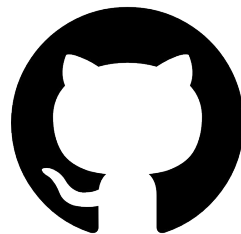
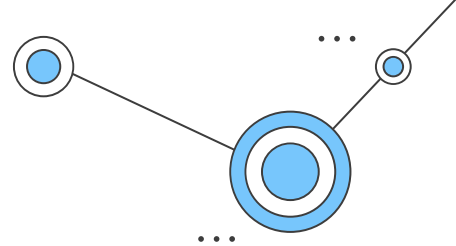


Git GitHub



Equipe Script



Paulo
Deyvid



Jonathan
Silva



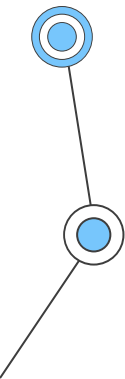
Luan
Machado



Hewerton
Brito

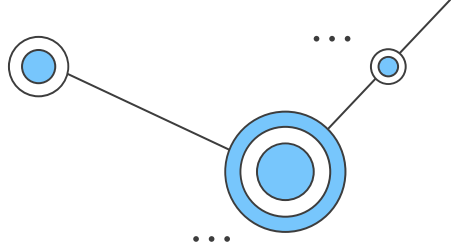


Neg Elias





O que é Git e GitHub?

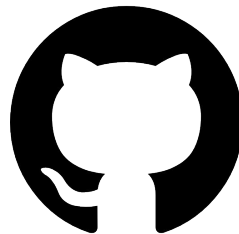
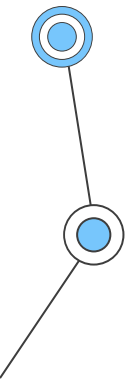


Git

O **Git** é um sistema de controle de versão de arquivos. É um software livre e muito utilizado no desenvolvimento de software onde diversas pessoas estão contribuindo simultaneamente, podendo criar e editar arquivos.

GitHub

O **GitHub** é uma plataforma onde você pode armazenar seus projetos. É como se fosse uma rede social, só que de códigos, onde seus desenvolvedores podem disponibilizá-los para outras pessoas verem.



Por que usar essas tecnologias?

01

Sistema de Controle de Versão

02

Auxiliar o Trabalho em Equipe

03

Git é economia de tempo

04

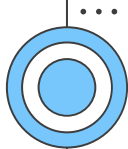
Registra Seus Andamento no Projeto.



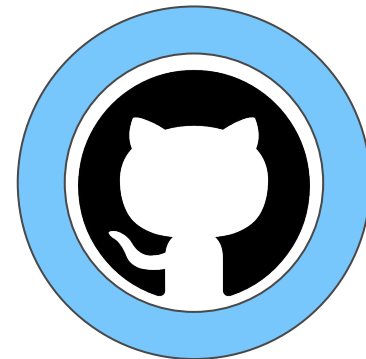


01

Criando uma Conta no GitHub



Passo a Passo



1º você entra no site do **GitHub** <https://github.com/>

2º clicar em **Sign up**

3º Digitar seu **email**(Enter your email) e **senha**(Create a password)

4º Criar um nome de **Usuário** (Enter a username)

