

Práticas de Engenharia de Software

Guia de uso do git/github

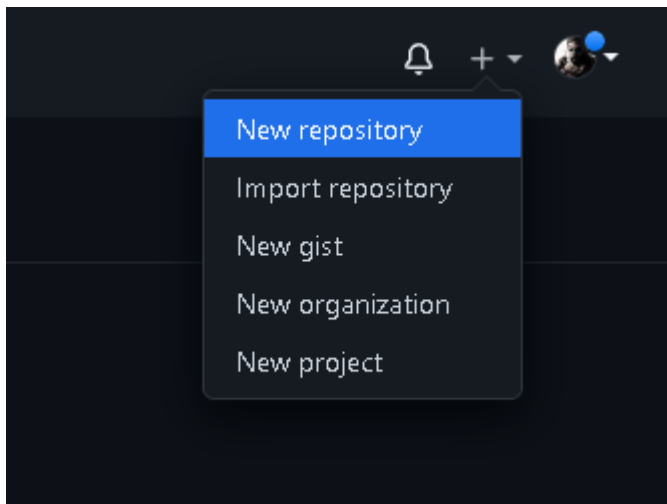
Centro Universitário Ritter dos Reis (UniRitter)

Porto Alegre – RS – Brazil

Ezequiel Bitencourt

ezequiel.bitencourt197@gmail.com

Para criar um repositório no gitHub basta ter uma conta criada, assim que acessa-lá, vá para o canto superior direito e clique no sinal de mais (+), logo após isso irá abrir uma caixa com opções, nela você clica na primeira opção, “New Repository”.




Assim irá aparecer uma janela de criação de repositório onde você deve inserir um nome para o repositório, definir se será um repositório público ou privado, e se quiser, adicionar uma descrição para ele.

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 *

 EzequielBitencourt ▾


/


guia_git_github_pes ✓

Great repository names are short and memorable. Need inspiration? How about **musical-disco**?

Description (optional)

Um guia de como criar seu repositório e utilizar os comandos do Git para subir alterações.

☒  **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.

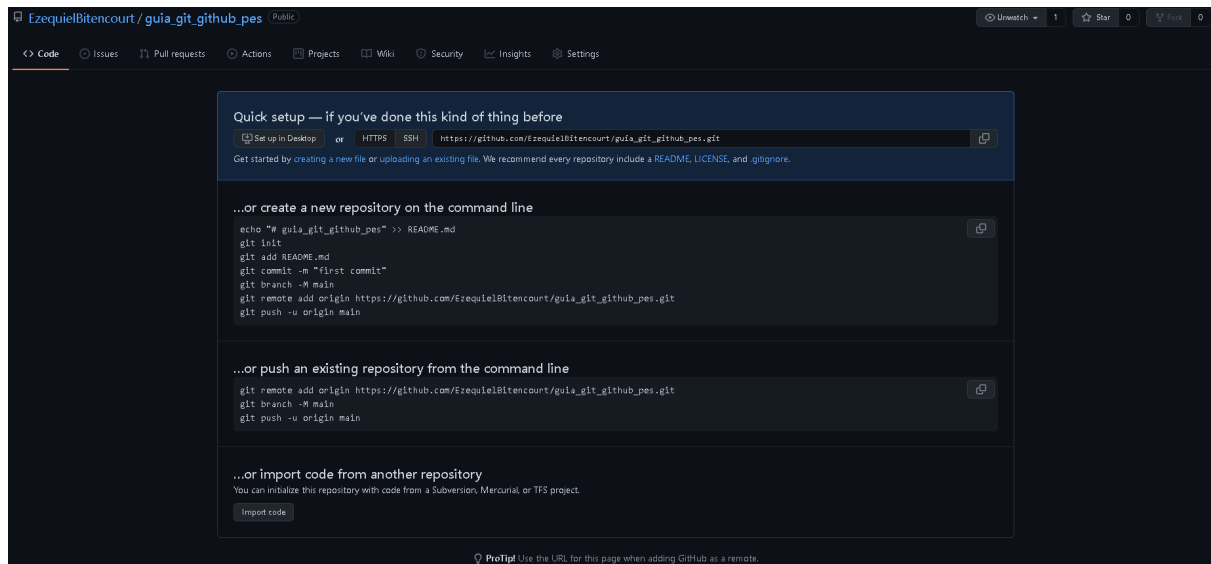
☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

