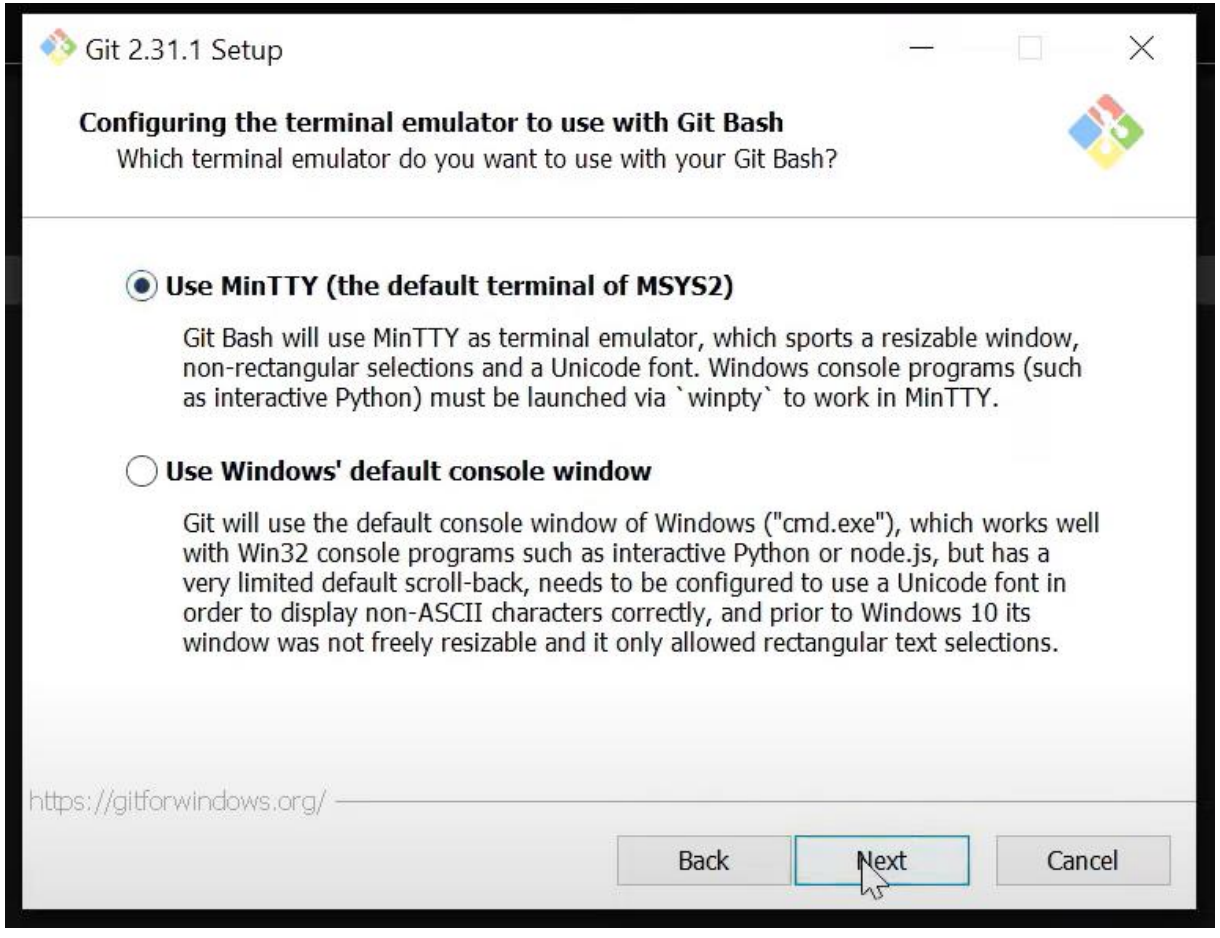

Git 2.31.1 Setup
Choosing the default editor used by Git
Which editor would you like Git to use?
Use Vim (the ubiquitous text editor) as Git's default editor
The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.
Note: Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.
Note: This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.
https://gitforwindows.org/
Back
Next
Cancel

