



GIT e GitHub

Angelo Cesar Colombini

Mas, afinal do que se trata e
por quê preciso conhecer
isso?

Instalando o GIT

Preparando seu ambiente!



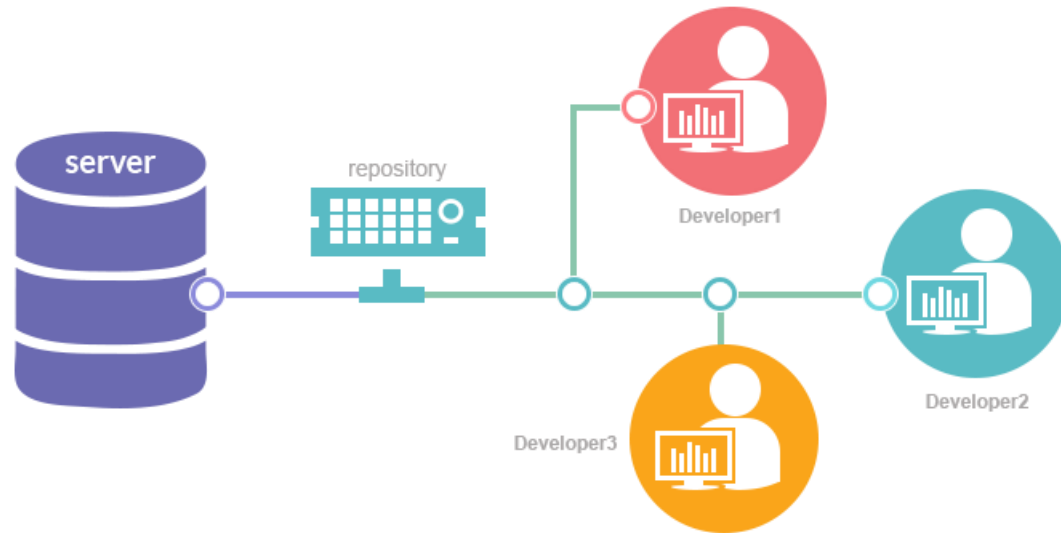
Git & GitHub → Instalando e configurando o GIT

- Passo 1: baixar o Git no site → git-scm.com
- Passo 2: baixar o Visual Studio Code → code.visualstudio.com
- **Iniciemos pelo Git**



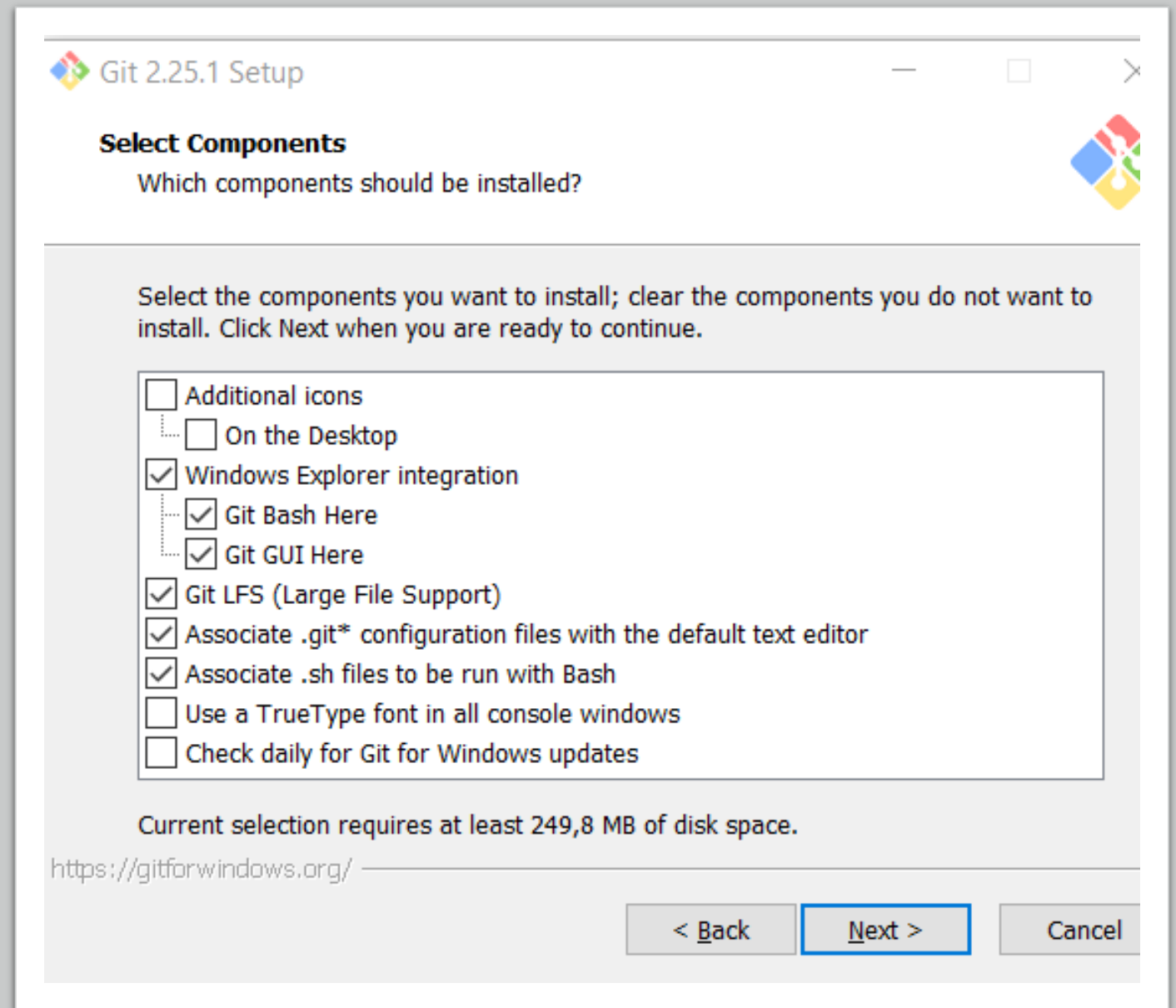
Git & GitHub → Instalando e configurando o GIT

- Passo 3: Iniciando a instalação do Git



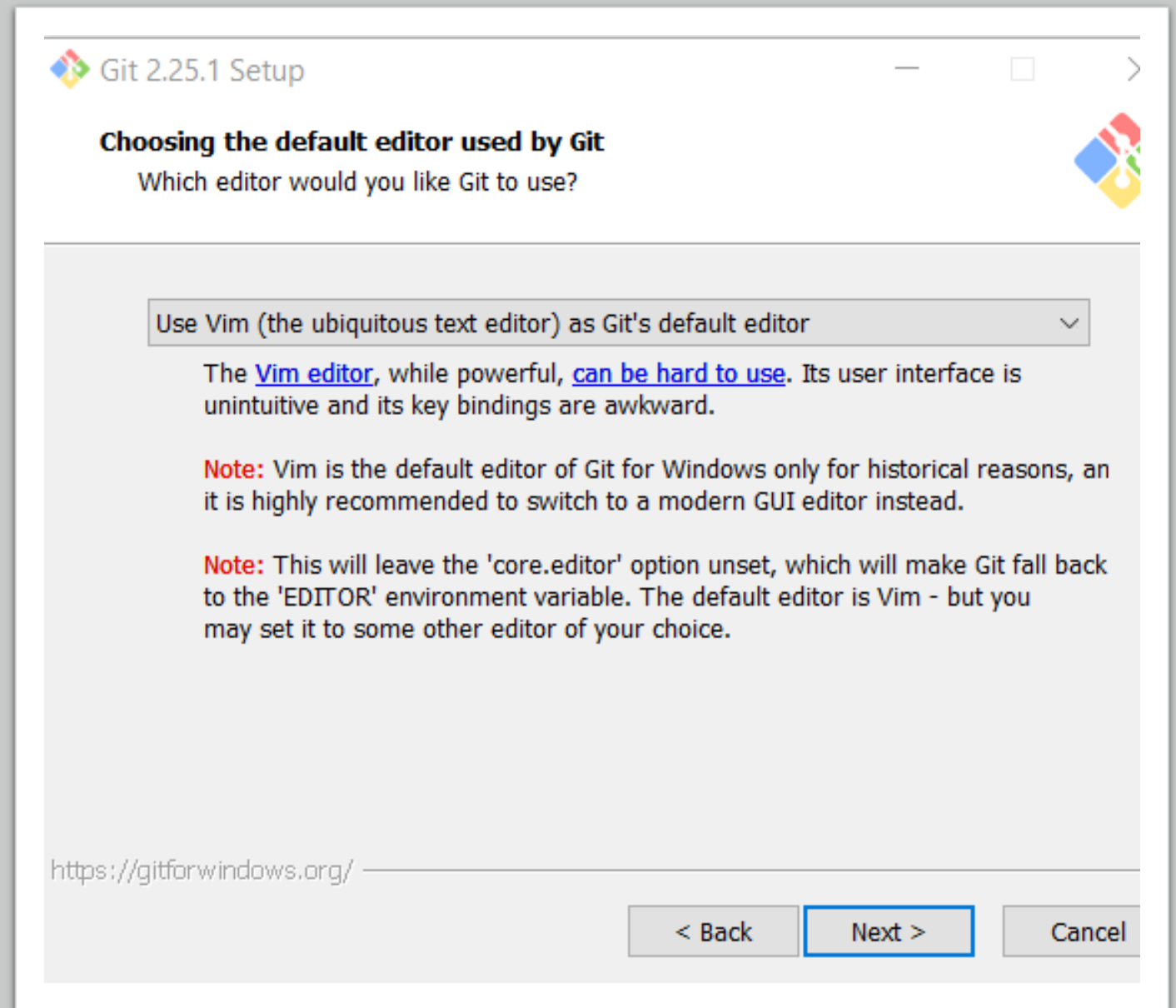
Git & GitHub →
Instalando e
configurando o GIT

