

🏠 [Início \(https://www.webdevdrops.com\)](https://www.webdevdrops.com) > [Posts \(https://www.webdevdrops.com/categorias/posts/\)](https://www.webdevdrops.com/categorias/posts/)
> [Git no Windows e GitHub: Como Instalar e Configurar](#)

Git no Windows e GitHub: Como Instalar e Configurar

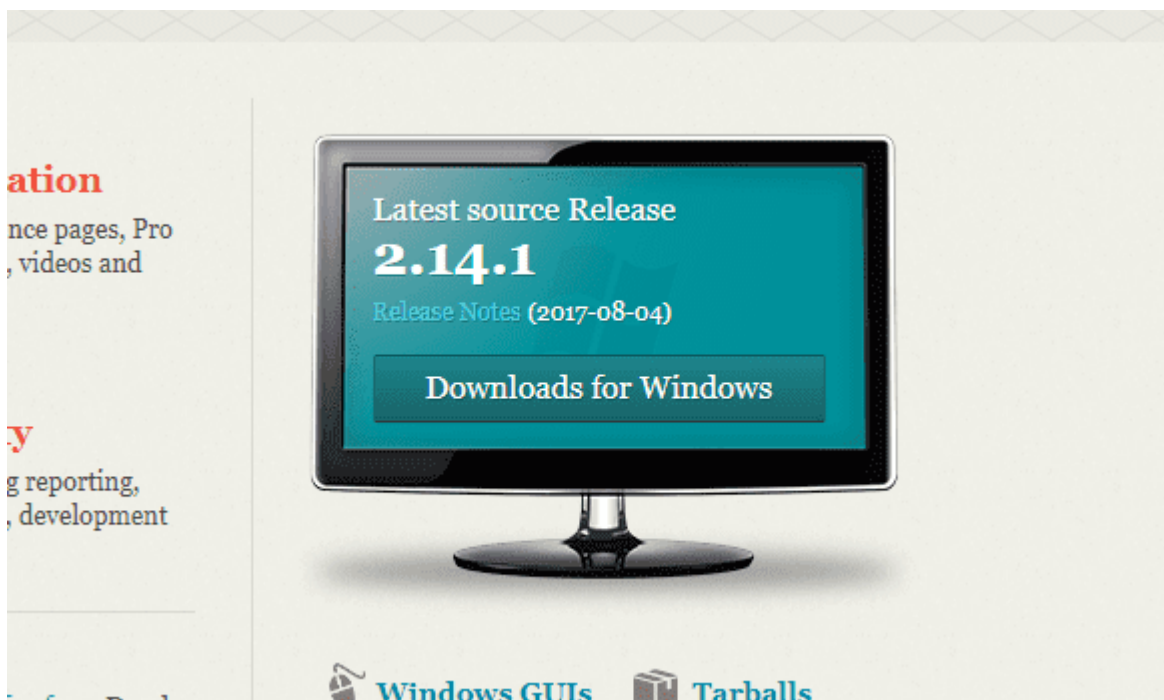


Douglas Matoso 📅 01/01/2017 💬 16 Comentários ⌚ Leitura: 5 min

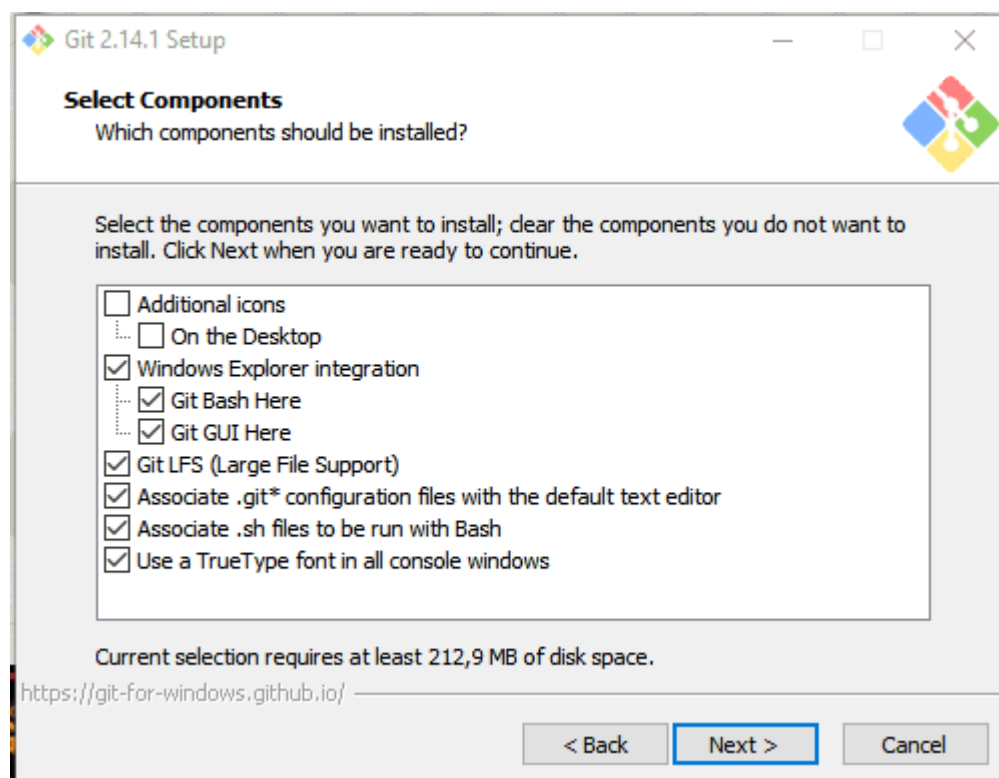
Neste post vou fazer um tutorial simples para utilizar o **Git** (<https://git-scm.com/>) no Windows e comitar seus códigos no **GitHub** (<https://www.github.com/>).