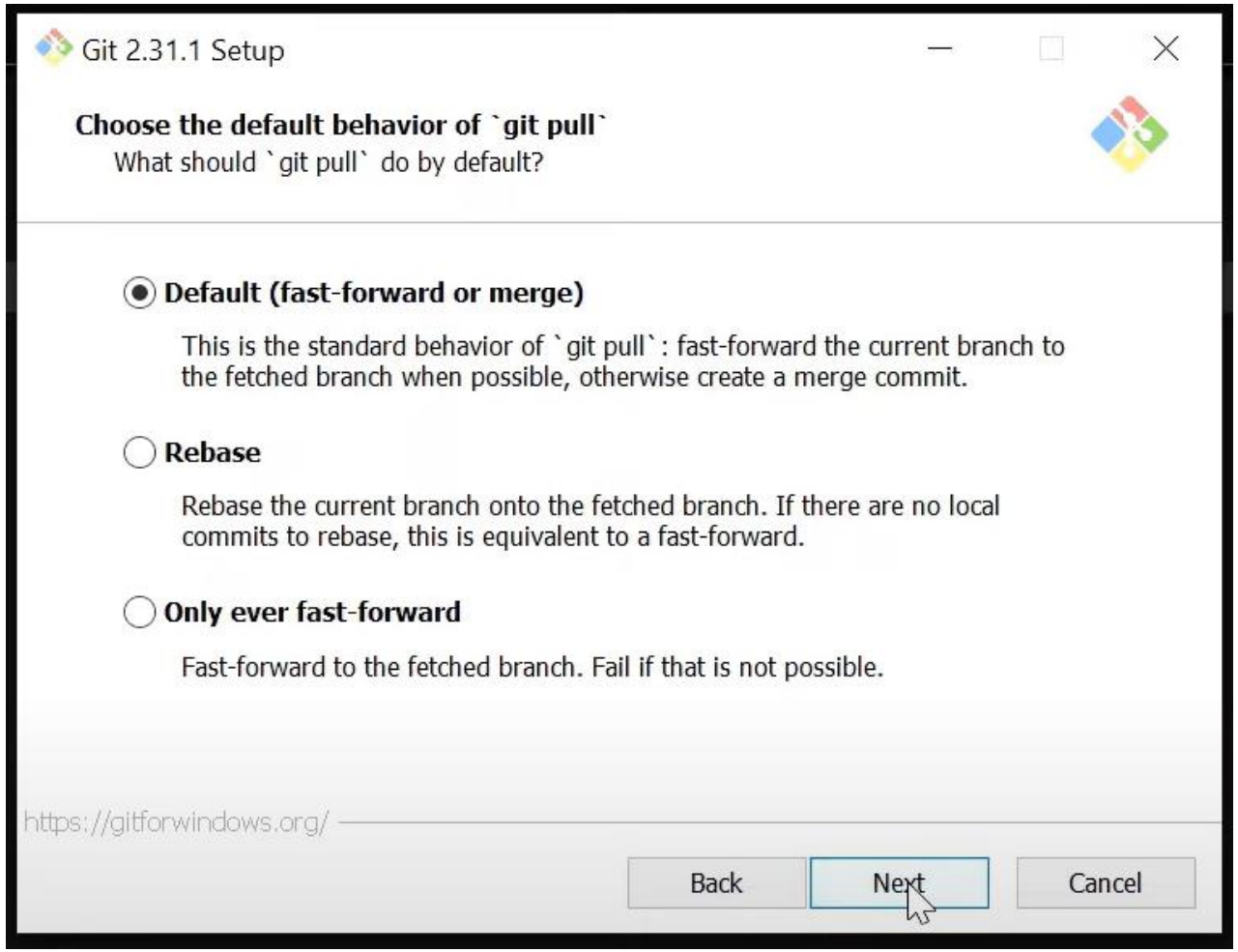

Git 2.31.1 Setup
Adjusting the name of the initial branch in new repositories
What would you like Git to name the initial branch after "git init"?
Let Git decide
Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project intends to 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:
main
This setting does not affect existing repositories.
https://gitforwindows.org/
Back
Next
Cancel

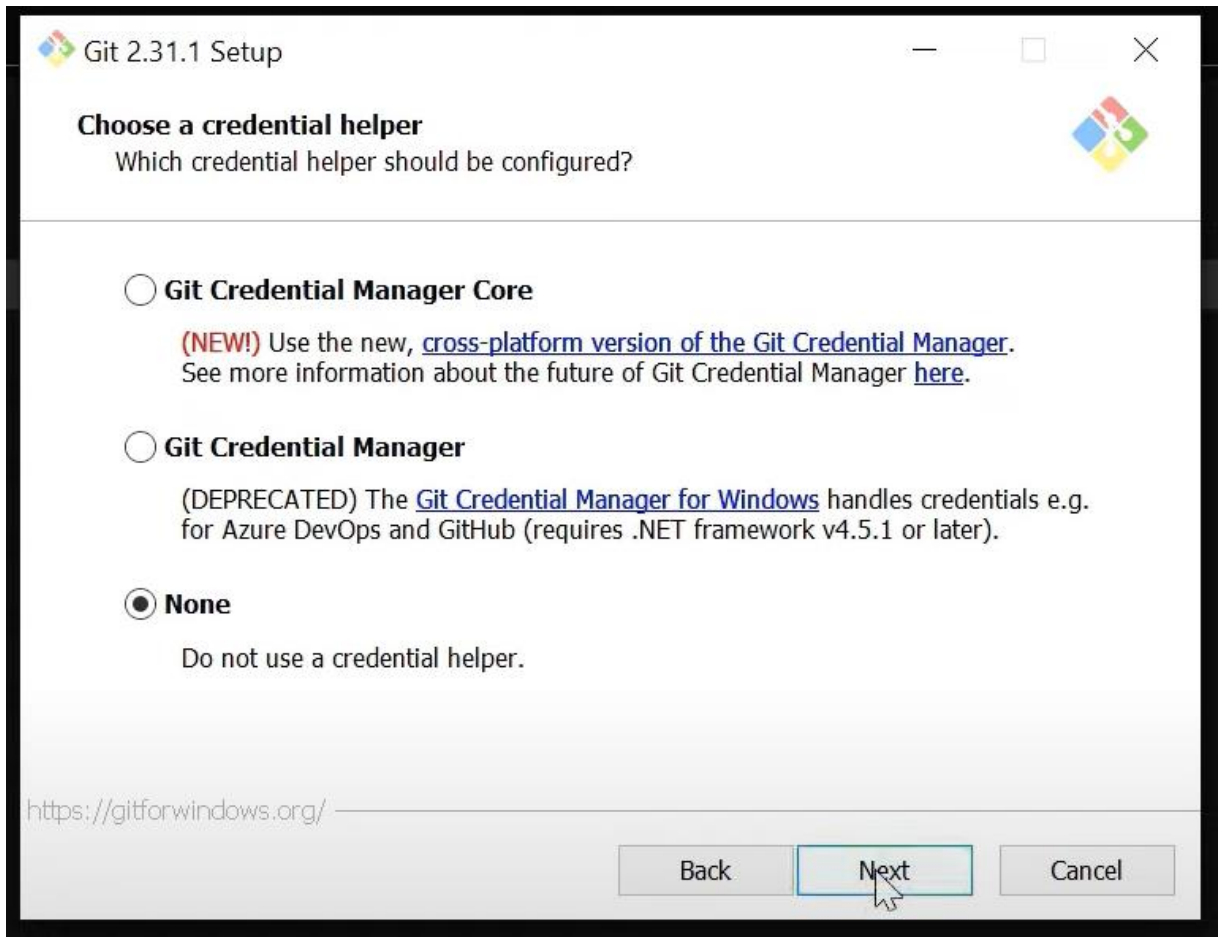

Git 2.31.1 Setup
Adjusting your PATH environment
How would you like to use Git from the command line?
Use Git from Git Bash only
This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.
Git from the command line 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