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

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

Create repository

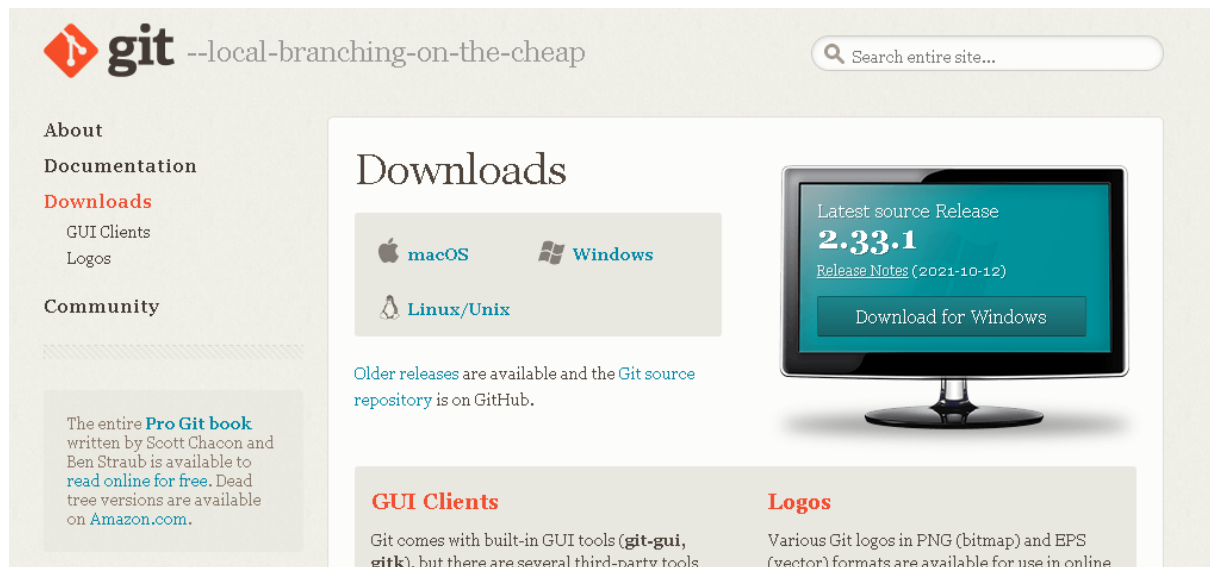
Assim que clicar no Create Repository, você vai ver a seguinte tela:



Nela podemos ver um link que será usado para clonar o repositório para nossa maquina e alguns comandos explicando como fazer e, se quiser, pode simplesmente importar seu projeto clicando em import code.

Neste guia eu irei utilizar o GitBash, que é um terminal, como o prompt de comando, só que exclusivo para comandos do Git.

Você pode baixar ele por aqui :
[Git - Downloads \(git-scm.com\)](https://git-scm.com/downloads)

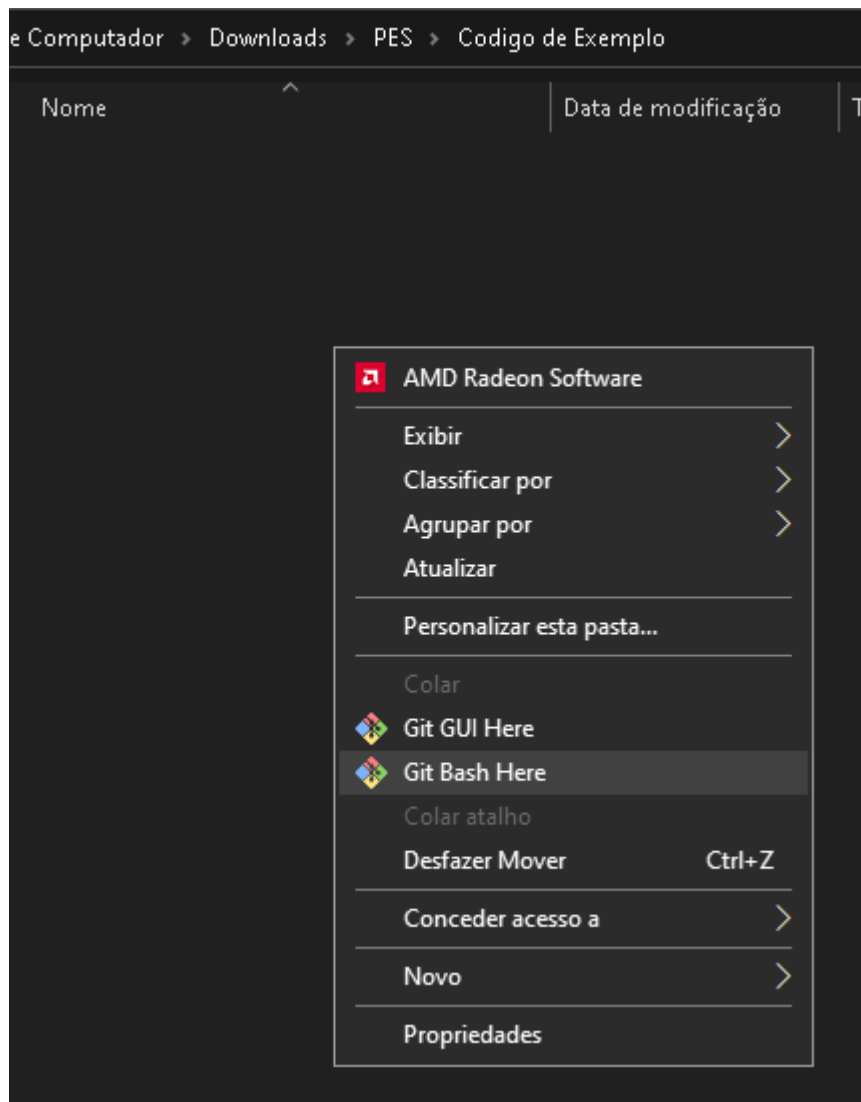


Para instalar é bem simples, baixe ele para o seu tipo de dispositivo e selecione as opções recomendadas.