1) Instalando o Git no Windows

No site oficial do **Git** (<http://git-scm.com/> (<https://git-scm.com/>)) clique em “Downloads for Windows”.



Execute o arquivo baixado e vá dando “Next” até a tela “Select Components”. Nesta tela eu escolho as opções como na imagem:



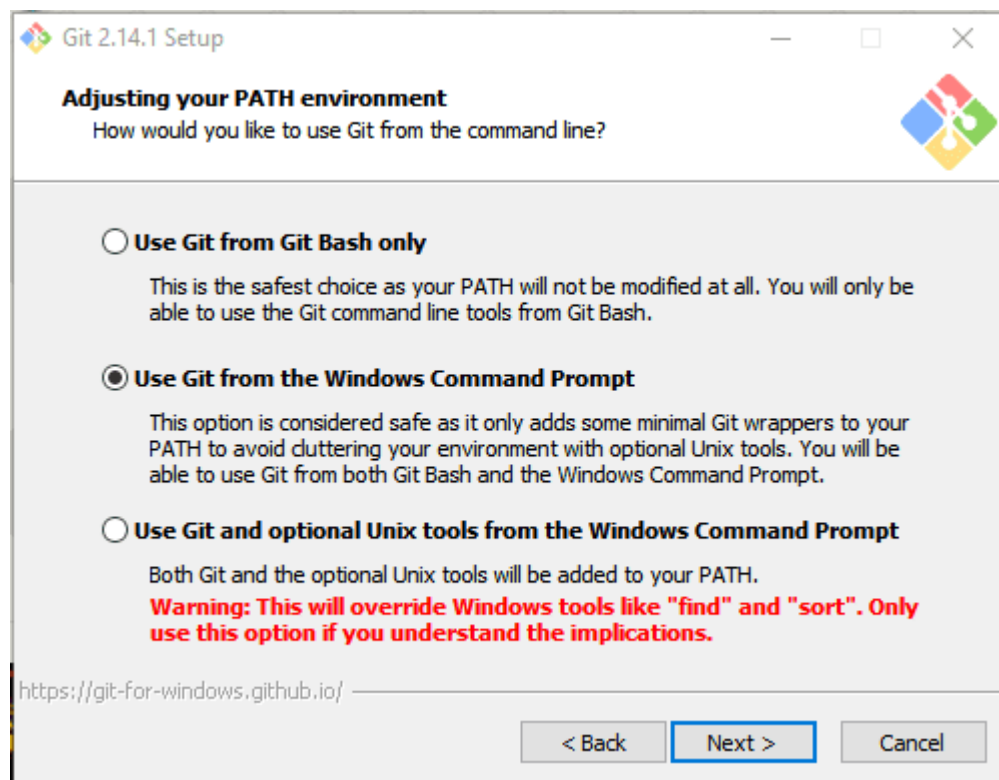
Em especial eu marco as opções em “Windows Explorer integration”, assim eu consigo abrir o prompt de comandos do Git (Git Bash (<https://webdevdrops.com/git-bash-como-instalar-usar/>)) em qualquer pasta,