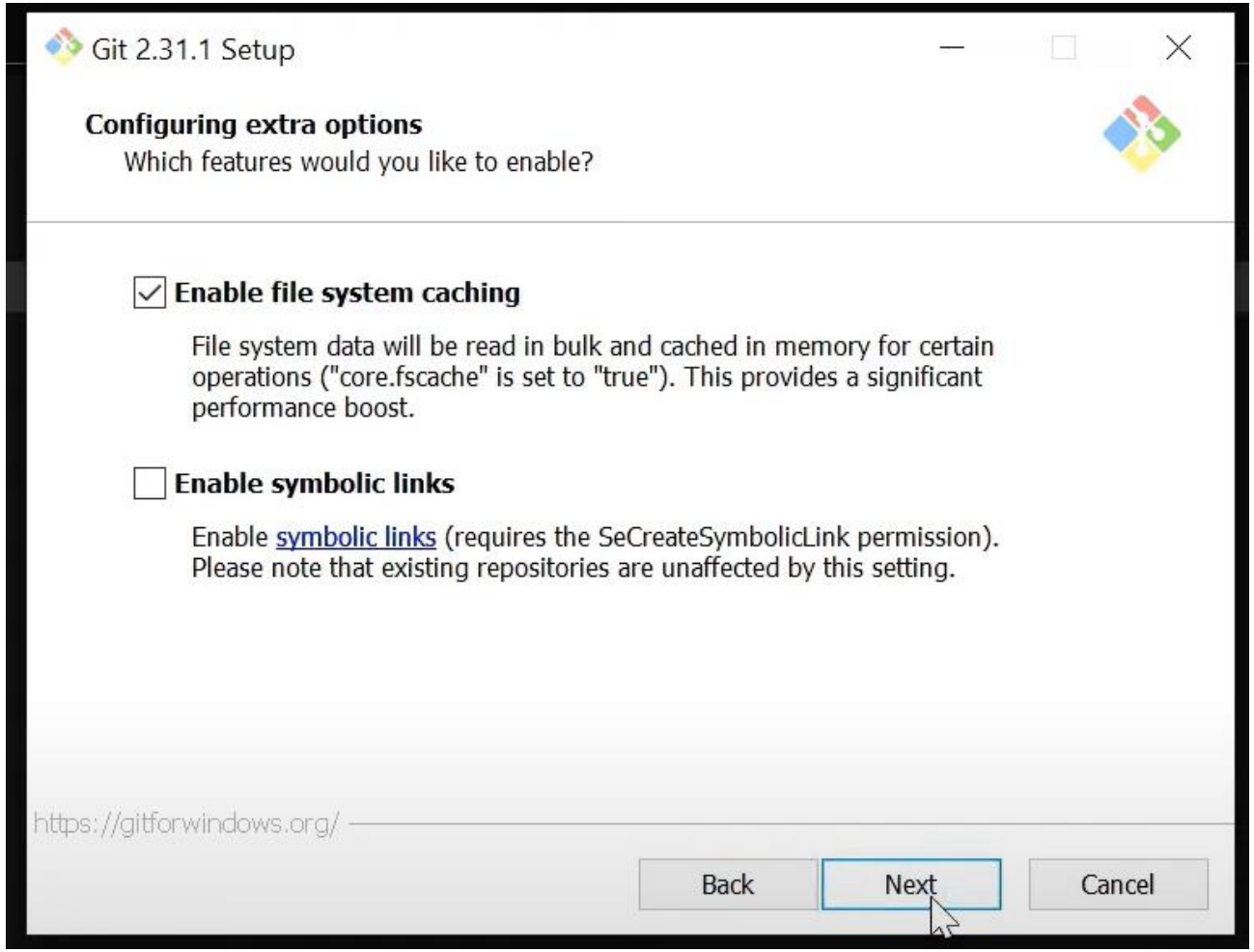

Git 2.31.1 Setup
Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?
Use the OpenSSL library
Server certificates will be validated using the ca-bundle.crt file.
Use the native Windows Secure Channel library
Server certificates will be validated using 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

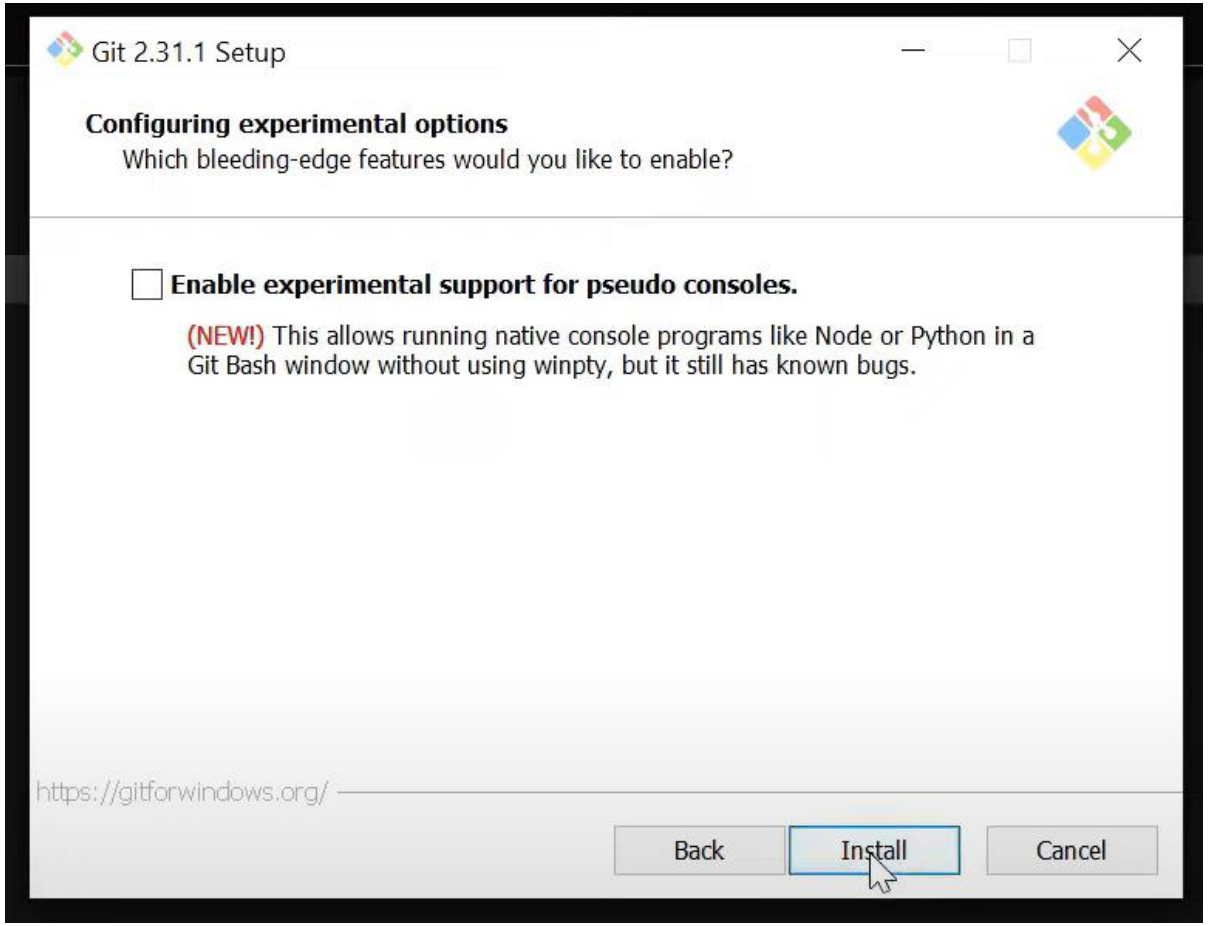

Git 2.31.1 Setup
Configuring the line ending conversions
How should Git treat line endings in text files?
Checkout Windows-style, commit Unix-style line endings
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, 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