Criei um código simples em C# que será utilizado de exemplo.

```
1  using System;
2
3  namespace HelloWorld
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              Console.WriteLine("Este é meu Guia pessoal de Git/GitHub");
10         }
11     }
12 }
```

Em uma pasta vazia que reservei para clonar nosso repositório, vou clicar com o botão direito e ir em GitBash Here, onde irá abrir um terminal.

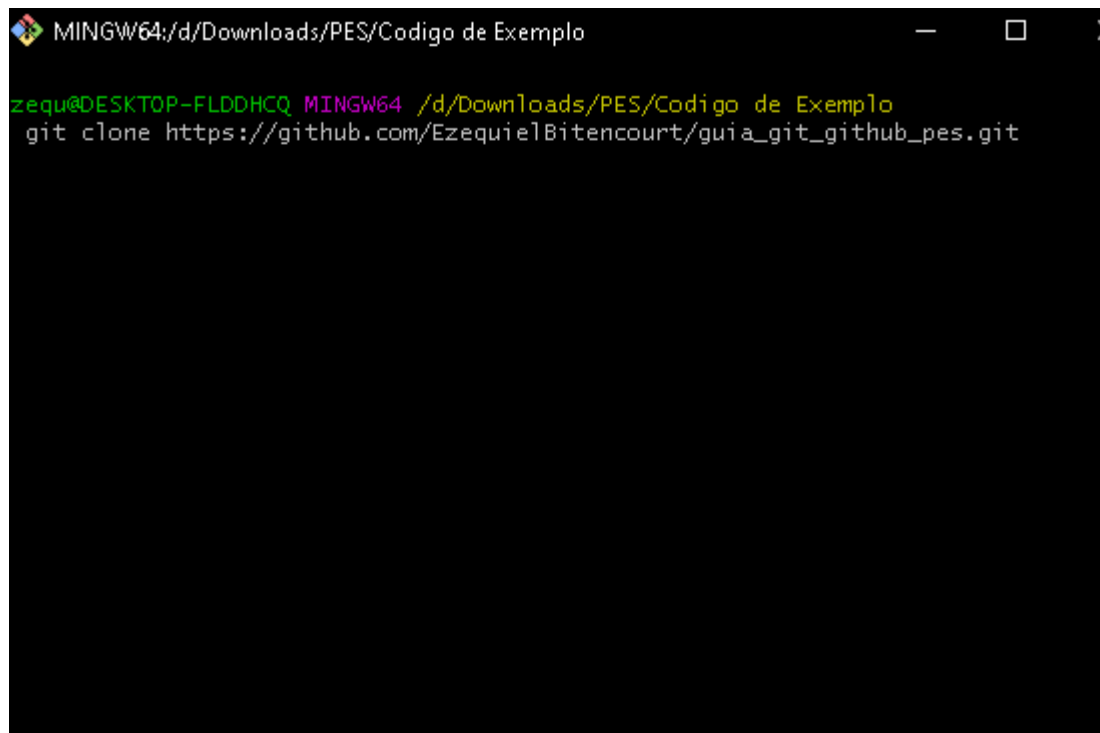


E neste terminal vou digitar o seguinte comando :

git clone https://github.com/EzequielBitencourt/guia_git_github_pes.git

A função do git clone é, como o próprio nome diz, clonar. Ele irá clonar o repositório do link que você inserir. Neste caso o link é o fornecido na página que aparece logo após criar o repositório:

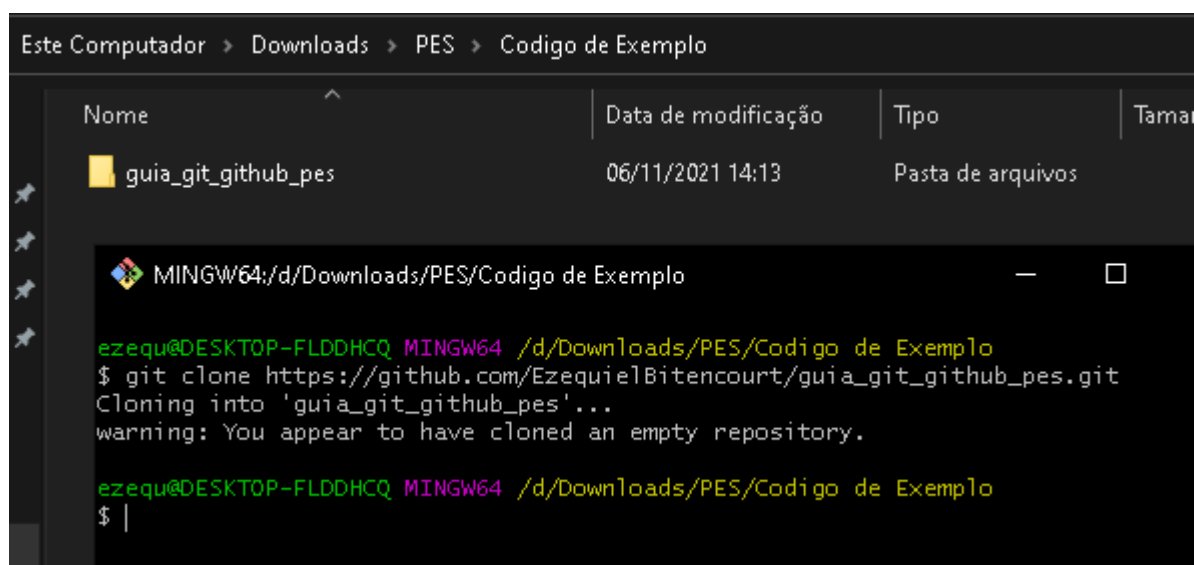




```
MINGW64:/d/Downloads/PES/Codigo de Exemplo

zequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo
git clone https://github.com/EzequielBitencourt/guia_git_github_pes.git
```