↑ ()

basta clicar com o botão direito e “**Git Bash Here**”. A última opção também é interessante, porque ele instala uma fonte melhorzinha para o prompt de comandos.

Nota: O Git para Windows vem com um prompt de comandos próprio (o **Git Bash** (<https://webdevdrops.com/git-bash-como-instalar-usar/>)), que além dos comandos git também fornece alguns comandos Unix que podem ser bem úteis (além de ser bem mais bonitinho que o prompt de comandos padrão do Windows).

Na próxima tela, eu escolho a opção: “**Use Git from the Windows Command Prompt**”.

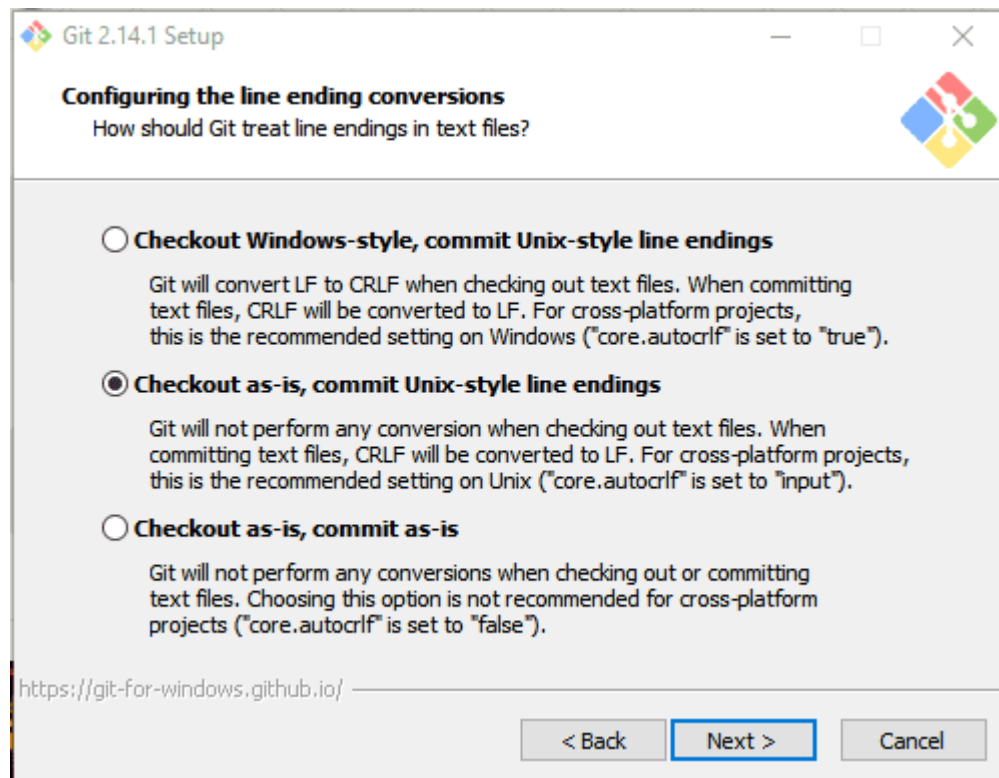


Esta opção permite usar o comando `git` tanto no Git Bash quanto no terminal padrão do Windows.