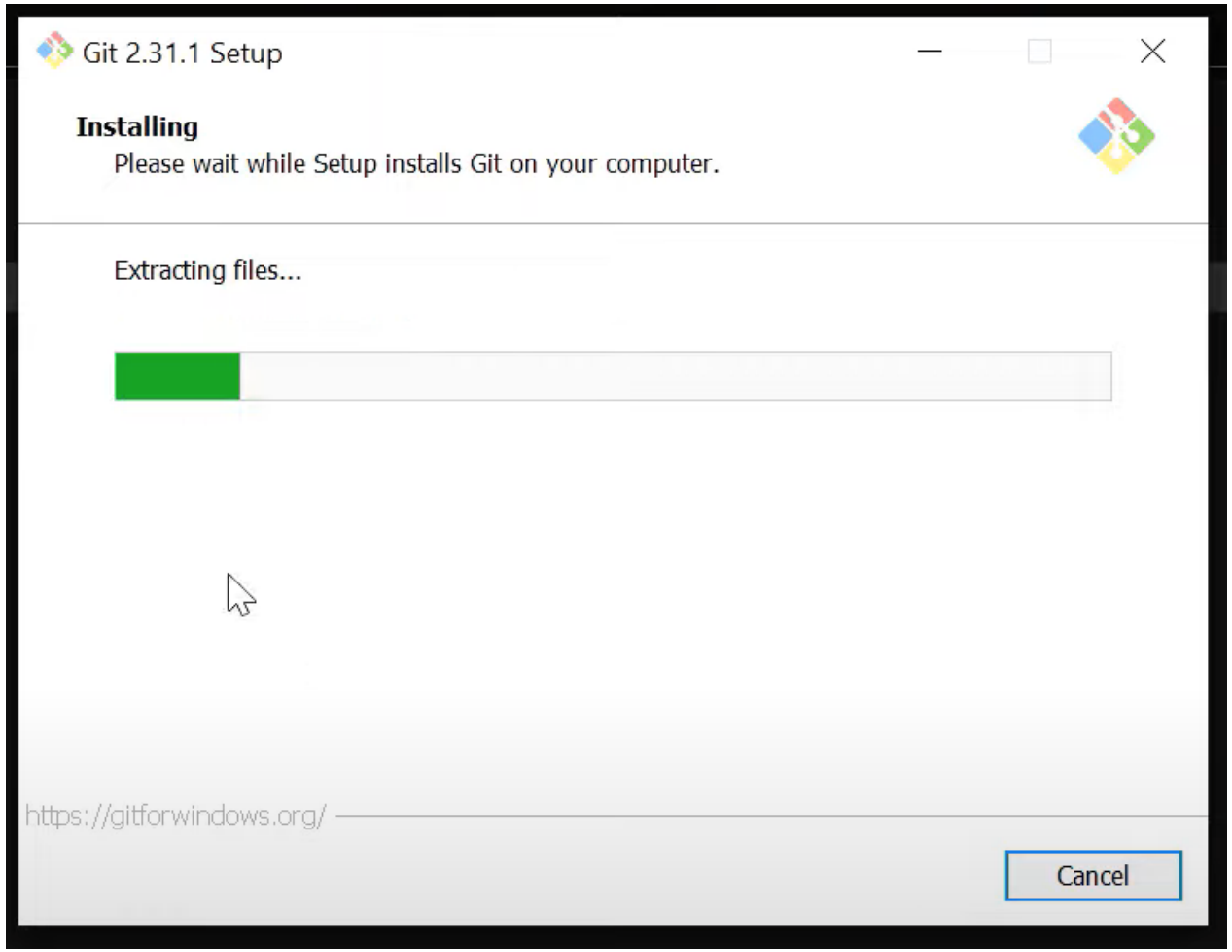

Git 2.31.1 Setup
Configuring the terminal emulator to use with Git Bash
Which terminal emulator do you want to use with your Git Bash?
Use MinTTY (the default terminal of MSYS2)
Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. 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

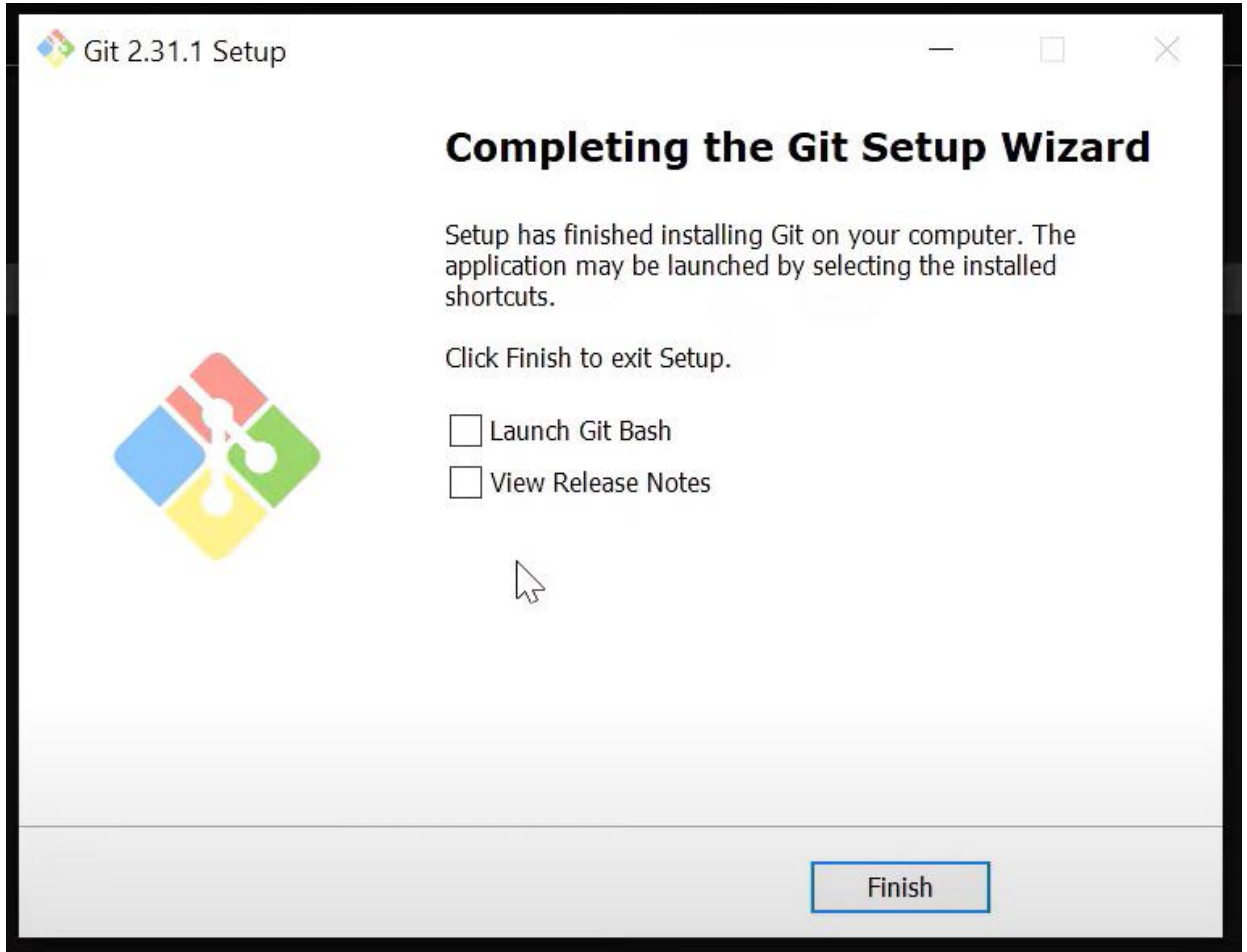

Git 2.31.1 Setup
Choose the default behavior of `git pull`
What should `git pull` do by default?
Default (fast-forward or merge)
This is the standard behavior of `git pull`: fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.
Rebase
Rebase the current 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