5º Criar Conta

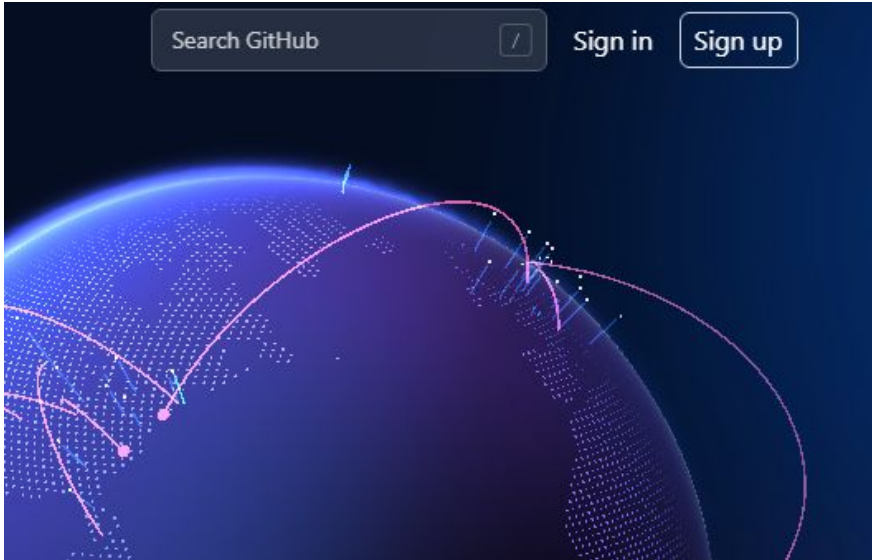
6º Verificação do Email

7º Conta Criada

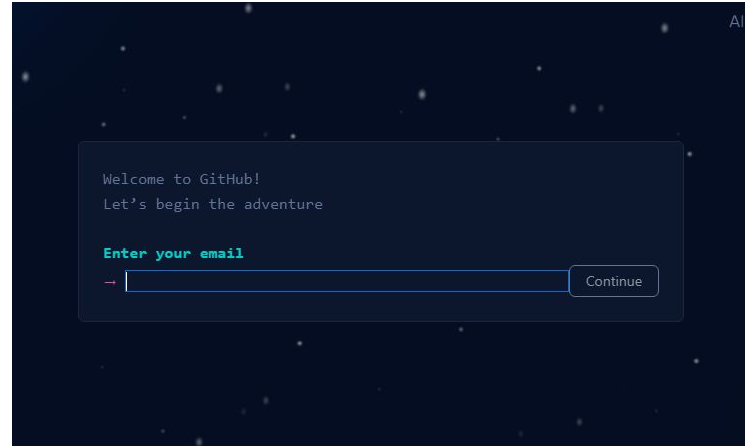
...

Prints do Passo a Passo

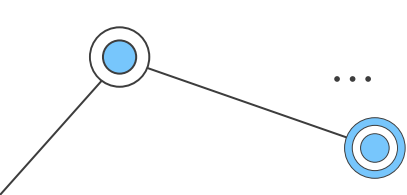
1º você entra no site do GitHub <https://github.com/>



2º clicar em Sign up

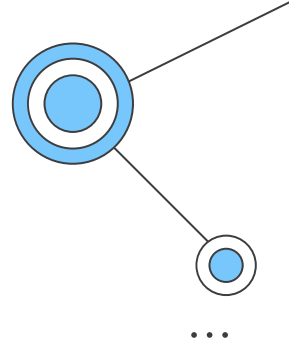


...



3º Digitar seu email(Enter your email) e senha(Create a password)

4º Criar um nome de Usuário (Enter a username)



Welcome to GitHub!
Let's begin the adventure

Enter your email
✓ scriptequipe25@gmail.com

Create a password
✓

Enter a username
→ ScriptEquipe

ScriptEquipe is available.

Bem-vindo ao GitHub!
Vamos começar a aventura

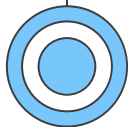
Digite seu e-mail
✓ scriptequipe25@gmail.com

Criar uma senha
✓

Digite um nome de usuário
✓ ScriptEquipe

Deseja receber atualizações e anúncios de produtos por e-mail?
Digite "y" para sim ou "n" para não

→ n



5º Criar Conta

Deseja receber atualizações e anúncios de produtos por e-mail?
Digite "y" para sim ou "n" para não

✓ n

Verifique sua conta

Criar Conta

6º Verificação do Email Código de acesso

 Your GitHub launch code Inbox x



GitHub <noreply@github.com>
to me ▾

11:40 PM (0 minutes ago)



Here's your GitHub launch code, @ScriptEquipe!




Continue signing up for GitHub by entering the code
below:

18223934


Open GitHub

7º Conta Criada



ScriptEquipe

[Edit profile](#)

 Joined 5 minutes ago

[Overview](#) [Repositories](#) [Projects](#) [Packages](#) [Stars](#)


Popular repositories

[Customize your pins](#)

You don't have any public repositories yet.