Passo 4: após
aceitarmos os
termos da licença →
estamos prontos
para iniciar.



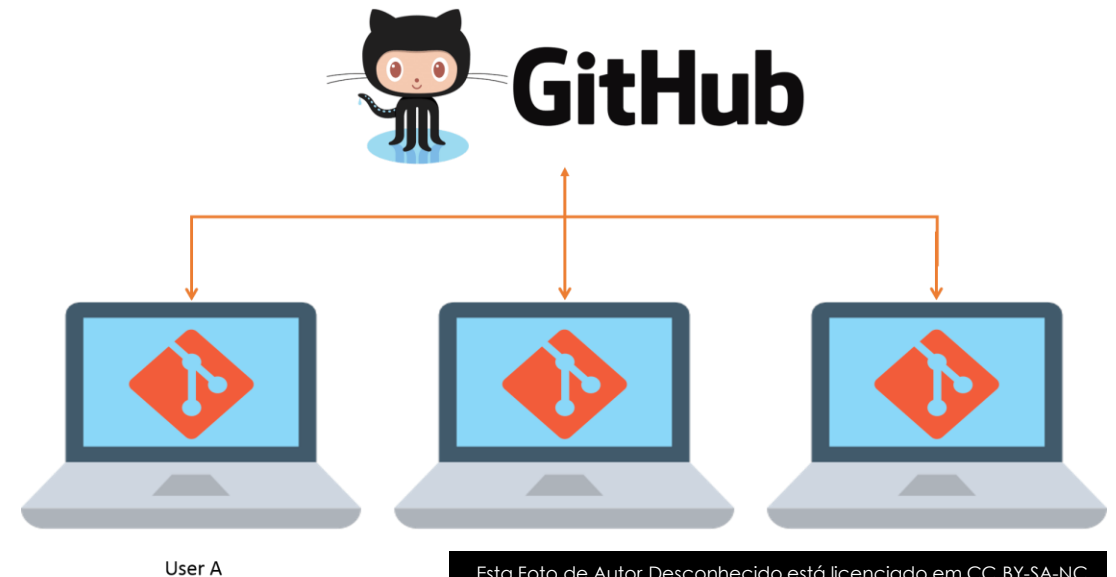
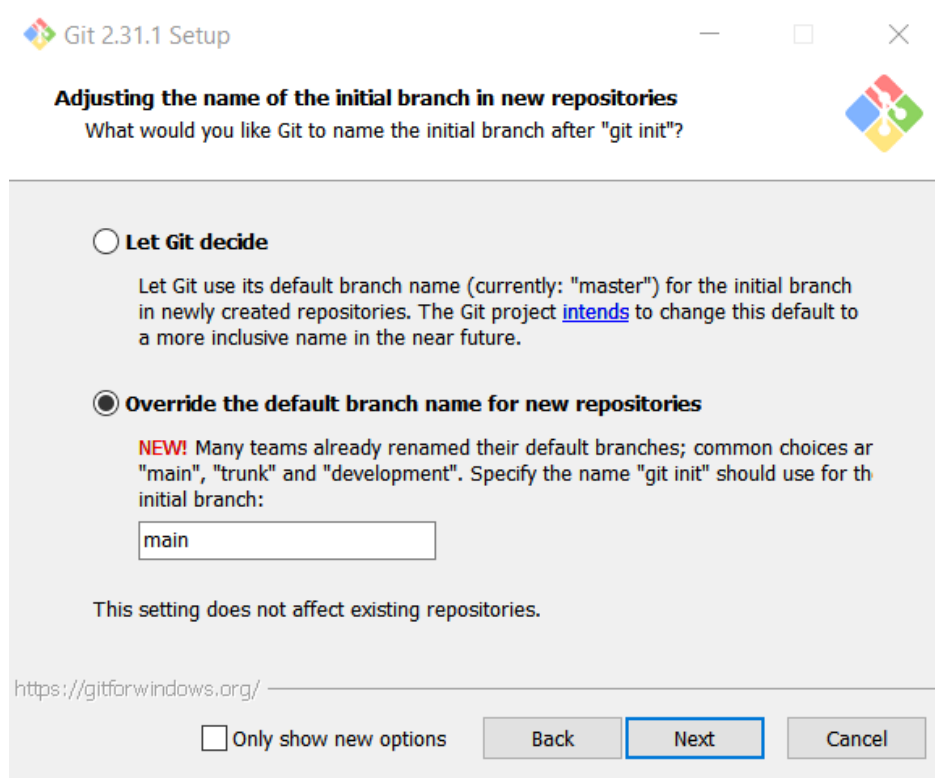
Git & GitHub →
Instalando e
configurando o GIT

Passo 5: Vamos
manter a
configuração de
Editor padrão →
poderá ser alterada
mais tarde, caso
queira.