Git 2.31.1 Setup
Choose a credential helper
Which credential helper should be configured?
Git Credential Manager Core
(NEW!) Use the new, cross-platform version of the Git Credential Manager.
See more information about the future of Git Credential Manager here.
Git Credential Manager
(DEPRECATED) The Git Credential Manager for Windows handles credentials e.g. for Azure DevOps and GitHub (requires .NET framework v4.5.1 or later).
None
Do not use a credential helper.
https://gitforwindows.org/
Back
Next
Cancel

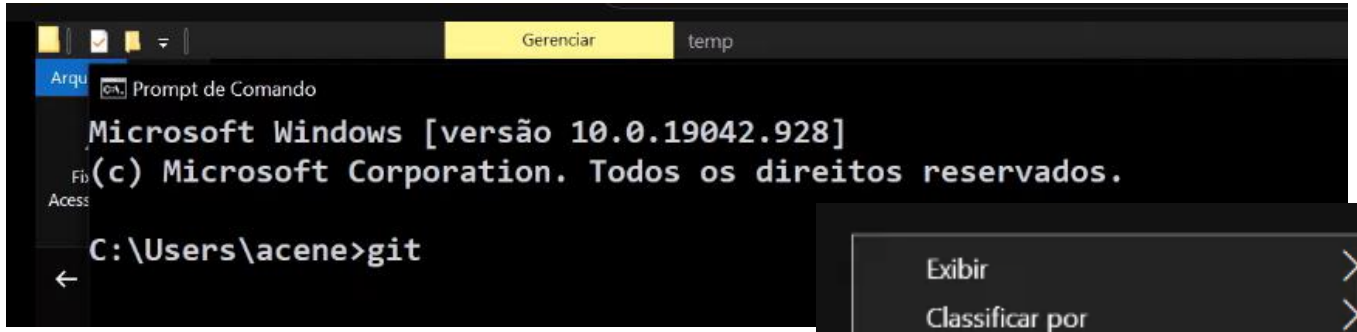

Git 2.31.1 Setup
Configuring extra options
Which features would you like to enable?
Enable file system caching
File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.
Enable symbolic links
Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.
https://gitforwindows.org/
Back
Next
Cancel


Git 2.31.1 Setup
Configuring experimental options
Which bleeding-edge features would you like to enable?
Enable experimental support for pseudo consoles.
(NEW!) This allows running native console programs like Node or Python in a Git Bash window without using winpty, but it still has known bugs.
https://gitforwindows.org/
Back
Install
Cancel

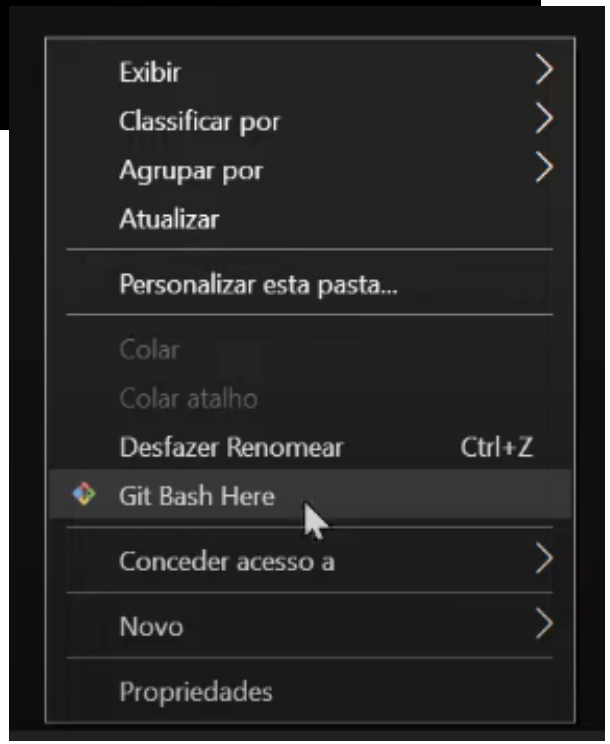

Git 2.31.1 Setup
Installing
Please wait while Setup installs Git on your computer.
Extracting files...
https://gitforwindows.org/
Cancel


Git 2.31.1 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

VERIFICAR SE FOI INSTALADO

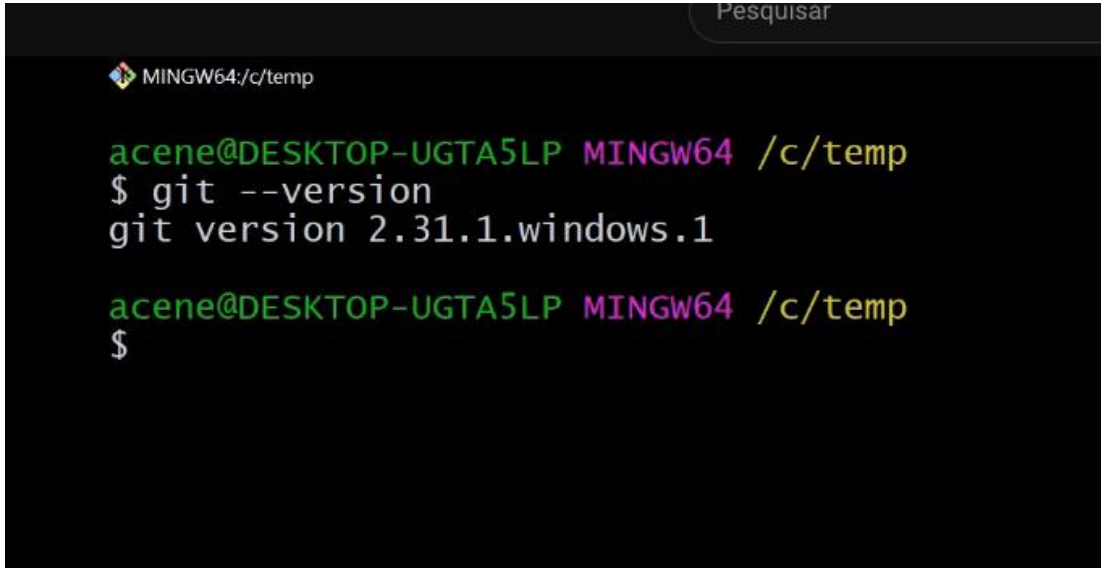


No cmd digitar git



Dentro diretório vija se aparece o Git Bash Here

No emulador do git, digitar git -- version



CRIANDO UMA CONTA NO GIT HUB

HTTPS://GIT-SCM.COM/

CONFIGURANDO O USUÁRIO NO GIT

CONFIGURANDO O GIT

- **Abra o git bash**
- **Digite os comandos:**
  - git config --global user.name "seu nome"
  - git config --global user.email "seu email"
  - git config --list

CONFIGURAÇÃO PARA VER ARQUIVOS OCULTOS (WINDOWS)

Iniciar -> Opções do explorador de arquivos

DESMARCAR: "Ocultar as extensões dos tipos de arquivos conhecidos"

MARCAR: "Mostrar arquivos, pastas e unidades ocultas"

CHAVE SSH PARA O GITHUB

- SSH é um protocolo para comunicação de dados com segurança.
- A ideia é cadastrar previamente quais computadores podem acessar o github em seu nome.