1 contribution in the last year [Contribution settings](#)

	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
Mon													
Wed													
Fri													

[Learn how we count contributions](#) [Less](#)  [More](#)

This is your **contribution graph**. Your first square is for joining GitHub and you'll earn more as you make **additional contributions**. More contributions means a darker green square for that day. Over time, your chart might start looking **something like this**.

We have a quick guide that will show you how to create your first repository and earn more green squares!

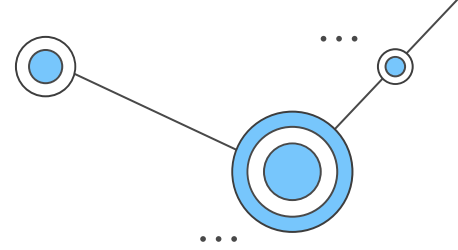
[Read the Hello World guide](#)

02

Criando um Repositório no GitHub



Passo a Passo



1º Clicar em **Repositories**

2º Clicar em **New**

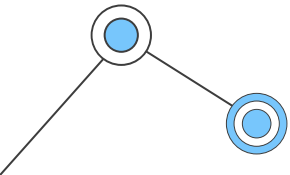
3º Digitar o Nome do seu **Repositório (CamelCase)**
(CamelCase) Ela é uma convenção de nomenclatura usada por developers para deixar o seu código mais legível para outros DEVs.

4º Definir se quer **Privado** ou **Público**

5º Adicionar um **Readme**(pois sua Descrição aparecerá nele)
Readme você pode escrever em **HTML** ou em **Markdown**
Linguagens de Marcação