Git & GitHub → Instalando e configurando o GIT

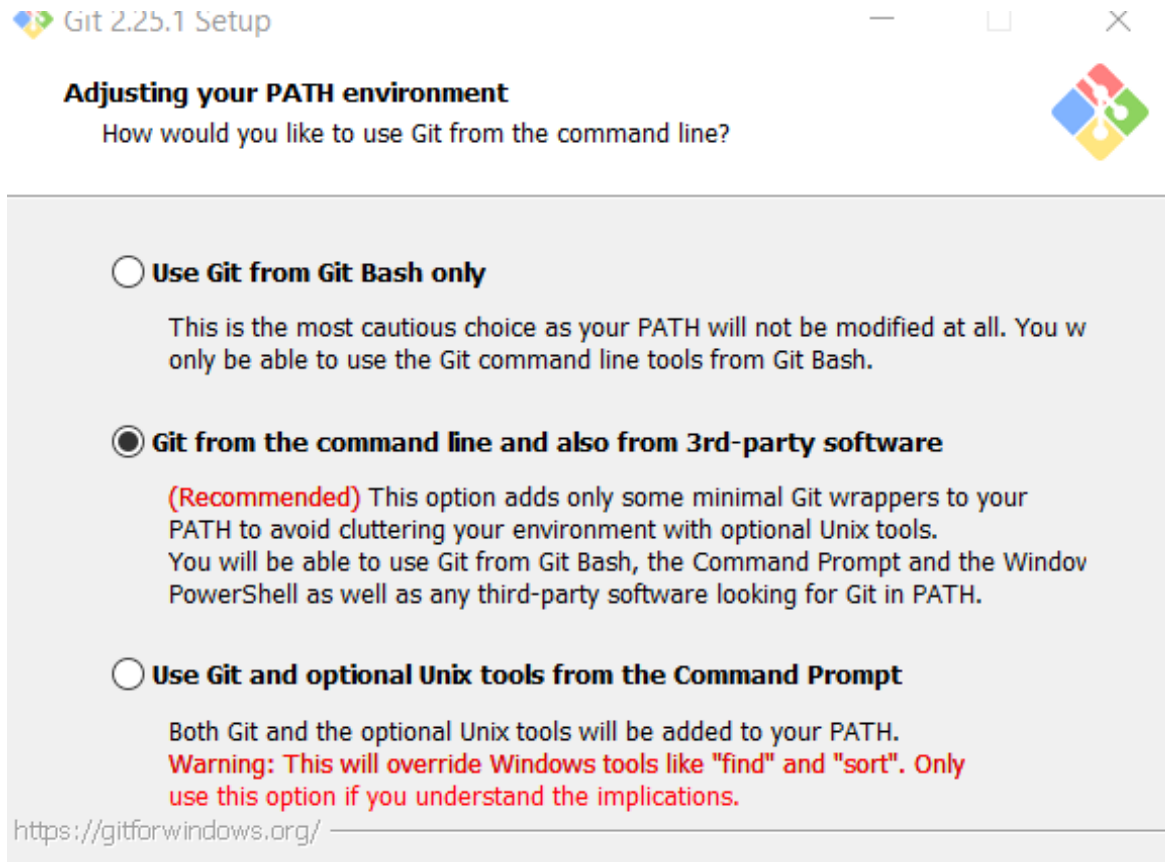
Passo 6: opte pela criação da branch main.



Esta Foto de Autor Desconhecido está licenciado em [CC BY-SA-NC](#)

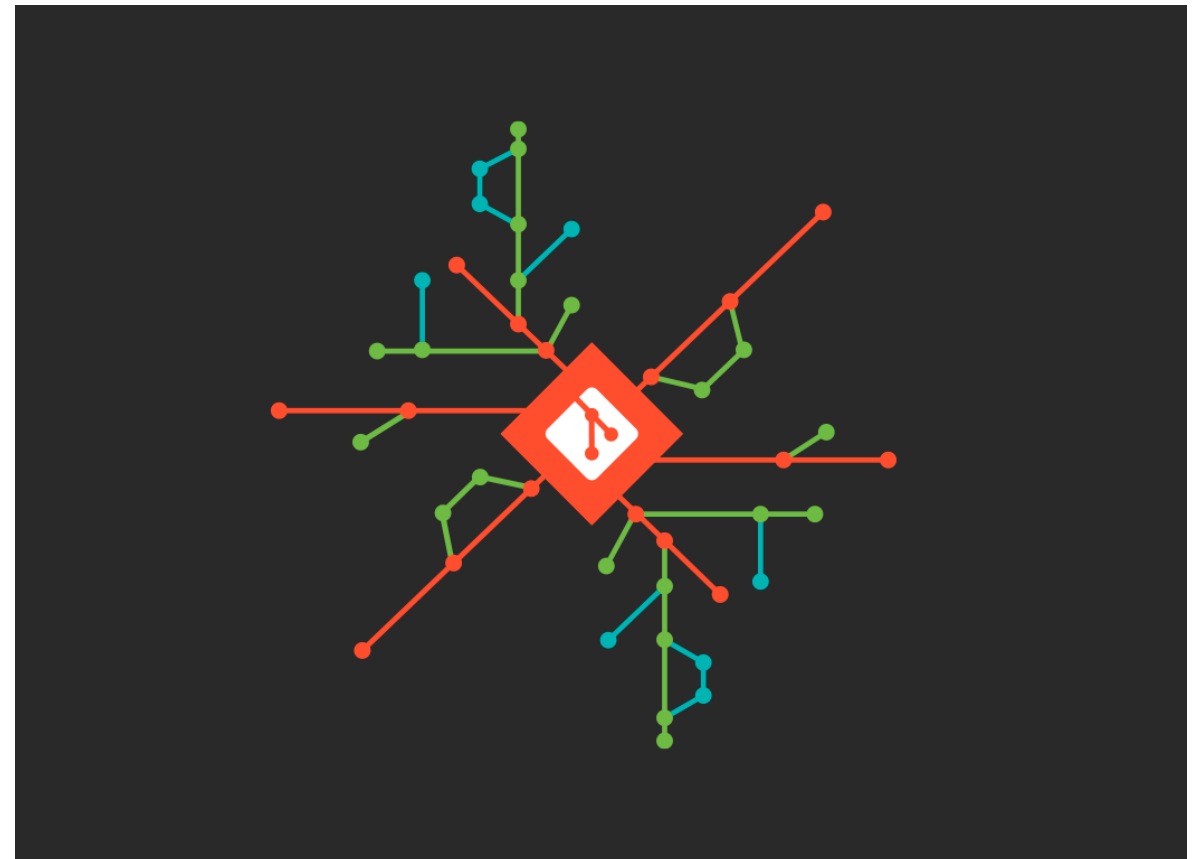
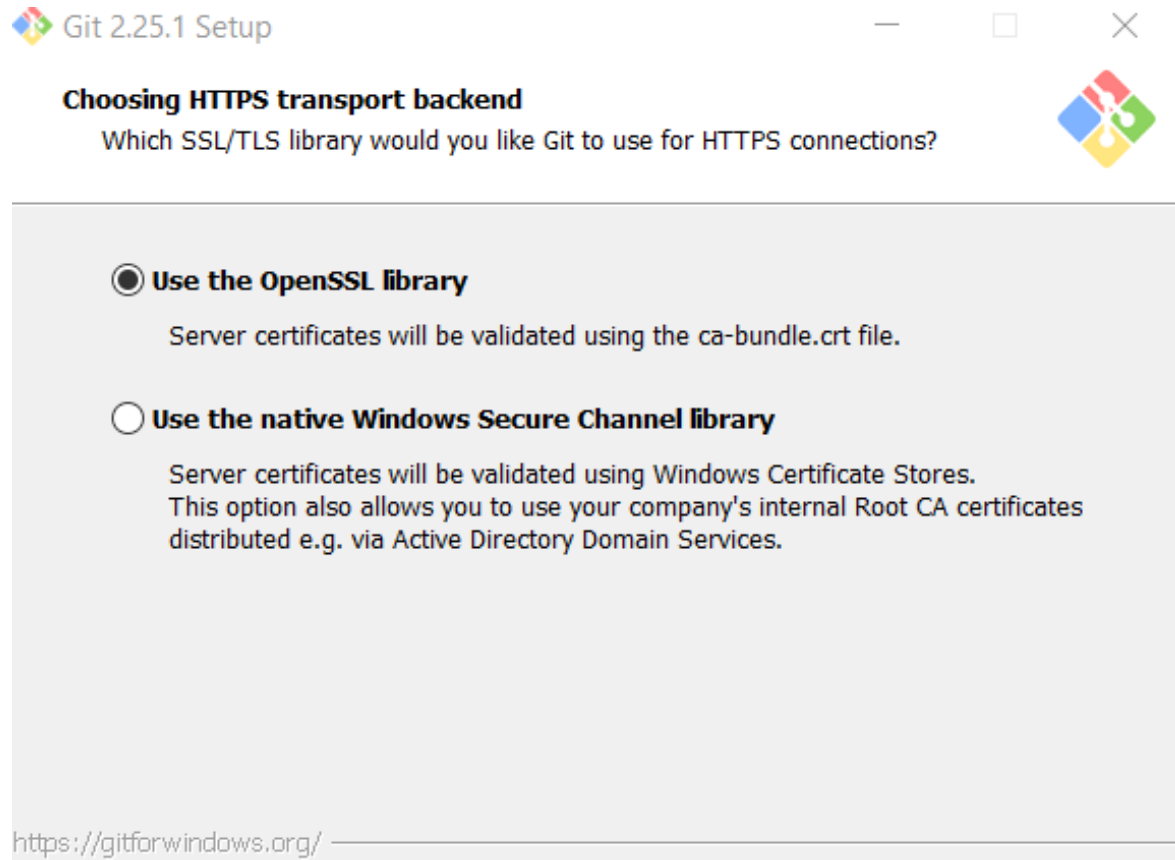
Git & GitHub → Instalando e configurando o GIT

Passo 7: o Git é um software de linha de comandos, então fiquemos com o padrão.