A terceira adiciona comandos do Unix no terminal do Windows, além do próprio comando `git`, mas alguns comandos do Windows serão substituídos por comandos Unix que tem o mesmo nome (como `find` e `sort`).

Outra configuração importante: **quebra de linhas**.





Como deve saber Windows e sistemas Unix (Linux, Mac) possuem formatos diferentes de quebra de linha em arquivos texto. Se você escreve um código com quebras de linha no formato Windows, outra pessoa pode ter problemas ao abrir o mesmo arquivo em um Linux, e vice-versa. Esta opção permite normalizar isso.

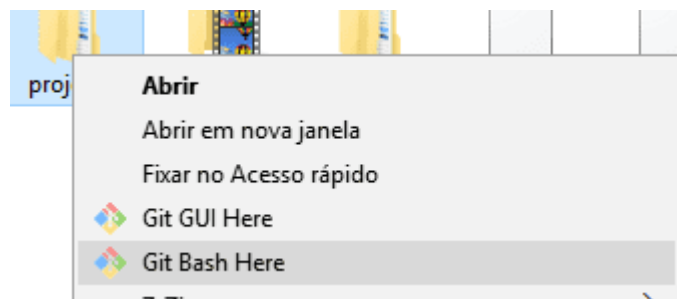
A primeira opção converte os arquivos para padrão Windows quando os arquivos chegam para você, e convertem para padrão Unix quando você os comita no repositório. A segunda não faz nenhuma conversão quando os arquivos chegam, mas convertem para padrão Unix quando você comita. Já a terceira opção não faz nenhuma conversão.

Eu opto pela segunda, pois prefiro manter tudo no padrão Unix (já que qualquer bom editor de código consegue ler arquivos no padrão Unix mesmo estando em Windows).

Feito isso, “Next”, “Finish” e o Git está instalado.

2) Criando um repositório Git local e comitando

Vamos testar? Crie uma pasta, botão direito nela e “Git Bash Here”.



Antes de mais nada, informe ao Git os seus dados, que irão identificar seus commits. Digite os comandos:

```
git config --global user.name "Nome Sobrenome"
git config --global user.email "seu_email@email.com"
```

Dica: para copiar e colar comandos no Git Bash, clique com o botão direito na tela do terminal.

Agora vamos inicializar um repositório Git nesta pasta que estamos.

```
git init
```

```
Douglas@DOUGLAS-PC ~/Desktop/projeto_tutorial
$ git init
Initialized empty Git repository in C:/Users/Douglas/Desktop/projeto_tutorial/.git/
Douglas@DOUGLAS-PC ~/Desktop/projeto_tutorial (master)
$
```

Viu esse (master) que apareceu na linha de comando? Ele indica que você está em um repositório Git, na branch master. Xique, hein?

Vamos adicionar um arquivo neste repositório que está vazio e comitá-lo. Veja a sequência de comandos:

```
touch teste.txt  
git add .  
git commit -m "Primeiro commit"
```

Primeiro criamos um arquivo **teste.txt** vazio. Depois adicionamos todos os novos arquivos (no caso só o teste.txt) ao índice do repositório, e por último comitamos todos os arquivos que estão no índice e foram modificados.

3) Compartilhando no GitHub

Legal, você tem um repositório Git na sua máquina, mas que tal compartilhar seus códigos no **GitHub** e usufruir de tudo que essa comunidade tem para oferecer?

3.1) Preparação Inicial

Vá em <https://github.com/> (<https://github.com/>) e clique em “**Sign Up**” para criar sua conta gratuita.