6º Clicar em **Create Repository**

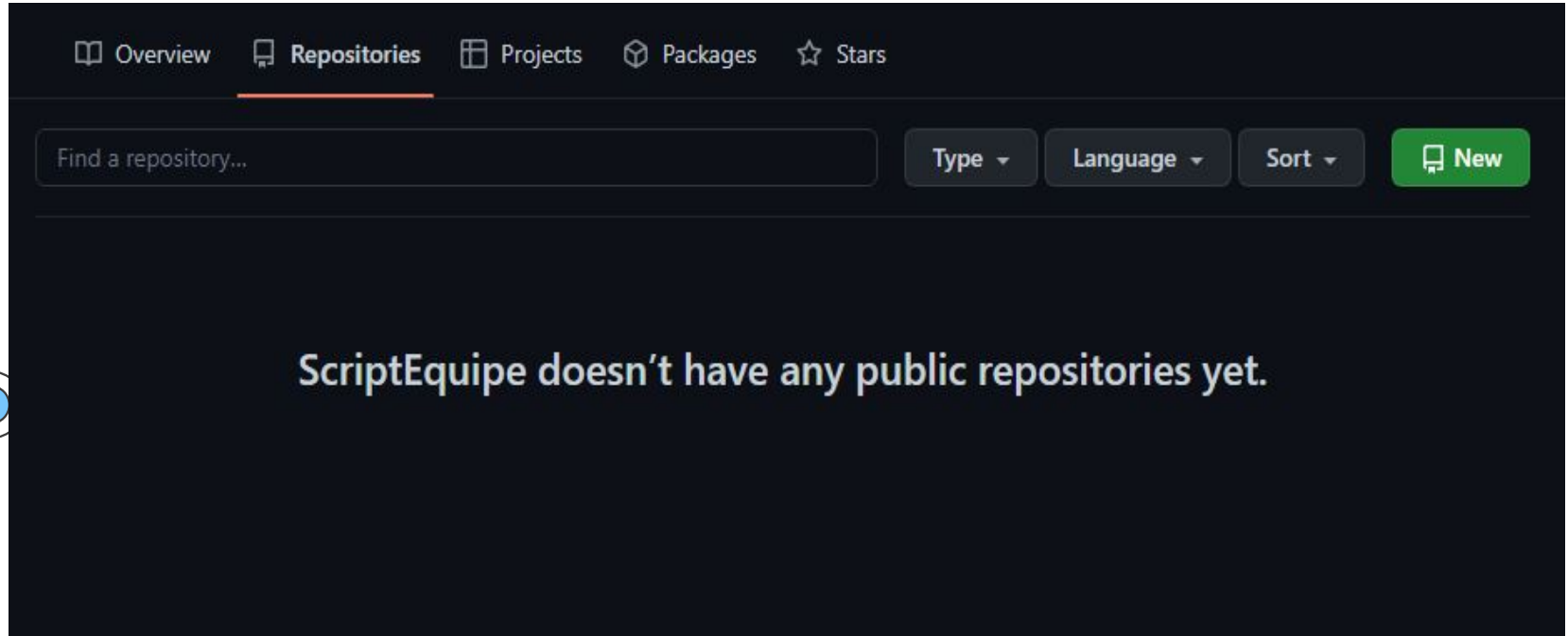
7º **Repositorio Criado**



Prints do Passo a Passo

1º Clicar em Repositories

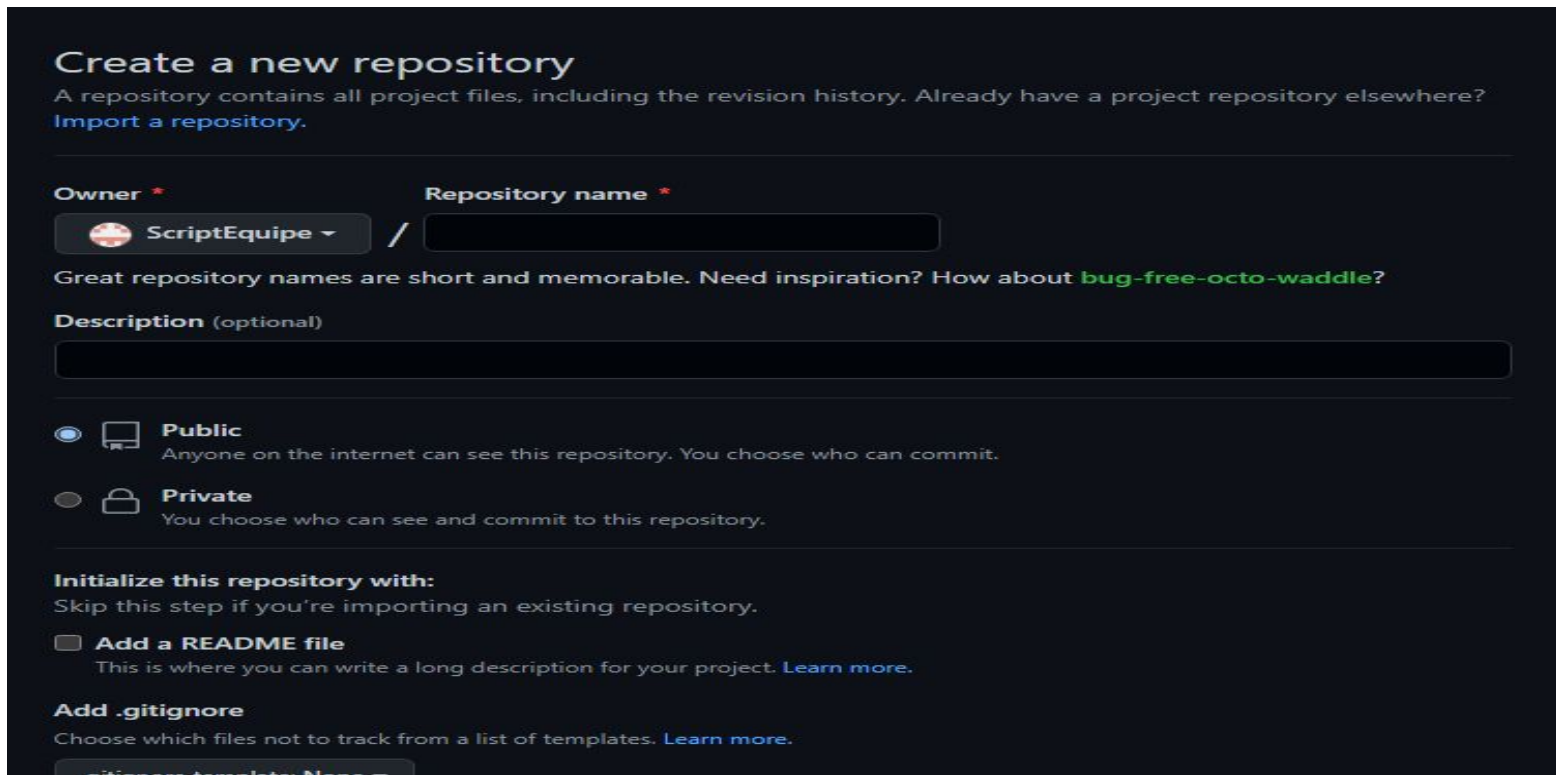
2º Clicar em New






3º Digitar o Nome do seu Repositório (CamelCase)
No caso foi PrimeiroCode

4º Definir se quer Privado ou Público




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 ***

 ScriptEquipe ▾ /

Great repository names are short and memorable. Need inspiration? How about **bug-free-octo-waddle?**

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.](#)



5º Adicionar um Readme(pois sua Descrição aparecerá nele)
Readme você pode escrever em HTML ou em Markdown
Linguagens de Marcação

6º Clicar em Create
Repository

Description (optional)

Teste para o Trabalho para Anhanguera

☒ **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** ▼

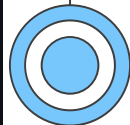
Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: **None** ▼

This will set  **main** as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository



7º Repositorio Criado