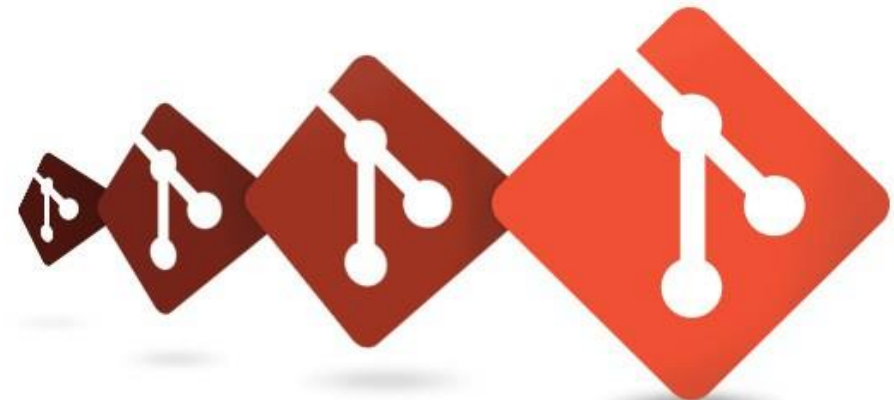
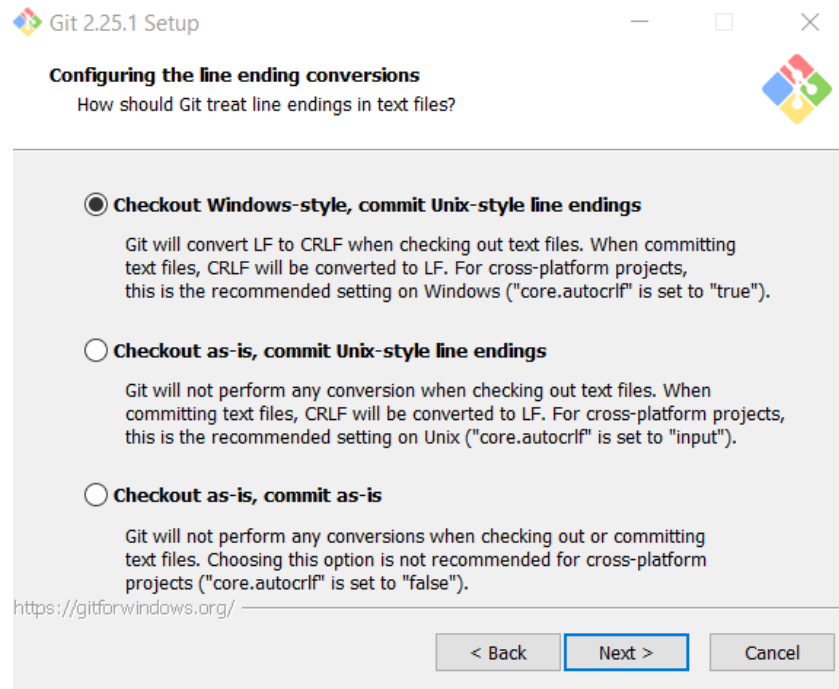
Git & GitHub → Instalando e configurando o GIT

Passo 8: Precisamos do SSL para fazermos conexões seguras



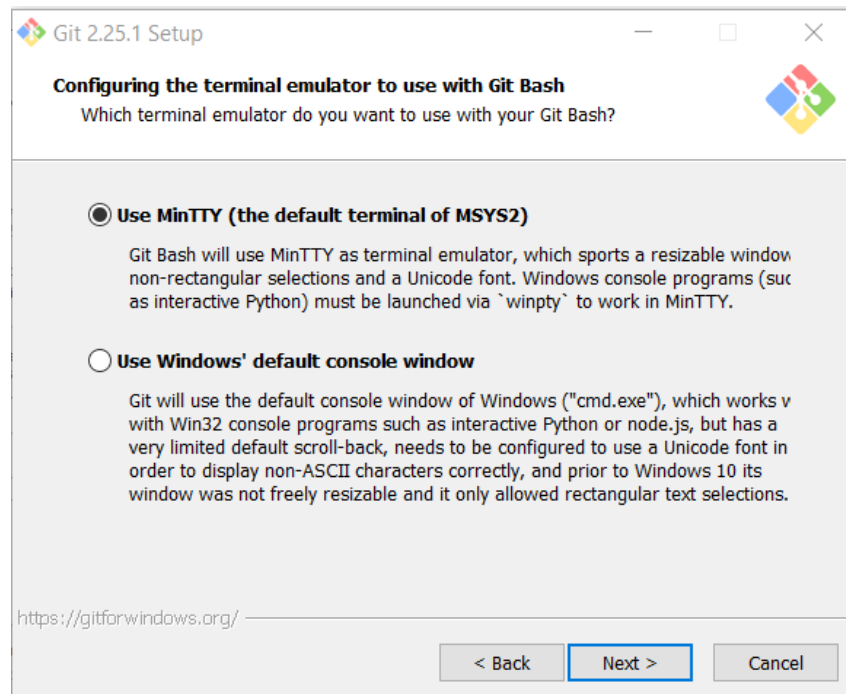
Git & GitHub → Instalando e configurando o GIT

Passo 9: Aqui é um ponto que o Windows se diferencia do Linux e do MAC → é preciso definir como o Git deverá interpretar final de uma linha de código



Git & GitHub → Instalando e configurando o GIT

- Passo 10: seguimos no default poderemos usar o git bash como terminal, veremos isso no Visual Studio Code

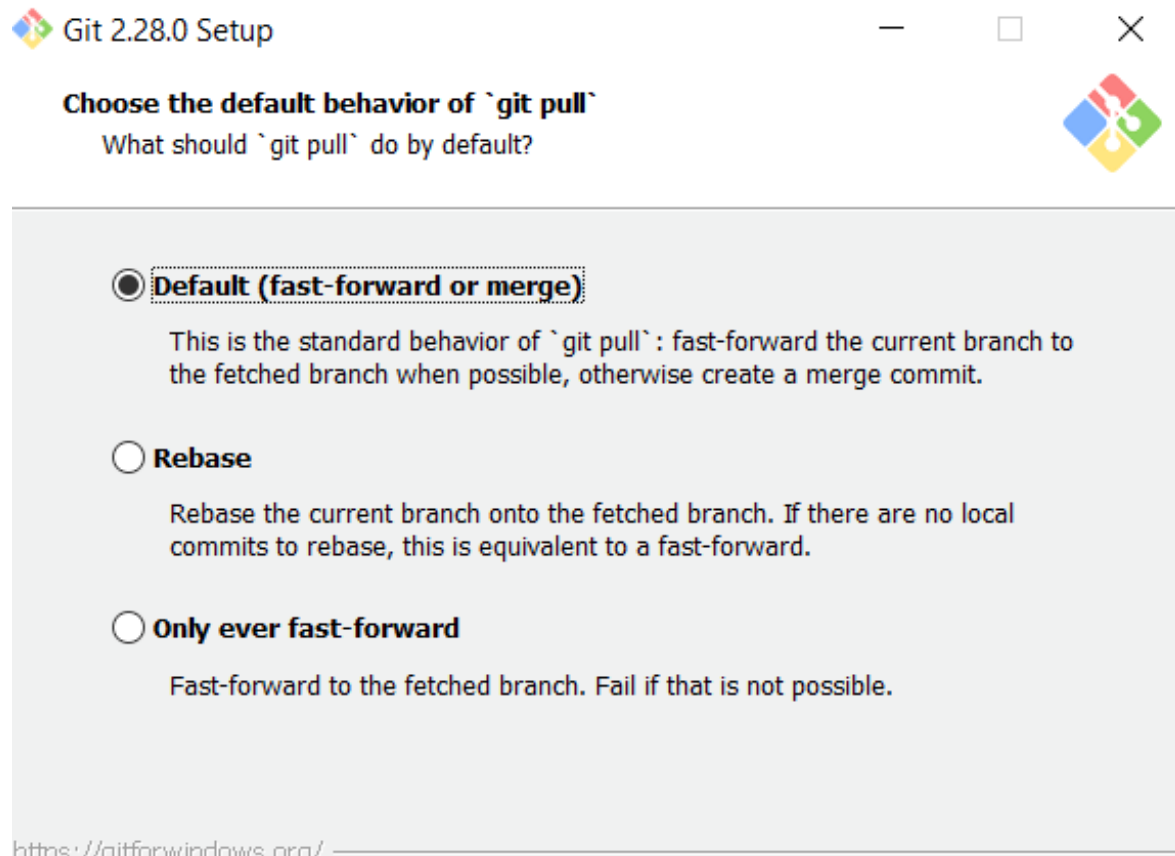


Git & GitHub → Instalando e configurando o GIT

Passo 11: nada a acrescentar aqui

Git & GitHub → Instalando e configurando o GIT

Passo 11: nada a acrescentar aqui



Git 2.28.0 Setup

Choose a credential helper

Which credential helper should be configured?