Tendo cadastrado e logado em sua conta, agora você precisa de uma **chave SSH** para poder começar a comitar. No Git Bash digite:

```
ssh-keygen -t rsa -C "seu_email@email.com"
```

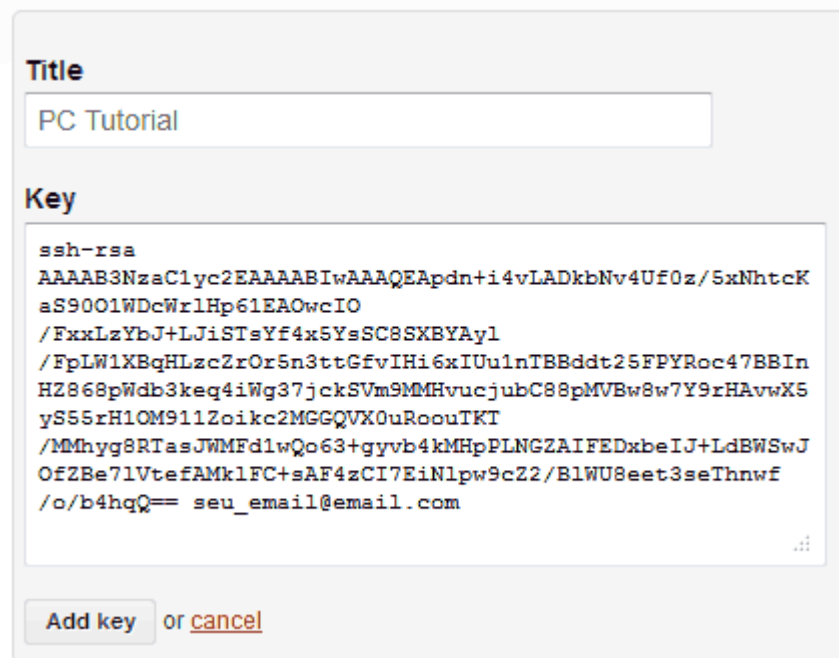
Informe no comando seu e-mail cadastrado no **GitHub**. Dê Enter na próxima pergunta (sobre o arquivo a ser criado—vamos deixar o padrão).

A próxima pergunta vai te pedir uma senha (**passphrase**). Invente uma senha e a informe. Ele vai pedir uma confirmação. Digite a senha novamente e Enter. Digite agora:

```
notepad ~/.ssh/id_rsa.pub
```

para abrir no Bloco de Notas o arquivo que foi criado.

Agora no **GitHub**, vá em “**Settings**” e depois “**SSH and GPG Keys**”. Clique “**New SSH key**”. Informe um título para identificar o computador onde a chave foi gerada e no campo “**Key**” cole todo o conteúdo do arquivo **id_rsa.pub** (que você abriu no Bloco de Notas)



Title

PC Tutorial

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEA...
aS90O1WDcWrlHp61EAOwcIO
/FxxLzYbJ+LJiSTsYf4x5YsSC8SXBYAyl
/FpLW1XBqHLzcZrOr5n3ttGfvIH6xIUuInTBBddt25FPYRoc47BBIn
HZ868pWdb3keq4iWg37jckSVm9MMHvucjubC88pMVBw8w7Y9rHAvwX5
ySS5rH1OM911ZoiKc2MGGQVX0uRoouTKT
/MMhyg8RTasJWMFdlwQo63+gyvb4kMHpPLNGZAIFEDxbeIJ+LdBWSwJ
Of2Be71VtefAMklFC+sAF4zCI7EiNlpw9cZ2/B1WU8eet3seThnwf
/o/b4hqQ== seu_email@email.com
```

Add key or cancel


Tome cuidado de copiar e colar todo o conteúdo do arquivo, começando por “ssh-rsa ...” até o seu e-mail (incluindo ele, como na imagem). Clique em “**Add Key**”.

Vamos testar pra ver se essa bagaça deu certo. No Git Bash digite:

```
ssh -T git@github.com
```

Ele vai perguntar se você tem certeza que quer conectar bla bla bla (yes/no). Digite **yes** e Enter. Na próxima pergunta (Enter passphrase...) informe sua senha (a que você escolheu ao criar a chave SSH).

Se você receber uma mensagem do tipo:

A screenshot of a terminal window with a light gray background. It displays the message "Hi doug2k1! You've successfully authenticated, but GitHub does not provide shell access." in a monospaced font. Below the text is a horizontal scrollbar with a gray track and a white slider.