1. Gerar uma chave SSH no seu computador.
2. Cadastrar essa chave no seu Github

PASSOS

- Digite no google: github ssh
- Ir para gerando um nova chave ssh ..
- https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent
- Seguir os passos do doc
- $ ssh-keygen -t ed25519 -C "your_email@example.com" (colocar o e-mail de cadastro do github)

PASSOS

```
acene@DESKTOP-UGTA5LP MINGW64 /c/temp
$ ssh-keygen -t rsa -b 4096 -C "[illegible]@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/[illegible]/.ssh/id_rsa):
Created directory '/c/Users/[illegible]/.ssh'.
Enter passphrase (empty for no passphrase):
```

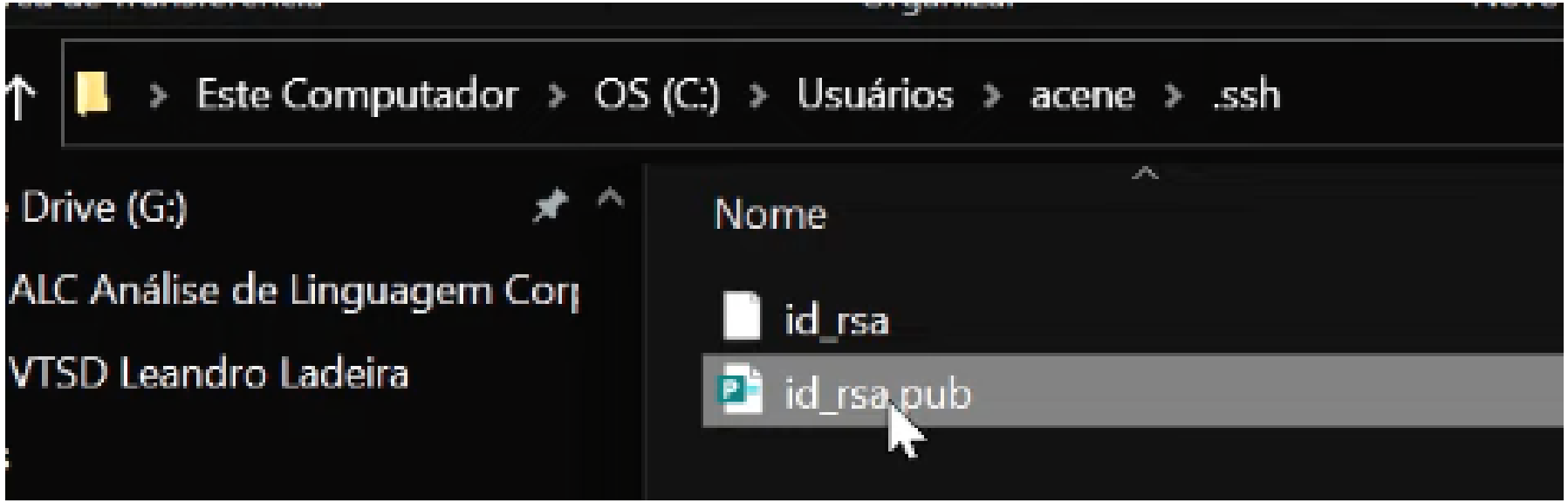
Caso queira colocar uma senha é neste momento, caso contrário é só da enter

```
acene@DESKTOP-UGTA5LP MINGW64 /c/temp
$ ssh-keygen -t rsa -b 4096 -C "acenelio@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/acene/.ssh/id_rsa):
Created directory '/c/Users/acene/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/acene/.ssh/id_rsa
Your public key has been saved in /c/Users/acene/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:8eVvDYpfuutCoQVn9rEUqU8rf9aOqIYxCDCVB2+gtDY acenelio@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
|   ..+o      .o  |
| .oo.o. . + +    |
|  Eo .o .= +.o   |
| . ...   o+o+    |
|      . .So.+... |
|       . + o.oo o|
|          =.o. +.|
|         . o..=o.|
|          ..+B=..|
+----[SHA256]-----+

acene@DESKTOP-UGTA5LP MINGW64 /c/temp
$
```

PASSOS

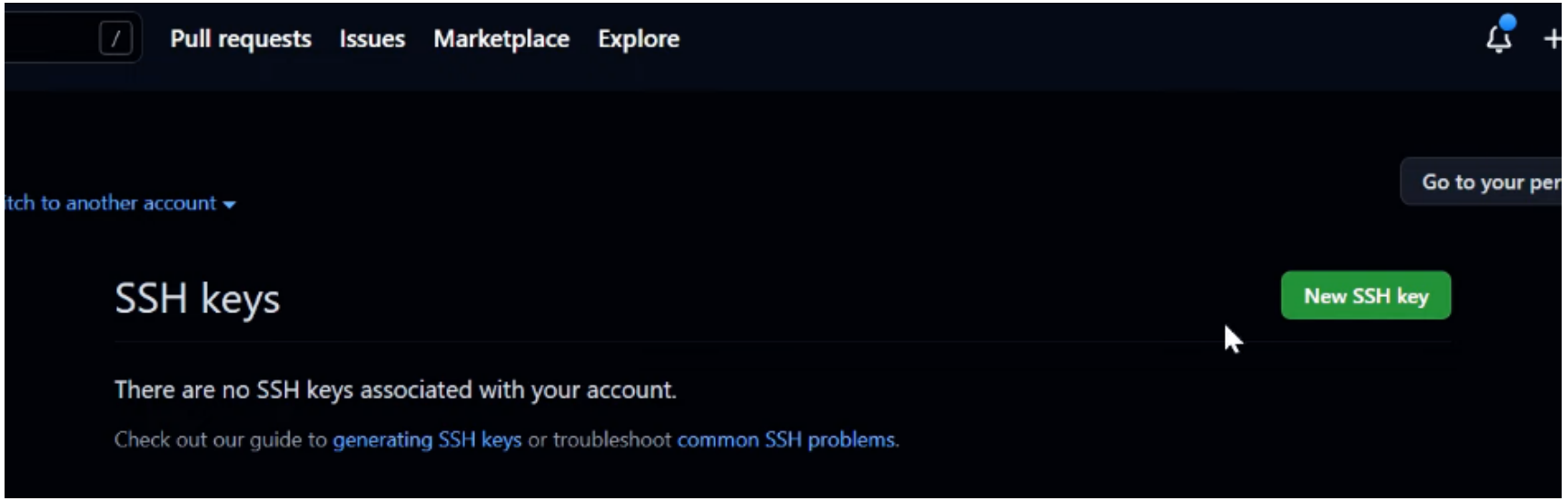
Entrar no local da pasta onde foi gerado a chave.