☐ None

Do not use a credential helper.

☒ **Git Credential Manager**

The [Git Credential Manager for Windows](#) handles credentials e.g. for Azure DevOps and GitHub (requires .NET framework v4.5.1 or later).

☐ **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](#).

<https://gitforwindows.org/>

☐ Only show new options

< Back

Next >

Passo 12: Siga no default

Git & GitHub →
Instalando e
configurando o
GIT

Git & GitHub →
Instalando e
configurando o GIT

Passo 13: Seleccionamos as três
opções a seguir.



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 Git Credential Manager**

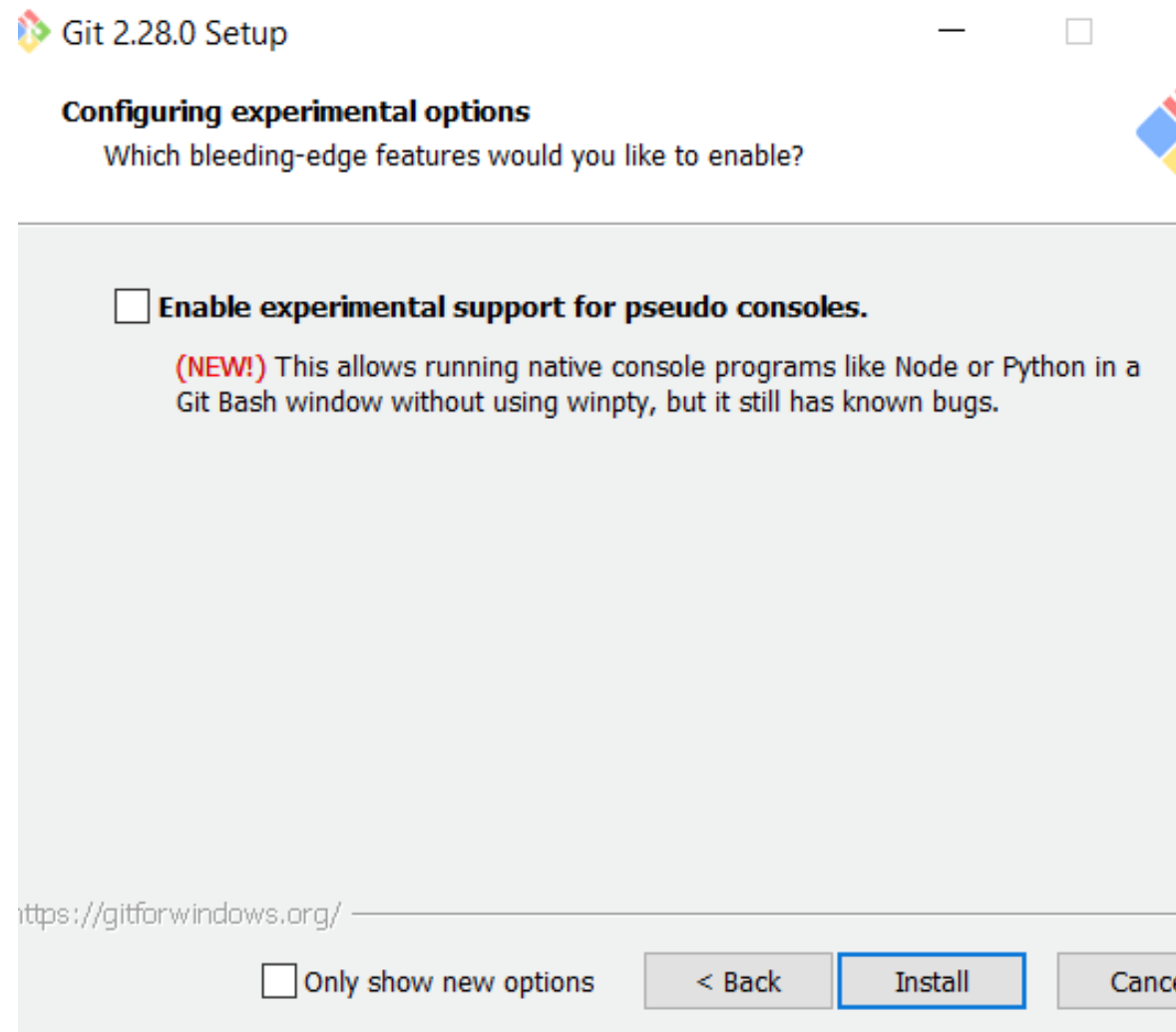
The [Git Credential Manager for Windows](#) provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or later).

☒ **Enable symbolic links**

Enable [symbolic links](#) (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

Passo 14: Deixe em branco e siga em frente

Git & GitHub →
Instalando e
configurando o
GIT



Git & GitHub →
Instalando e
configurando o GIT

Passo 15: Fim → a seguir
configurando o **Visual Studio
Code**. Atenção você pode usar o terminal de
sua preferência

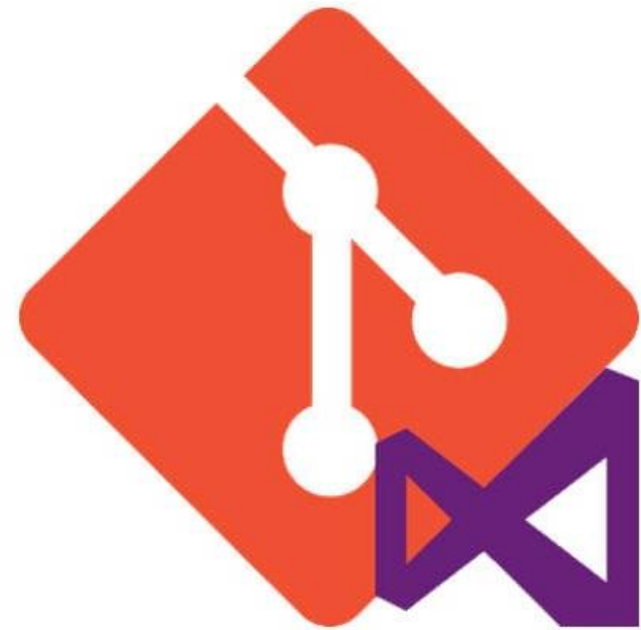
Git 2.25.1 Setup

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



Git & GitHub → Instalando e configurando o GIT

Baixar e instalar o Visual Studio Code

Processo de Instalação normal ... Padrão Windows

Lembrando → o Git é um software de linha de comando, o que demanda o uso de um **terminal**. É aí que entra o **Visual Studio Code**