The screenshot shows a GitHub repository page for 'ScriptEquipe / PrimeiroCode'. The repository is public and has a 'main' branch with 1 branch and 0 tags. The commit history shows an 'Initial commit' by 'ScriptEquipe' with the file 'README.md'. The README content includes the title 'PrimeiroCode' and the subtitle 'Teste para o Trabalho para Anhanguera'.

ScriptEquipe / PrimeiroCode Public

<> Code • Issues 🔗 Pull requests ▶ Actions 📁 Projects 📖 Wiki 🛡 Security 📈 Insights ⚙ Settings

🔗 main ▾ 🔗 1 branch 🔗 0 tags Go to file Add file ▾ Code ▾

👤 ScriptEquipe Initial commit 5a3e980 now 🕒 1 commit

📄 README.md Initial commit now

README.md

PrimeiroCode

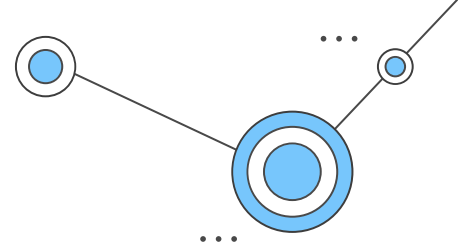
Teste para o Trabalho para Anhanguera

03

Adicionando os Arquivos no seu Repositório



Passo a Passo



1º Baixar e Instalar o GIT <https://git-scm.com/>

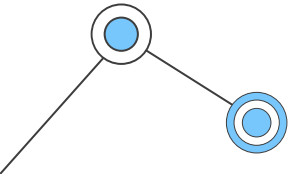
2º Abrir o Terminal do GIT bash
(pode ser na pasta ou via terminal usando os comandos)

3º Configurar seu Email e Nome
git config --global user.name ""
git config --global user.email ""

4º Iniciar um Repositório
git init

5º Adicionar os Arquivos na Área de Staging
git add .

6º Verificar se foram Adicionados
git status
Verde ok
Vermelho não



Passo a Passo

7º Fazer o commit

git commit -m "Primeiro commit"

8º Mudar a branch para main (Boa praticar feita por Desenvolvedores)

git branch -M "main"