```
Hi doug2k1! You've successfully authenticated, but GitHub does not provide shell access.
```

Então tudo deu certo!


3.2) Criando o primeiro repositório remoto

No **GitHub** vamos criar um novo repositório (botão “**New Repository**” no seu dashboard). Informe um nome sem espaços e caracteres especiais. As outras opções não precisa alterar.


Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 doug2k1

Repository name

projeto-tutorial 

Description (optional)

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None**

Add a license: **None** 

Create repository

Você cairá na página do seu repositório, que por enquanto ainda não tem arquivos.

Importante! Se o e-mail informado ao Git no início do **passo 2** não for o mesmo usado para se cadastrar no **GitHub**, refaça o comando informando o e-mail cadastrado. Assim o **GitHub** vai poder ligar os commits à sua conta.

No Git Bash (na pasta do seu repositório local) digite:

```
git remote add origin git@github.com:login/repositorio.git
```

Note que **login/repositorio** deve ser digitado como aparece na URL do seu repositório, no exemplo:

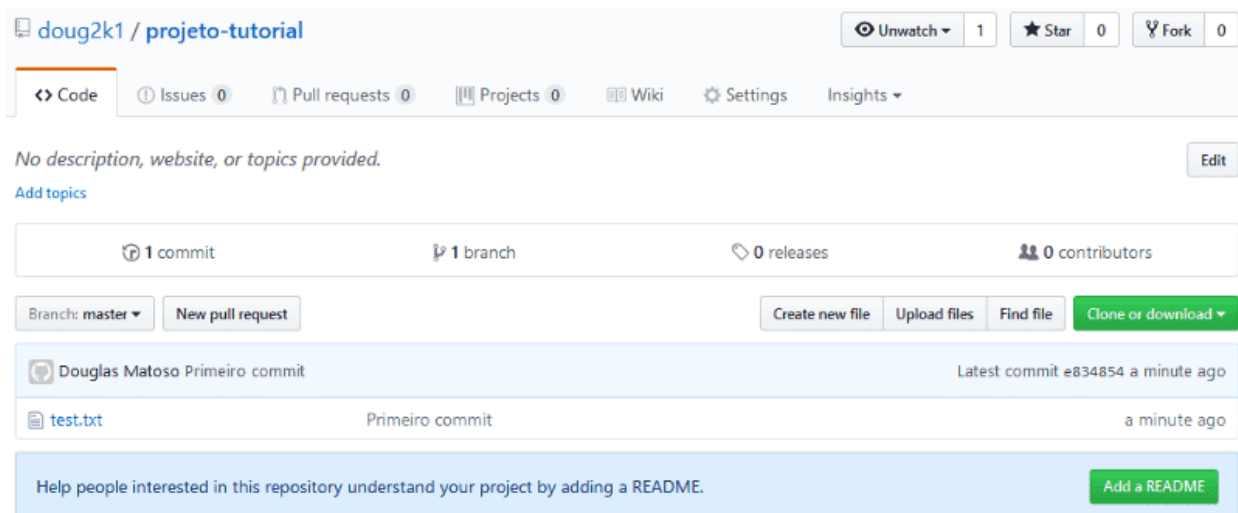
<https://github.com/doug2k1/projeto-tutorial> (<https://github.com/doug2k1/projeto-tutorial>)

Agora para atualizar o **GitHub** com o que está no seu repositório local, digite:

```
git push -u origin master
```

Informe a sua senha (da chave SSH) quando pedido.

Recarregue a página do seu repositório e agora, ao invés da mensagem inicial, você verá seus commits e arquivos.



4) Conclusão

Apesar do Git ter se originado no Linux (Sabia que foi o Linus Torvalds que o criou?), a turma que usa Windows (seja porque gosta ou porque precisa) também pode se beneficiar com ele, graças ao Git for Windows. Além do próprio Git ser um excelente sistema de controle de versão, a comunidade open-source que popula o GitHub é vibrante. Tão gratificante quanto achar lá um código que “salva a sua pele” é poder contribuir com algum projeto, fazer *forks*, compartilhar.

Até a próxima!



Quer aprender a usar Git e as principais ferramentas e linguagens para **desenvolver sites e sistemas web**? Conheça o **Pacote Fullstack** (<https://go.hotmart.com/H11987895V?ap=7cc1&src=postgitwindows>), da Danki Code, o mais completo treinamento para desenvolvimento web no Brasil.

- Git—Site Oficial (<https://git-scm.com/>)
- GitHub (<https://github.com/>)
- Git no Wikipédia (<https://pt.wikipedia.org/wiki/Git>)
- Micro Tutorial de Git—AkitaOnRails (<http://akitaonrails.com/2008/04/02/micro-tutorial-de-git>)
- Minha página no GitHub (<https://github.com/doug2k1>)

11/18

(<https://www.webdevdrops.com/node-js-typescript/>)

Continue Lendo ➡

(<https://www.webdevdrops.com/node-js-typescript/>)

node-react-graphql-

postgresql-

sequelize-11e646979b27/)

Fullstack com Node.js, React e GraphQL – 3: PostgreSQL e Sequelize (<https://www.webdevdrops.com/fullstack-node-react-graphql-postgresql-sequelize-11e646979b27/>)

Continue Lendo ➡

(<https://www.webdevdrops.com/fullstack-node-react-graphql-postgresql-sequelize-11e646979b27/>)

pacote-full-stack-danki-code/)

Aulas Grátis do Curso Pacote Full-Stack da Danki Code (<https://www.webdevdrops.com/aulas-gratis-curso-pacote-full-stack-danki-code/>)

Continue Lendo ➡

(<https://www.webdevdrops.com/aulas-gratis-curso-pacote-full-stack-danki-code/>)

gerador-site-estatico-40-linhas-nodejs-f1902dd41adb/)

Construindo um gerador de site estático em 40 linhas com Node.js (<https://www.webdevdrops.com/construindo-gerador-site-estatico-40-linhas-nodejs-f1902dd41adb/>)

Continue Lendo ➡

(<https://www.webdevdrops.com/construindo-gerador-site-estatico-40-linhas-nodejs-f1902dd41adb/>)

Sobre o Autor

Douglas Matoso

(<https://www.webdevdrops.com/author/dmatoso/>)

Desenvolvedor web desde 2008. Criador do Web Dev Drops.

16 Comentários

Deixe um comentário

↑ ()

O seu endereço de e-mail não será publicado.

Campos obrigatórios são marcados com *

Comentário

Nome *

E-mail *

Site

☐

Salvar meus dados neste navegador para a próxima vez que eu comentar.

PUBLICAR COMENTÁRIO

Ricardo disse:

15/09/2020 às 11:46 (<https://www.webdevdrops.com/git-no-windows-github/#comment-297>)

Cara muito obrigado vc mandou muito bem direto no ponto mas no passo-a-passo.



Responder

Douglas Matoso disse:

15/09/2020 às 21:42 (<https://www.webdevdrops.com/git-no-windows-github/#comment-298>)

Obrigado pelo feedback, Ricardo! Tamo junto! 🙌

Responder

Bárbara Xavier disse:

31/08/2020 às 22:41 (<https://www.webdevdrops.com/git-no-windows-github/#comment-281>)

Obrigada por compartilhar o conhecimento. Ajudou bastante 😊

Responder

Douglas Matoso disse:

31/08/2020 às 22:58 (<https://www.webdevdrops.com/git-no-windows-github/#comment-282>)

Oi, Bárbara! Que bom saber que ajudou! Obrigado pelo feedback!

Responder

alberto oliveira disse:

04/08/2020 às 19:27 (<https://www.webdevdrops.com/git-no-windows-github/#comment-236>)