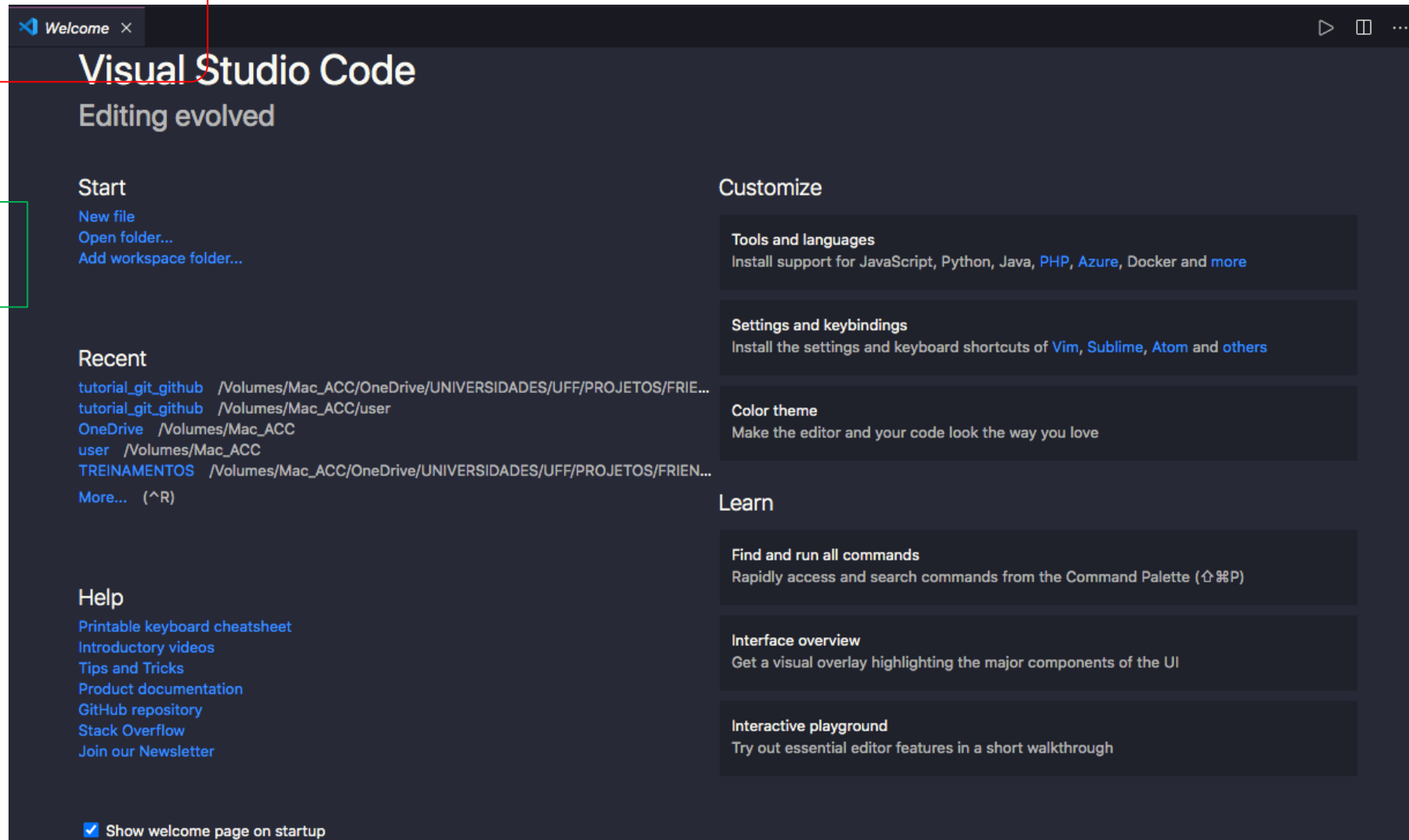
Preparar o **Terminal** para trabalhar com Git, não é obrigatório, mas facilita bastante, veja como:



Ao abrir o VsCode você terá a seguinte tela

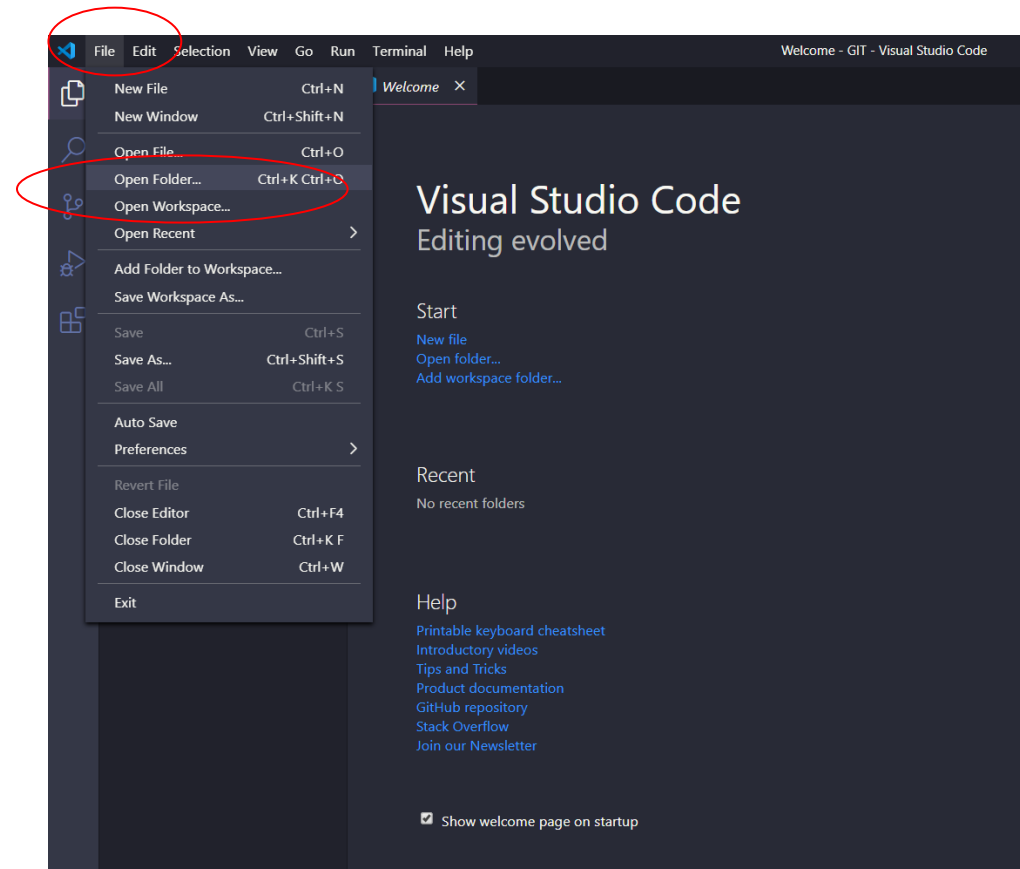


Feche essa
aba



Git & GitHub → Instalando e configurando o GIT

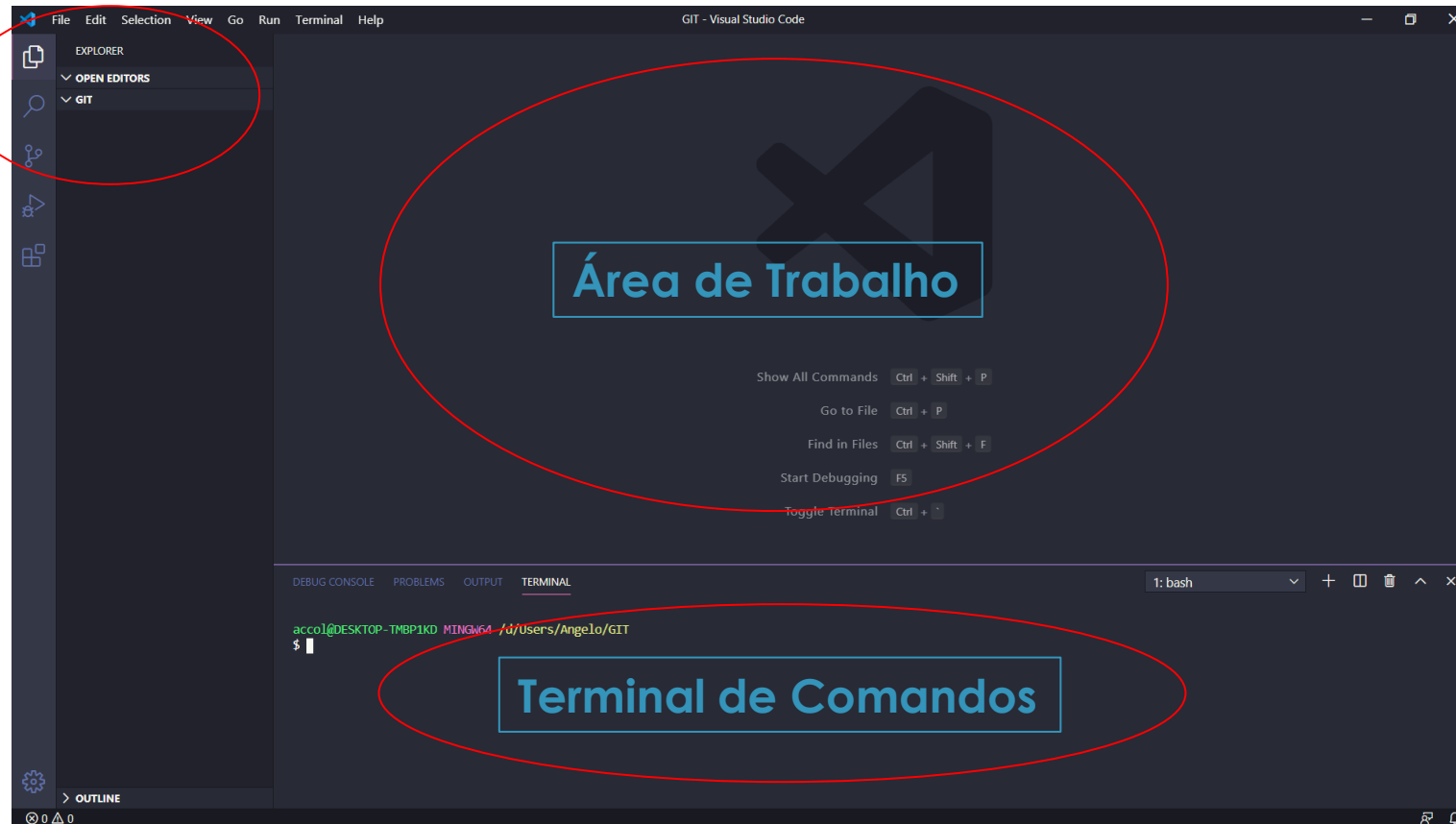
- Passo 16: O próximo passo é criar uma pasta de trabalho. No meu caso, criei como exemplo, a pasta GIT. Feito isso, vamos abri-la no Visual Studio Code.



Git & GitHub → Instalando e configurando o GIT

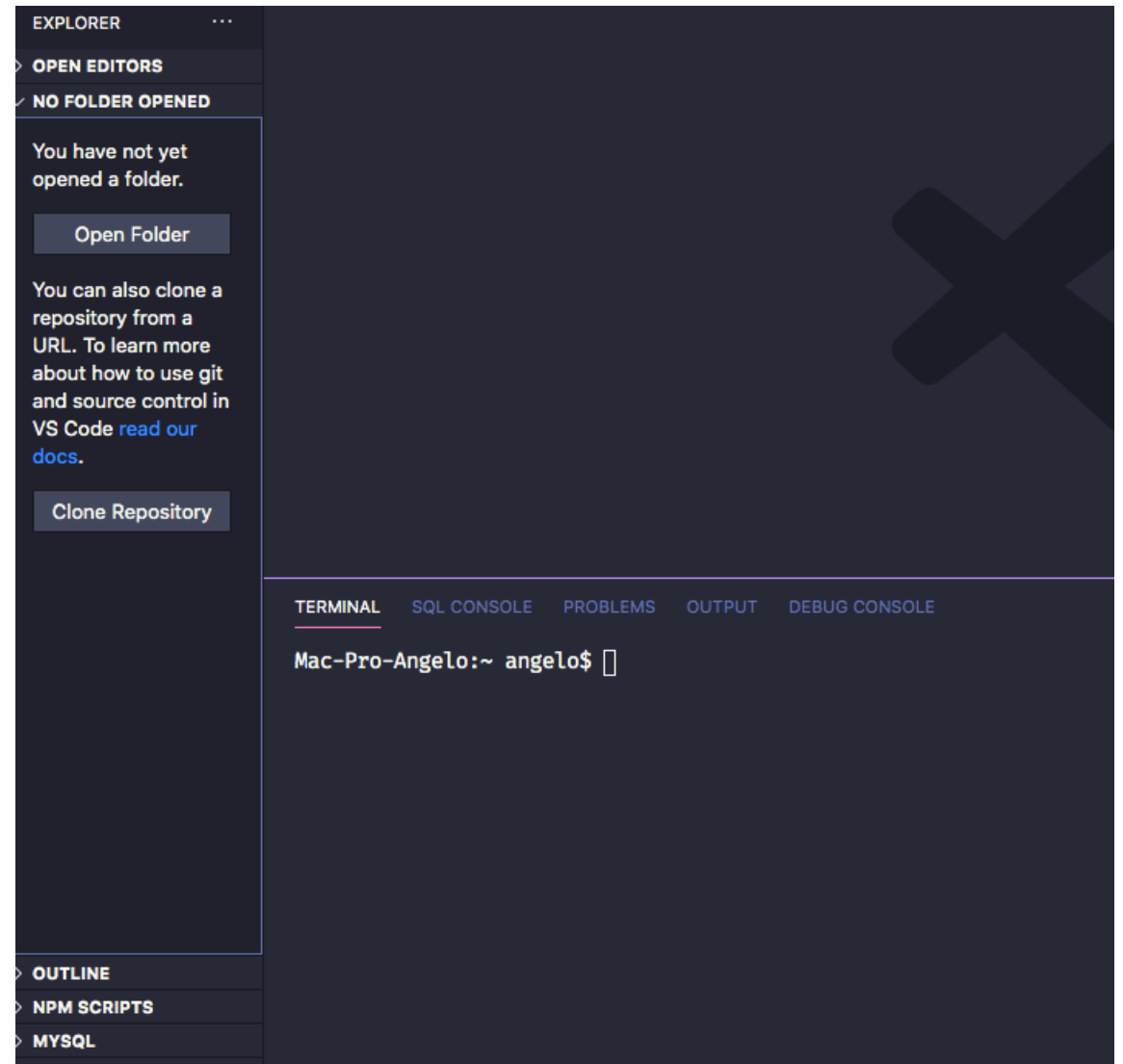
- Passo 17 Agora estamos prontos para trabalhar criando novos arquivos e acessando o terminal do Visual Studio Code.