Dando um Enter o comando irá rodar e será criado uma pasta com o mesmo nome do nosso repositório



Este Computador > Downloads > PES > Codigo de Exemplo

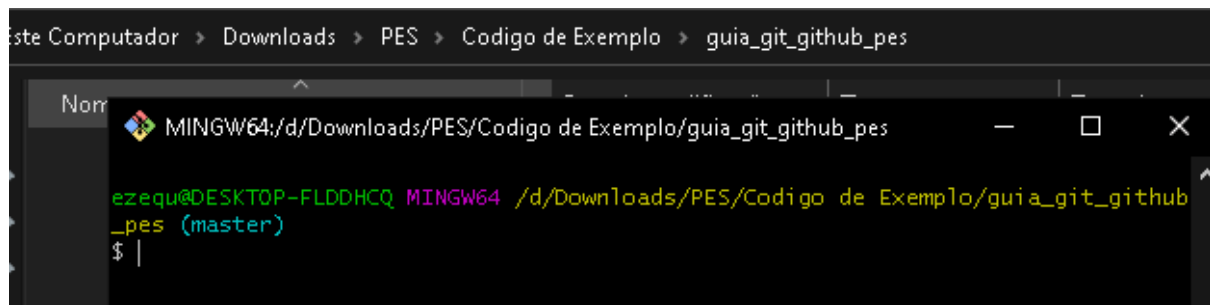
Nome	Data de modificação	Tipo	Tamanho
guia_git_github_pes	06/11/2021 14:13	Pasta de arquivos	

```
MINGW64:/d/Downloads/PES/Codigo de Exemplo

zequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo
$ git clone https://github.com/EzequielBitencourt/guia_git_github_pes.git
Cloning into 'guia_git_github_pes'...
warning: You appear to have cloned an empty repository.

zequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo
$ |
```

Agora vamos entrar nesta pasta e abrir outro terminal com o GitBash Here:



```
Este Computador > Downloads > PES > Codigo de Exemplo > guia_git_github_pes
```

```
Nome
```

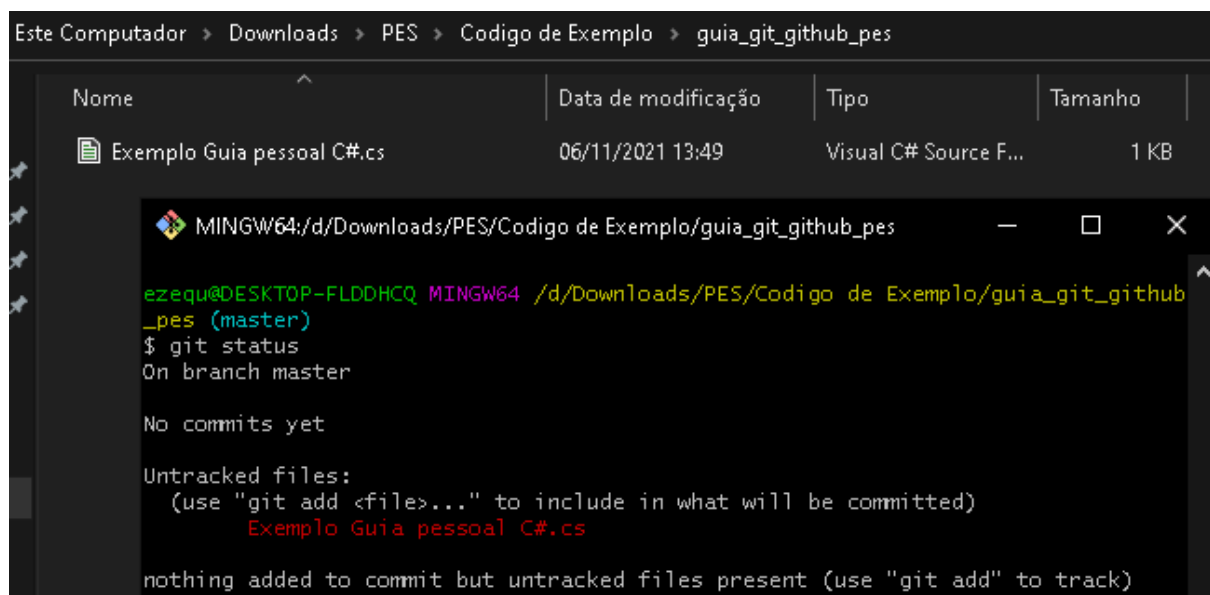
```
MINGW64:/d/Downloads/PES/Codigo de Exemplo/guia_git_github_pes
```

```
ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo/guia_git_github_pes (master)
```

```
$ |
```