excelente post! bastante esclarecedor

Responder

Douglas Matoso disse:

↑ ()

04/08/2020 às 21:34 (<https://www.webdevdrops.com/git-no-windows-github/#comment-237>)

Bom saber que foi útil, Alberto. Obrigado pelo feedback!

Responder

Gilson disse:

02/08/2020 às 17:03 (<https://www.webdevdrops.com/git-no-windows-github/#comment-232>)

Opa Muito Obrigado fui salvo.

Responder

Douglas Matoso disse:

02/08/2020 às 17:05 (<https://www.webdevdrops.com/git-no-windows-github/#comment-233>)

Opa! Bom saber que ajudou, Gilson! Obrigado pelo feedback!

Responder

George disse:

11/07/2020 às 14:09 (<https://www.webdevdrops.com/git-no-windows-github/#comment-217>)

ME SALVOU! MARAVILHA!

Responder

Douglas Matoso disse:

11/07/2020 às 22:39 (<https://www.webdevdrops.com/git-no-windows-github/#comment-218>)

Opa! Que bom que foi útil, George! Obrigado pelo feedback. 🙌

Responder

EMERSON FAUSTO LISBOA disse:

06/07/2020 às 22:53 (<https://www.webdevdrops.com/git-no-windows-github/#comment-201>)
Parabéns pelo post.. Depois de várias tentativas, li e segui o seu post e deu certo.. Muito obrigado!

Responder

Douglas Matoso disse:

07/07/2020 às 21:56 (<https://www.webdevdrops.com/git-no-windows-github/#comment-209>)

Que bom que foi útil, Emerson! Obrigado pelo feedback! 🙌

Responder

Elen Araujo disse:

12/04/2020 às 02:03 (<https://www.webdevdrops.com/git-no-windows-github/#comment-98>)
Muito obrigada mesmo pelo tutorial! Git é bem completo (e complexo), então pode parecer bastante assustador pra quem ta começando. Achei tua forma de ensinar bem lúdica 😂 Arrasou demais!

Responder

Douglas Matoso disse:

12/04/2020 às 14:34 (<https://www.webdevdrops.com/git-no-windows-github/#comment-100>)

Obrigado pelo feedback, Elen! Quem bom que gostou! 😊

Responder

maisa disse:

↑ ()

03/04/2020 às 16:26 (<https://www.webdevdrops.com/git-no-windows-github/#comment-86>)

Melhor artigo, o mais auto explicativo e claro (sempre que entro em um pc novo eu tenho que ficar caçando tutorial pra configurar o github pq nunca lembro de cabeça).

Melhor parte do artigo "Ele vai perguntar se você tem certeza que quer conectar bla bla bla (yes/no). "

hahaha um trecho simples e descontraído que de alguma forma me fez sentir conectada. De fato, o que eu quero é só conectar o git e nao entender o que cada item faz no mais profundo do seu core.

Responder

Douglas Matoso disse:

03/04/2020 às 16:33 (<https://www.webdevdrops.com/git-no-windows-github/#comment-87>)

Obrigado pelo feedback, Maisa! Que bom que gostou do conteúdo e da didática. 😊

Responder



**DESENVOLVIMENTO WEB FULL STACK –
PROGRAMADOR BR**

**([HTTPS://GO.HOTMART.COM/X41967180X?
AP=2482&SRC=SIDEWDD](https://go.hotmart.com/X41967180X?AP=2482&SRC=SIDEWDD))**




**FRONTEND DO ZERO AO
PROFISSIONAL**

**([HTTPS://GO.HOTMART.COM/R16612227G?
SRC=SIDEWDD](https://go.hotmart.com/R16612227G?SRC=SIDEWDD))**

\$100 EM CRÉDITOS NA DIGITAL OCEAN([HTTPS://M.DO.CO/C/C31F1ECAB4AA](https://m.do.co/C/C31F1ECAB4AA))



 Português (<https://www.webdevdrops.com/git-no-windows-github/>)

 English (<https://www.webdevdrops.com/en/git-on-windows-github-install-configure/>)

Contato

(<https://www.webdevdrops.com/contact/>)

Política de Privacidade

(<https://www.webdevdrops.com/privacidade/>)

This site is protected by

reCAPTCHA and the


Google Privacy Policy

(<https://policies.google.com/privacy>)

and Terms of Service

(<https://policies.google.com/terms>)

apply.

© Web Dev Drops. Feito com  por Douglas Matoso (<https://github.com/doug2k1>)