Criação



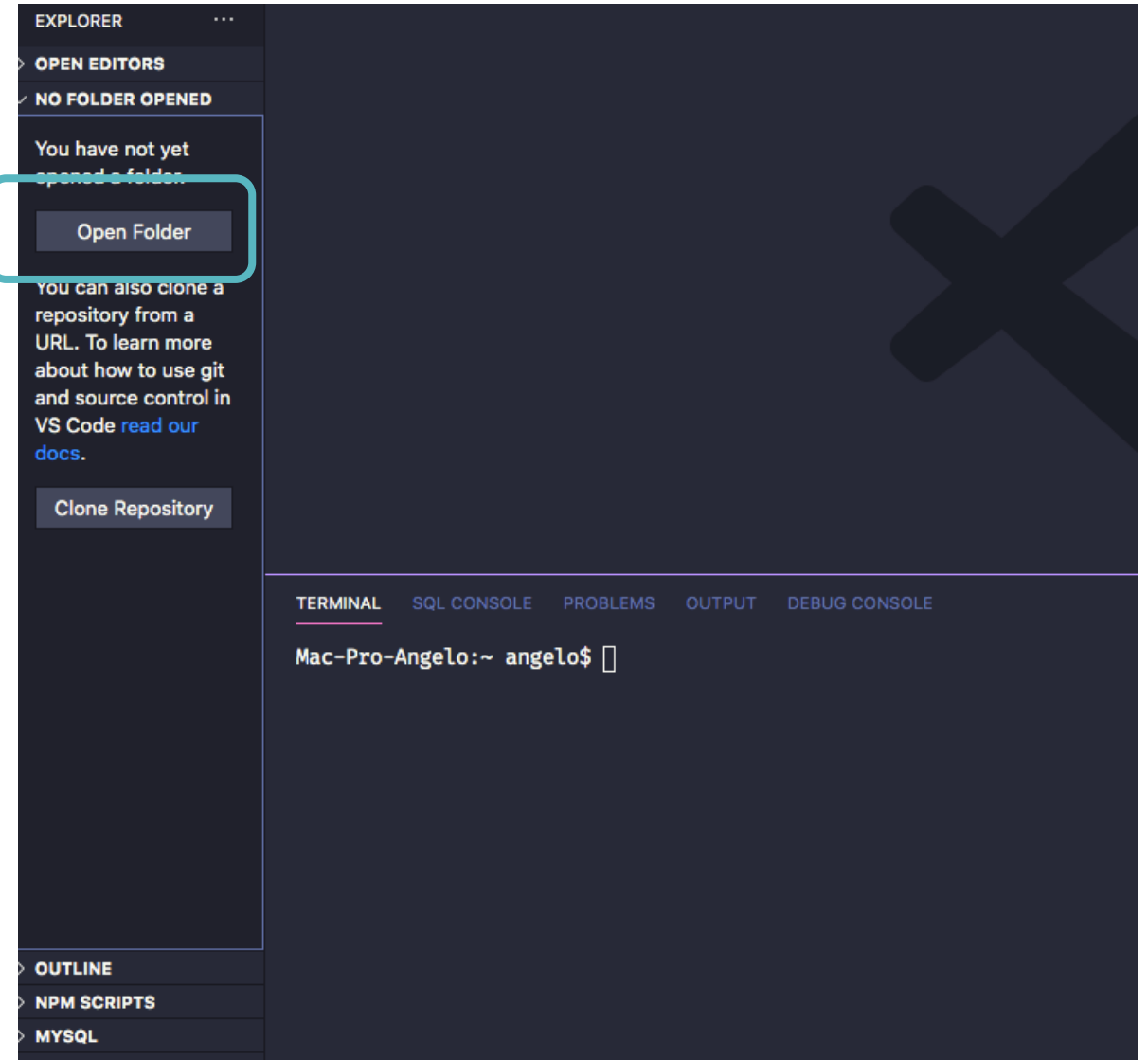
Git & GitHub → Instalando e configurando o GIT

- Abra seu VsCode, fecha a aba Welcome.
- Pressione CTRL + SHIFT + ' (no MAC)
- Pressione CTRL + ' (no Windows)
- Você terá algo como:



Git & GitHub → Instalando e configurando o GIT

- Neste momento você deverá abrir seu diretório de trabalho, no caso, criei um diretório chamado tutorial_git_github, mas lembre-se, você terá seus diretórios de trabalho.
- Use essa tela de início como atalho, veja:
- Processo normal, escolha seu diretório de trabalho



Find and run all commands

TERMINAL

SQL CONSOLE: MESSAGES

PROBLEMS

OUTPUT

DEBUG CONSOLE

1: sh

```
Mac-Pro-Angelo:tutorial_git_github angelo$
```



Git & GitHub →
Instalando e
configurando
o GIT

- Aberto seu diretório, terá algo como:
- Observe que no terminal tenho exatamente meu diretório de trabalho
- A partir daí aplique tudo que aprender neste tutorial

Git & GitHub → Instalando e configurando o GIT

- Antes de finalizar, uma última palavra sobre o uso do terminal.
 - Pode acontecer de ao usar um comando **git** ele necessite mais de uma tela do terminal.
 - Quando isso acontecer aparecerá na tela o sinal de dois pontos :
 - Você deverá digitar **<Enter>** para continuar rolando a tela, ou digitar a letra **q** para sair da rolagem de tela e ter o terminal de volta a seu controle
-

TERMINAL

SQL CONSOLE: MESSAGES

PROBLEMS

OUTPUT

DEBUG CONSOLE

```
credential.helper=osxkeychain
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
core.ignorecase=true
core.precomposeunicode=true
remote.origin.url=https://mfake0323@github.com/mfake0323/tutorial_git_gi
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
```

:



Git & GitHub →
Instalando e
configurando o GIT

- Neste ponto, digite **enter** para rolar a tela, ou q para sair da tela.
- Atenção se optar por digitar **enter** chegará a um ponto onde terá **END**: neste momento terá que digitar **q**

TERMINAL

SQL CONSOLE: MESSAGES

PROBLEMS

OUTPUT

DEBUG CONSOLE

```
core.logallrefupdates=true
core.ignorecase=true
core.precomposeunicode=true
remote.origin.url=https://mfake0323@github.com/mfake0323/tutorial_git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.master.remote=origin
branch.master.merge=refs/heads/master
user.name=mfake0323
user.email=mfake0323@gmail.com
```

(END)



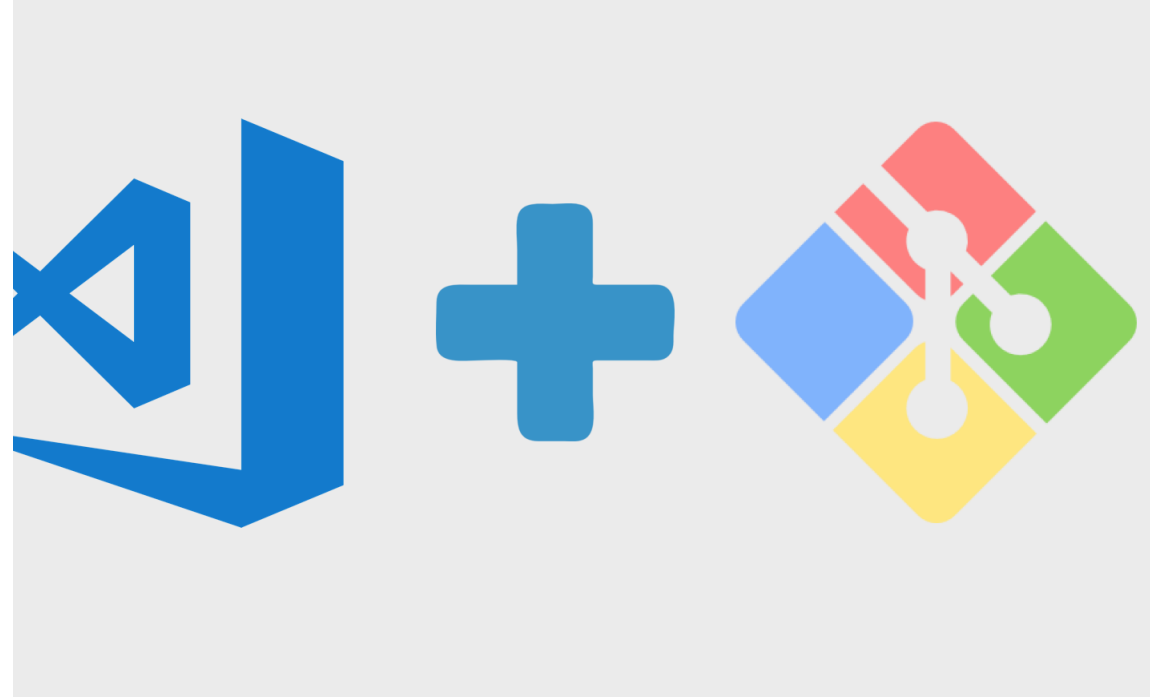
Git & GitHub →
Instalando e
configurando o G

- Neste ponto, necessariamente terá que digitar a letra **q**

GIT INIT & GIT CONFIG

Para iniciarmos o versionamento de um projeto, tudo o que você precisa é do comando → git init

Não importa se a pasta está vazia, ou se arquivos de seu projeto já estão salvos, o git irá habilitar seu repositório para o versionamento



```
DEBUG CONSOLE  PROBLEMS  OUTPUT  TERMINAL

ccol@DESKTOP-TMBP1KD MINGW64 /d/Users/Angelo/GIT
git init
initialized empty Git repository in D:/Users/Angelo/GIT

ccol@DESKTOP-TMBP1KD MINGW64 /d/Users/Angelo/GIT (master)
█
```

GIT INIT & GIT CONFIG

- Iniciando um repositório **git: git init**
- Configurações iniciais:
 - **git config user.name <seu nome de usuário>**
 - **git config user.email <seu email>**



GitHub



Esta Foto de Autor Desconhecido está licenciado em [CC BY-NC](#)

Sucesso a todos!!!

Com o tempo
teremos mais ações
a serem
incorporadas em
nossa rotina