9º Fazer o link com seu Repositório no Github

git remote add origin https://github.com/ScriptEquipe/PrimeiroCode.git

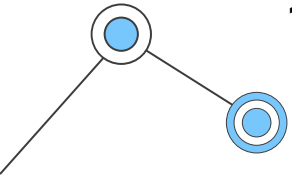
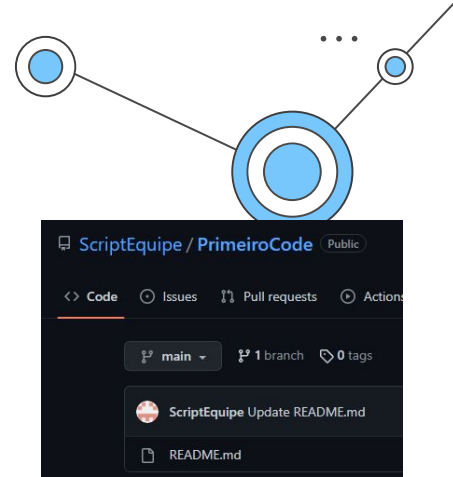
10º Fazer o Push

git push -u origin main

11º caso tiver algum impedimento você pode fazer isso

git push -force origin main

12º Verificar se Realmente esta no Repositorio

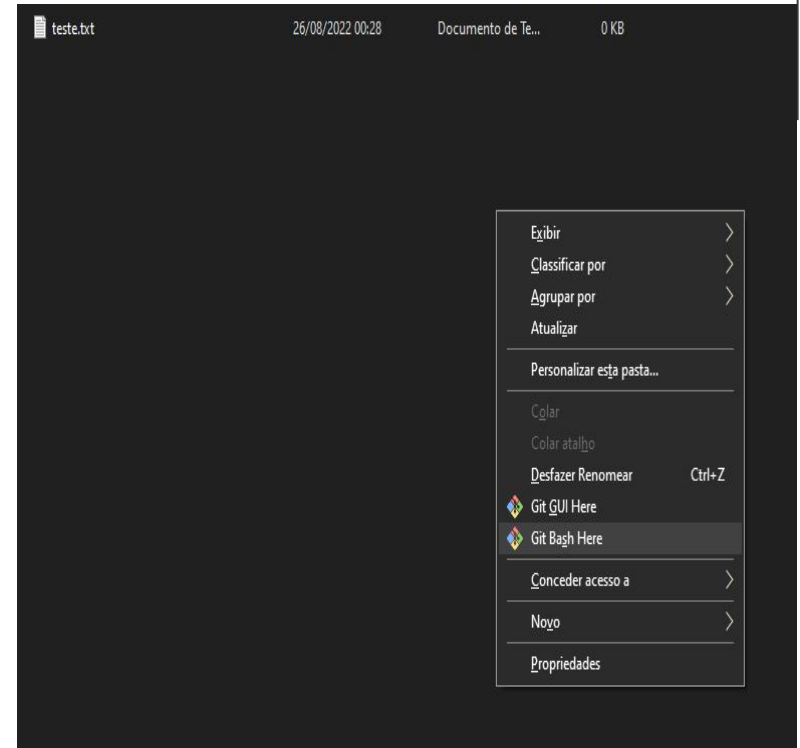


Prints do Passo a Passo

1º Baixar e Instalar o GIT <https://git-scm.com/>



2º Abrir o Terminal do GIT bash





Prints do Passo a Passo

3º Configurar seu Email e Nome

git config --global user.name ""

git config --global user.email ""



```
MINGW64:/c/Users/PAULO DEYVID/Downloads/teste git
PAULO DEYVID@DESKTOP-5UHFB7S MINGW64 ~/Downloads/teste git (master)
$ git config --global user.name "Script"

PAULO DEYVID@DESKTOP-5UHFB7S MINGW64 ~/Downloads/teste git (master)
$ git config --global user.name
Script

PAULO DEYVID@DESKTOP-5UHFB7S MINGW64 ~/Downloads/teste git (master)
$
```

```
MINGW64:/c/Users/PAULO DEYVID/Downloads/teste git
PAULO DEYVID@DESKTOP-5UHFB7S MINGW64 ~/Downloads/teste git (master)
$ git config --global user.email "scriptequipe25@gmail.com"

PAULO DEYVID@DESKTOP-5UHFB7S MINGW64 ~/Downloads/teste git (master)
$ git config --global user.email
scriptequipe25@gmail.com

PAULO DEYVID@DESKTOP-5UHFB7S MINGW64 ~/Downloads/teste git (master)
$ |
```