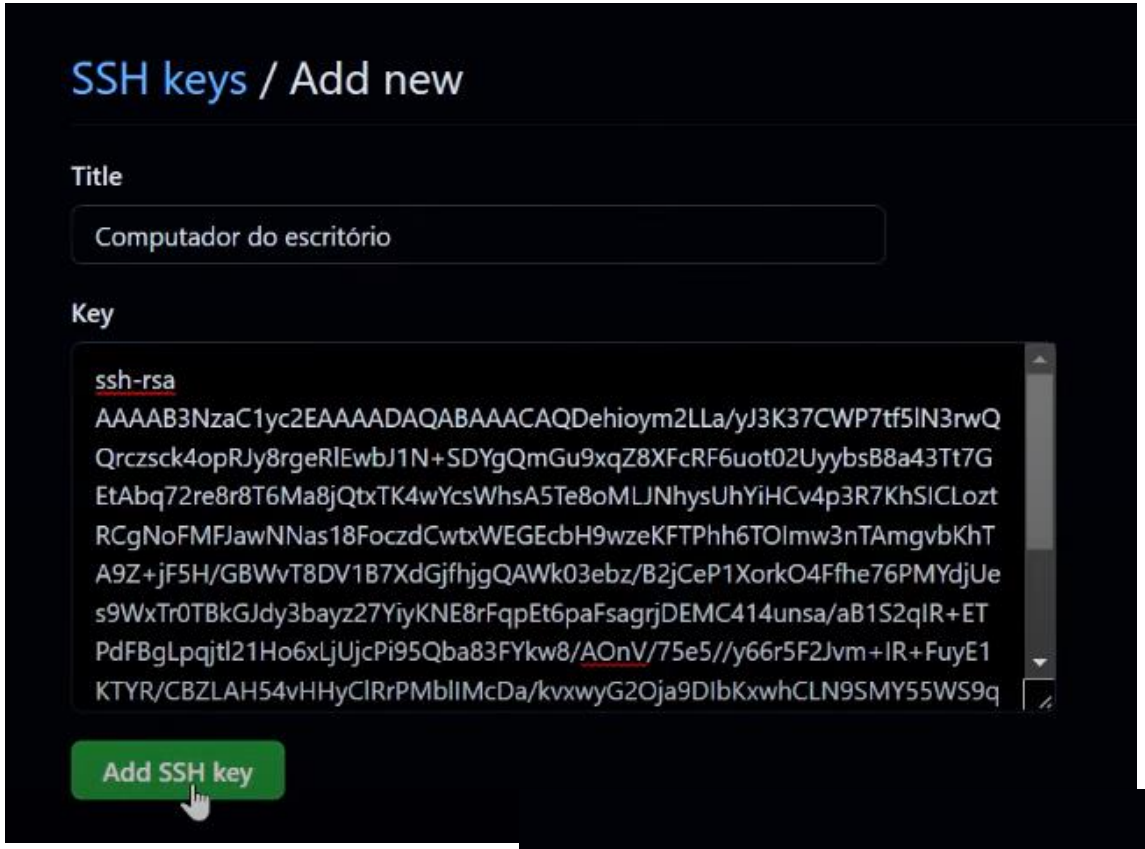


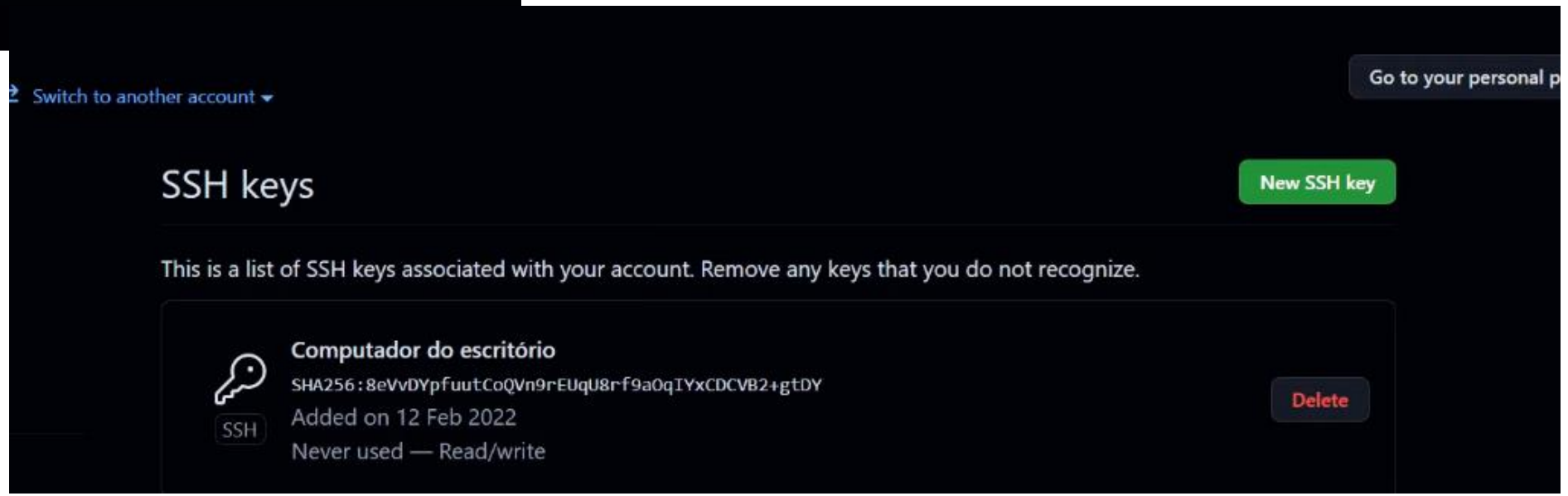
Abrir o público, pegar a chave e colocar no github

PASSOS NO GITHUB

- No perfil vai em configurações (settings)
- Procurar o SSH and GPG




SSH keys / Add new
Title
Computador do escritório
Key
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQDehioym2LLa/yJ3K37CWP7tf5lN3rwQ
QrczscK4opRJy8rgeRlEwbJ1N+SDYgQmGu9xqZ8XFcRF6uot02UyybsB8a43Tt7G
EtAbq72re8r8T6Ma8jQtxTK4wYcsWhsA5Te8oMLJNhysUhYiHCv4p3R7KhSlCLozt
RCgNoFMFJawNNas18FoczdCwtxWEGEcbH9wzeKFTPhh6TOlmw3nTAmgvbKhT
A9Z+jF5H/GBWvT8DV1B7XdGjfhjgQAWk03ebz/B2jCeP1XorkO4Ffhe76PMYdjUe
s9WxTr0TBkGJdy3bayz27YiyKNE8rFqpEt6paFsagrjDEMC414unsa/aB1S2qIR+ET
PdFBgLpqjtl21Ho6xLjUjcPi95Qba83FYkw8/AOnV/75e5//y66r5F2Jvm+IR+FuyE1
KTYR/CBZLAH54vHHyClRrPMblIMcDa/kvxwyG2Oja9DIbKxwhCLN9SMY55WS9q
Add SSH key


Switch to another account
Go to your personal p
SSH keys
New SSH key
This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.
Computador do escritório
SHA256:8eVvDYpfuutCoQVn9rEUqU8rf9aOqIYxCDCVB2+gtDY
SSH
Added on 12 Feb 2022
Never used — Read/write
Delete


PRIMEIRA VERSÃO
Do computador para a nuvem

PASSO A PASSO: SALVAR PRIMEIRA VERSÃO

- Abra um terminal dentro do projeto, pode ser no Vs code ou no diretório com o bash.