Note que agora temos um (master) em azul. Isto significa que esta pasta está conectada ao nosso repositório na “Branch” master. A “Branch” é uma ramificação, logo que você cria um repositório será criado uma branch chamada main ou master, este é o ramo principal do seu repositório onde terá o primeiro código ou como no nosso caso nada.

Agora e u adicionei meu código dentro dessa pasta e rodei um git status, este comando serve para ver o que temos de alterações feitas nessa branch.



```
Este Computador > Downloads > PES > Codigo de Exemplo > guia_git_github_pes
```

Nome	Data de modificação	Tipo	Tamanho
Exemplo Guia pessoal C#.cs	06/11/2021 13:49	Visual C# Source F...	1 KB

```
MINGW64:/d/Downloads/PES/Codigo de Exemplo/guia_git_github_pes
```

```
ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo/guia_git_github_pes (master)
```

```
$ git status
```

```
On branch master
```

```
No commits yet
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
Exemplo Guia pessoal C#.cs
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

Podemos ver que ele detectou que o nosso código foi adicionado na nossa pasta mas ainda não foi comentado e nem adicionado.

Para mandarmos alterações para o nosso repositório no GitHub nós devemos seguir um padrão, primeira mente dando um Git Add, o comando git add é para adicionar arquivos que você quer commitar ou seja mandar para o repositório, o git add pode ser usado com um ponto (.) ou um caminho do arquivo(...\guia_git_github_pes\Exemplo Guia pessoal C#.cs), utilizando o ponto ele irá adicionar todas as alterações que você fez, muito cuidado com

isso pois pode adicionar algo que não queira. Já passando o caminho do arquivo você vai adicionar somente este arquivo.

Nosso comando será git add .

```
ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de
_pes (master)
$ git add .

ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de
_pes (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Exemplo Guia pessoal C#.cs
```

rodando um git status podemos ver que o nome ficou em vermelho, significando que este arquivo está adicionado.

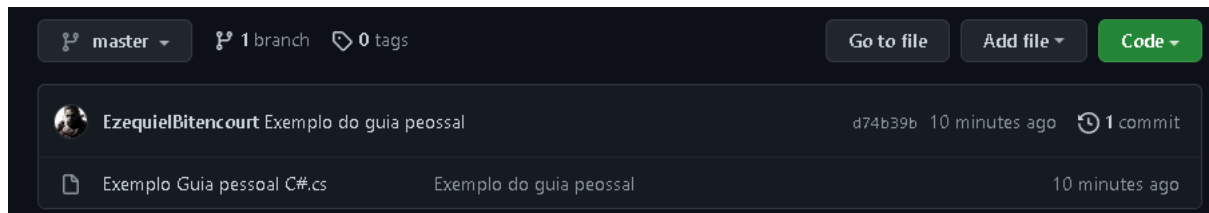
Agora vamos rodar o comando chamado git commit, ele irá pegar essas alterações e salvar elas localmente.

```
ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo/guia_git_github
_pes (master)
$ git commit -m"Exemplo do guia peossal"
[master (root-commit) d74b39b] Exemplo do guia peossal
 1 file changed, 12 insertions(+)
 create mode 100644 Exemplo Guia pessoal C#.cs
```

tendo um commit podemos rodar o git push, ele serve para subir todos os commits para o repositório, lembrando que ele pode pedir para você logar no gitHub para conseguir dar um push, mas é bem simples o procedimento.

```
ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo/guia_git_github
_pes (master)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 403 bytes | 403.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/EzequielBitencourt/guia_git_github_pes.git
 * [new branch]      master -> master
```

Agora podemos ver no GitHub nossas alterações.



Mas digamos que eu quero fazer alguma alteração no código. Utilizando a mecânica de ramificação podemos separar trabalhos, como por exemplo, eu quero fazer algumas implementações no código e meu colega que fazer alguns testes, então o correto é eu criar uma outra branch para alterar o código e meu colega criar outra para seus testes.

Vou rodar o seguinte comando:

```
git checkout -b nova_implementacao
```

```
ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo/guia_git_github
_pes (master)
$ git checkout -b nova_implementacao
Switched to a new branch 'nova_implementacao'

ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo/guia_git_github
_pes (nova_implementacao)
$ |
```

O git checkout serve para você se mover de branch em branch, já o -b é para criar uma branch nova e já selecionar ela, podemos ver que já estou com a branch nova_implementacao selecionada, caso eu queria voltar para a master é só dar um git checkout master.

Agora nessa branch vou fazer adicionar um arquivo de texto e uma alteração no código, rodando um git status podemos ver o seguinte:

```
ste Computador > Downloads > PES >Codigo de Exemplo > guia_git_github_pes
```

Nome	Data de modificação	Tipo	Tamanho
Exemplo Guia pessoal C#.cs	06/11/2021 15:15	Visual C# Source F...	1 KB
README	06/11/2021 15:13	Documento de Te...	1 KB

```
MINGW64:/d/Downloads/PES/Codigo de Exemplo/guia_git_github_pes

ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo/guia_git_github_pes (nova_implementacao)
$ git status
On branch nova_implementacao
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Exemplo Guia pessoal C#.cs

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

Temos um arquivo que já existe no repositório que foi modificado e temos um novo arquivo, agora seguimos o mesmo padrão feito anteriormente:

git add .

git commit -m"Adicionando README e implementando o codigo"

git push

```
ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo/guia_git_github_pes (nova_implementacao)
$ git add .

ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo/guia_git_github_pes (nova_implementacao)
$ git commit -m"Adicionando README e implementando o codigo"
[nova_implementacao 8a332bf] Adicionando README e implementando o codigo
2 files changed, 2 insertions(+)
create mode 100644 README.txt

ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo/guia_git_github_pes (nova_implementacao)
$ git push
fatal: The current branch nova_implementacao has no upstream branch.
To push the current branch and set the remote as upstream, use

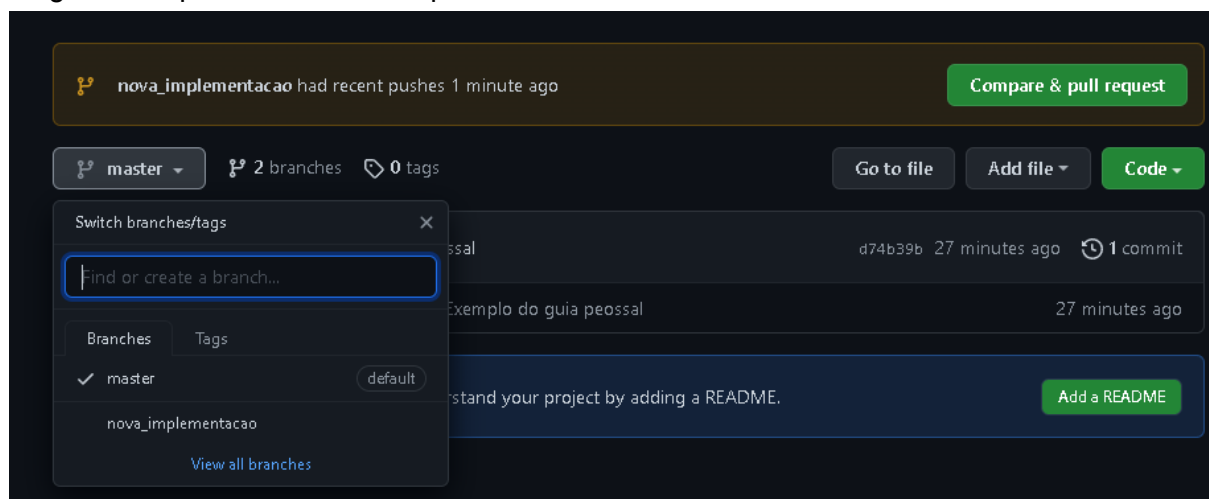
    git push --set-upstream origin nova_implementacao
```

Podemos ver que houve um erro aqui. O problema é que não temos um caminho exato para essa branch que criamos, ela não existe ainda no repositório, só localmente, e só com esse push ela vai aparecer no GitHub.

Agora utilizando o comando que o terminal no disse podemos dar o push :
`git push --set-upstream origin nova_implementacao`

```
ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo/guia_git_github_pes (nova_implementacao)
$ git push --set-upstream origin nova_implementacao
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 468 bytes | 468.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'nova_implementacao' on GitHub by visiting:
remote:   https://github.com/EzequielBitencourt/guia_git_github_pes/pull/new/nova_implementacao
remote:
To https://github.com/EzequielBitencourt/guia_git_github_pes.git
 * [new branch]      nova_implementacao -> nova_implementacao
Branch 'nova_implementacao' set up to track remote branch 'nova_implementacao' from 'origin'.
```

E agora ela aparece no nosso repositório:



Para mandarmos essa alteração para a master, podemos fazer pelo próprio GitHub clicando em Compare & Pull request ou fazer pelo GitBash. Neste caso vamos pelo GitBash:

Primeiramente temos que voltar para nossa master

```
ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo/guia_git_github_pes (nova_implementacao)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo/guia_git_github_pes (master)
$ |
```

E agora rodar o seguinte comando:

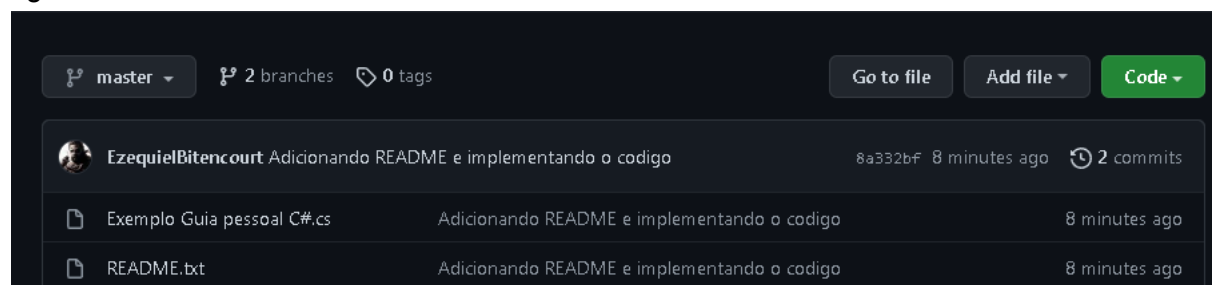
`git merge nova_implementacao`

```
ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downlo
_pes (master)
$ git merge nova_implementacao
Updating d74b39b..8a332bf
Fast-forward
 Exemplo Guia pessoal C#.cs | 1 +
 README.txt                 | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 README.txt
```

Por ultimo um git push

```
ezequ@DESKTOP-FLDDHCQ MINGW64 /d/Downloads/PES/Codigo de Exemplo/
_pes (master)
$ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/EzequielBitencourt/guia_git_github_pes.git
 d74b39b..8a332bf master -> master
```

Agora nossa master está atualizada com a outra branch.



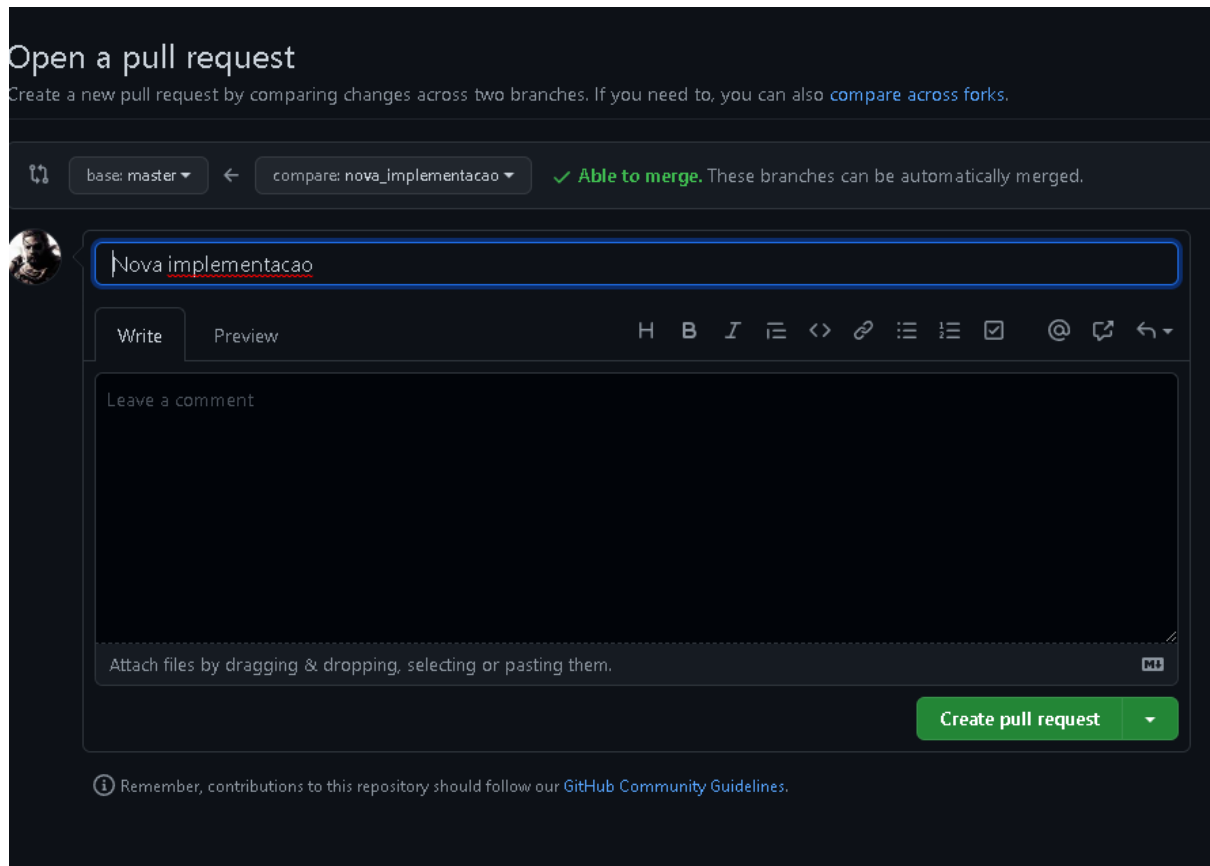
The screenshot shows the GitHub interface for a repository named 'guia_git_github_pes'. At the top, it indicates the current branch is 'master', with 2 branches and 0 tags. Below this, a commit by 'EzequielBitencourt' is displayed, titled 'Adicionando README e implementando o codigo', with commit hash '8a332bf' and a timestamp of '8 minutes ago'. The commit message is 'Adicionando README e implementando o codigo'. Below the commit details, a table lists the files changed in this commit:

File	Commit Message	Time
Exemplo Guia pessoal C#.cs	Adicionando README e implementando o codigo	8 minutes ago
README.txt	Adicionando README e implementando o codigo	8 minutes ago

Dependendo do ambiente de trabalho, talvez você deve solicitar o pull request e fazer o procedimento pelo próprio GitHub, pois terá algum administrador que irá avaliar seu código e aprovar/reprovar ele. O método que fizemos é o mais rápido e altera automaticamente a branch.

Para demonstrar o pull request fiz uma rápida alteração de código na branch nova_implementacao e dei push para a mesma.



Clicando em Compare & Pull request sou direcionado para uma janela onde irá mostrar de qual branch é a origem da alteração e para onde ela vai dar merge










The screenshot shows the GitHub 'Open a pull request' page. At the top, it says 'Open a pull request' and 'Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).' Below this, there are two dropdown menus: 'base: master' and 'compare: nova_implementacao'. To the right of these is a green checkmark and the text 'Able to merge. These branches can be automatically merged.' Below the dropdowns is a text input field containing 'Nova implementacao'. To the left of the input field is a small circular profile picture. Below the input field are two tabs: 'Write' and 'Preview'. To the right of the tabs is a rich text editor toolbar with icons for bold, italic, link, code, list, and other formatting options. Below the toolbar is a large text area with the placeholder text 'Leave a comment'. At the bottom of the text area is a small icon for attaching files. Below the text area is a green button labeled 'Create pull request' with a dropdown arrow. At the very bottom, there is a small information icon and the text 'Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).'

Ao clicar em Create pull Request vou para a janela que realiza tal ação.



Nova implementacao #1

 Open EzequielBitencourt wants to merge 1 commit into `master` from `nova_implementacao` 


 Conversation **0**  Commits **1**  Checks **0**  Files changed **1**


 **EzequielBitencourt** commented now Owner  


No description provided.


  Nova implementacao 6292bca














Add more commits by pushing to the `nova_implementacao` branch on [EzequielBitencourt/guia_git_github_pes](#).




 Continuous integration has not been set up
[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.



 **This branch has no conflicts with the base branch**
Merging can be performed automatically.


Merge pull request  You can also [open this in GitHub Desktop](#) or [view command line instructions](#).

 **Write** **Preview**            

Leave a comment

Attach files by dragging & dropping, selecting or pasting them. 

 Close pull request  Comment

 Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

O próprio GitHub verifica se é possível dar merge automaticamente ou tem algum conflito no código e após isso já nos habilita a dar merge pull request. Lembrando que neste caso somos os donos do repositório então eu mesmo posso aceitar o merge, mas dependendo do ambiente, temos que esperar outra pessoa aceitar.

Confirmando o merge, ele já é realizado automaticamente.

The screenshot shows a GitHub pull request interface. At the top, it says 'Nova implementacao #1' and 'Merged' with a status bar indicating 'EzequielBitencourt merged 1 commit into master from nova_implementacao now'. Below this, there are tabs for 'Conversation' (0), 'Commits' (1), 'Checks' (0), and 'Files changed' (1). A comment from 'EzequielBitencourt' is shown, stating 'No description provided.' Below the comment, a commit 'Nova implementacao' is listed with hash '6292bca'. A merge action is shown: 'EzequielBitencourt merged commit edb6726 into master now', with a 'Revert' button. A green banner states 'Pull request successfully merged and closed' with a 'Delete branch' button. At the bottom, there is a 'Write' tab and a 'Preview' tab, with a text area for 'Leave a comment' and a 'Comment' button.

E assim já atualizando a nossa master

The screenshot shows the GitHub repository file list. At the top, it says 'master' (selected), '2 branches', and '0 tags'. There are buttons for 'Go to file', 'Add file', and 'Code'. Below this, a table lists the files in the repository:

File	Commit	Time
Exemplo Guia pessoal C#.cs	Nova implementacao	6 minutes ago
README.txt	Adicionando README e implementando o codigo	27 minutes ago

Ao realizar um push ou um pull request nós podemos também fazer certas configurações, para ver se o código comentado está rodando sem erros, se os testes passaram ou até mesmo dar um deploy em algum servidor. Essas ações são chamadas de Pipelines e podem ser configuradas no GitHub ou outra plataforma.

Link para o repositório:

[EzequielBitencourt/guia_git_github_pes: Um guia de como criar seu repositório e utilizar os comandos do Git para subir alterações.](#)