Prints do Passo a Passo

4º Iniciar um Repositório git init

```
MINGW64:/c/Users/PAULO DEYVID/Downloads/teste git
PAULO DEYVID@DESKTOP-5UHF87S MINGW64 ~/Downloads/teste git (master)
$ git init
Initialized empty Git repository in C:/Users/PAULO DEYVID/Downloads/teste git/.git/
PAULO DEYVID@DESKTOP-5UHF87S MINGW64 ~/Downloads/teste git (master)
$ |
```

5º Adicionar os Arquivos na Área de Staging git add .

```
MINGW64:/c/Users/PAULO DEYVID/Downloads/teste git
PAULO DEYVID@DESKTOP-5UHF87S MINGW64 ~/Downloads/teste git (master)
$ git add .

PAULO DEYVID@DESKTOP-5UHF87S MINGW64 ~/Downloads/teste git (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   teste.txt

PAULO DEYVID@DESKTOP-5UHF87S MINGW64 ~/Downloads/teste git (master)
$
```

Prints do Passo a Passo

6º Verificar se foram Adicionados

git status

Verde ok Vermelho não

```
PAULO DEYVID@DESKTOP-5UHFB7S MINGW64 ~/Downloads/teste git (master)
$ git status
On branch master

No commits yet

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

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

```
PAULO DEYVID@DESKTOP-5UHFB7S MINGW64 ~/Downloads/teste git (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
       new file:   teste.txt
```

...

Prints do Passo a Passo

7º Fazer o commit

git commit -m "Primeiro commit"

```
MINGW64:/c/Users/PAULO DEYVID/Downloads/teste git
PAULO DEYVID@DESKTOP-5UHF87S MINGW64 ~/Downloads/teste git (master)
$ git commit -m "Primeiro Commit"
[master (root-commit) 7a89645] Primeiro Commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 teste.txt

PAULO DEYVID@DESKTOP-5UHF87S MINGW64 ~/Downloads/teste git (master)
$
```

8º Mudar a branch para main

git branch -M "main"

```
MINGW64:/c/Users/PAULO DEYVID/Downloads/teste git
PAULO DEYVID@DESKTOP-5UHF87S MINGW64 ~/Downloads/teste git (master)
$ git commit -m "Primeiro Commit"
[master (root-commit) 7a89645] Primeiro Commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 teste.txt

PAULO DEYVID@DESKTOP-5UHF87S MINGW64 ~/Downloads/teste git (master)
$ git branch -M "main"