PASSO A PASSO: SALVAR PRIMEIRA VERSÃO

- Toda vez que você criar um novo projeto, esses serão os passos:
- No terminal Digite:

  ➉ git init (cria um repositório na pasta)

  ➉ git add . (envia os arquivos para uma área temporária "stage")

  ➉ git commit –m "Mensagem explicativa" (é o que vai salvar)

  ➉ git branch –M main (garantir que está salvo no main)

  ➉ git remote add origin git@github.com: seuusuario/seurepositorio.git

    ➉ Neste passo criar um projeto no github, para poder enviar para a plataforma.

    ➉ Associa a máquina com a plataforma (escolher o ssh)

    ➉ Se não quiser fazer com o ssh, neste momento é só copiar a url do repositório e colar depois de origin

  ➉ git push –u origin main (envia para o git)

PASSO A PASSO: SALVAR UMA NOVA VERSÃO

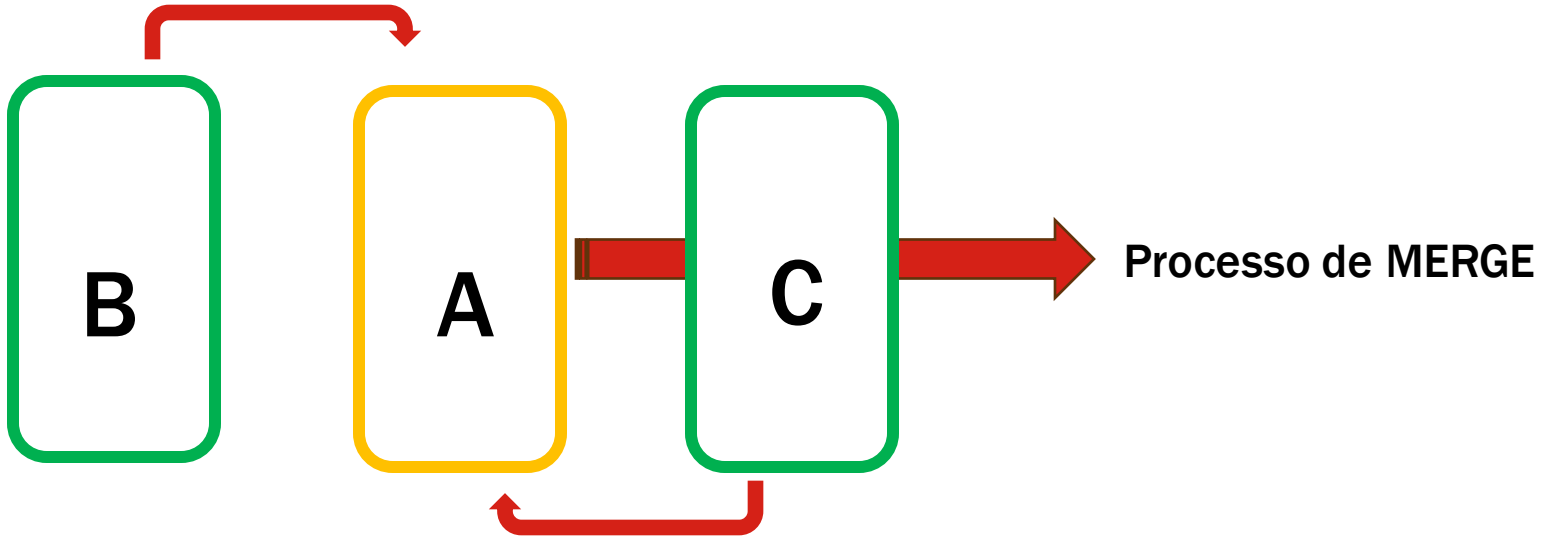
- git status (mostra como está o projeto)
- git add .
- git commit –m "mensagem explicativa"
- git push (envia para o Github)

É possível ver no VS Code.

TRABALHANDO COM O BRANCH

BRANCH

- É criado outras versões do código, para não prejudicar a principal, criando uma branch paralela
- Ideal para se trabalhar em equipe.



PASSO A PASSO – BRANCH ESTÁVEL

- git branch (para ver as branchs em execução)
- git branch teste (staging – criando uma nova branch)
- git branch (rode o comando mais uma vez para ver o que tem e qual é o principal – com asterisco e estará em verde)
- git checkout teste (para mudar de branch)
- git status (para ver se está na branch correta)
- Modifique o arquivo e faça o commit (slide 33)

TRABALHANDO COM ATUALIZAÇÃO DA BRANCH MAIN

- Entrar na branch que receberá as atualizações (git checkout)
- git pull (faz as atualizações da plataforma em sua máquina)
- git merge teste (puxando as atualizações da branch teste para a principal)
  - Obs: precisa estar dentro da branch que irá receber a atualização.

- git push (subir na plataforma)

CRIANDO UMA NOVA BRANCH

- git branch
- git pull
- git checkout –b sistema-de-login (crindo uma branch para o sistema de login)
- Crie o código
- git add . (adicionando o arquivo)
- git commit –m "mensagem"
- Atualize na branch principal (slide 37)

GIT IGNORE

Não sobe determinadas pastas ou arquivos.

- touch .gitignore (irá criar um arquivo na pasta, neste aquivo clicar com o botão direito e colocar os arquivos que eu não quero subir...exemplos fotos/ (pasta) ou fotos.png (arquivo)
- Fazendo um git status, ele não irá encontrar esses arquivos.
- git commit –m "add o arquivo git ignore"
- git push

GIT CLONE

- Vá no gihub e veja o que vc quer clonar.
- Clique no botão clone ou download
- Copie o https
- Crie uma pasta no seu computador
- git clone "cole o http"
- Ou direto no Vs code quando entrar ..

FIM

VÍDEOS REFERÊNCIA

- https://www.youtube.com/watch?v=kB5e-gTAI_s
  Merge
- https://www.youtube.com/watch?v=_hZf1teRFNg
  Commite e instalação
- https://www.youtube.com/watch?v=OIArEishhQg
  clone - Guanabara