PAULO DEYVID@DESKTOP-5UHF87S MINGW64 ~/Downloads/teste git (main)
$ |
```


Prints do Passo a Passo

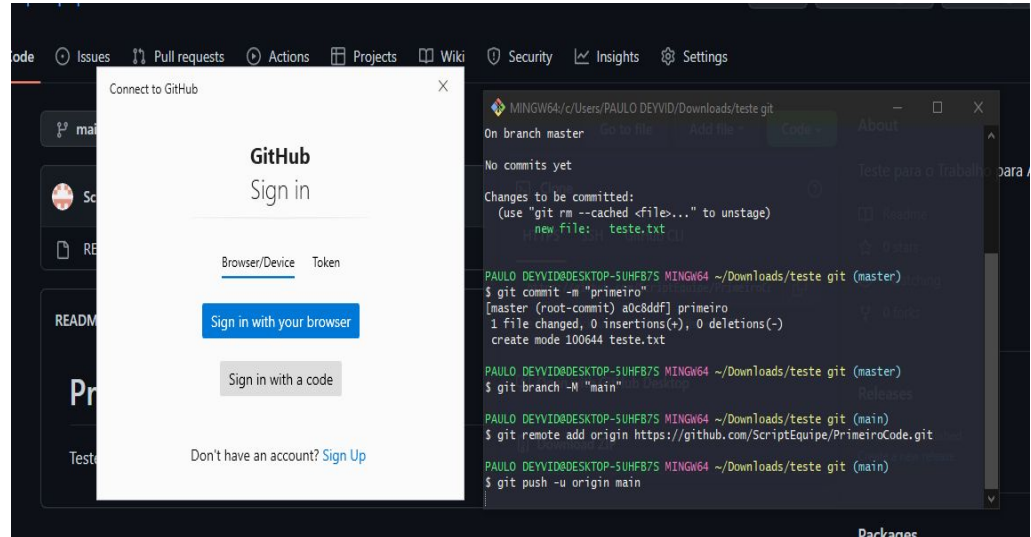
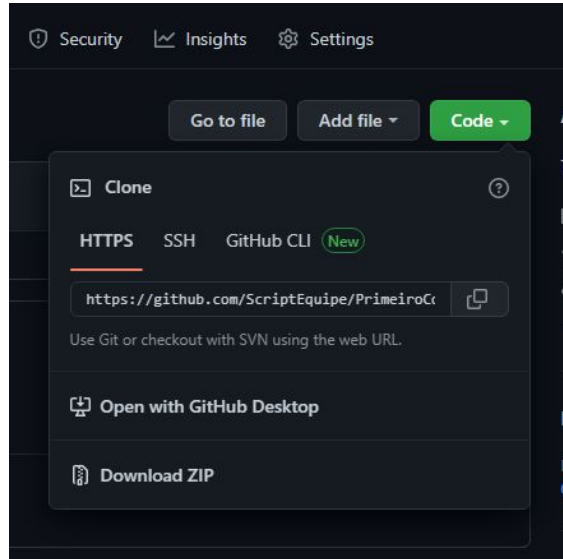
9º Fazer o link com seu Repositório no Github

git remote add origin

<https://github.com/ScriptEquipe/PrimeiroCode.git>

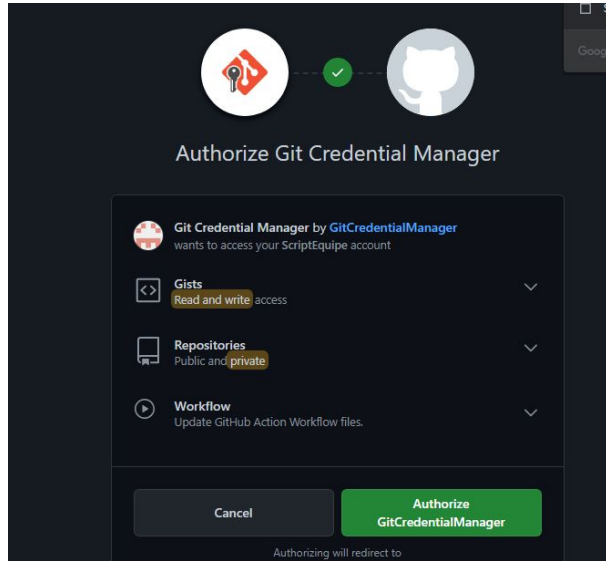
10º Fazer o Push

git push -u origin main



Ao fazer o push o github pede uma autorização

Prints do Passo a Passo



11º caso tiver algum impedimento você pode fazer isso
git push -force origin main

```
MINGW64:/c/Users/PAULO DEYVID/Downloads/teste git
[master (root-commit) a0c8ddf] primeiro
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 teste.txt

PAULO DEYVID@DESKTOP-5UHF875 MINGW64 ~/Downloads/teste git (master)
$ git branch -M "main"

PAULO DEYVID@DESKTOP-5UHF875 MINGW64 ~/Downloads/teste git (main)
$ git remote add origin https://github.com/ScriptEquipe/PrimeiroCode.git

PAULO DEYVID@DESKTOP-5UHF875 MINGW64 ~/Downloads/teste git (main)
$ git push -u origin main
To https://github.com/ScriptEquipe/PrimeiroCode.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/ScriptEquipe/PrimeiroCode.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

PAULO DEYVID@DESKTOP-5UHF875 MINGW64 ~/Downloads/teste git (main)
$
```

Prints do Passo a Passo

11º caso tiver algum impedimento você pode fazer isso
git push -force origin main

```
MINGW64:/c/Users/PAULO DEYVID/Downloads/teste git
[master (root-commit) a0c8ddf] primeiro
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 teste.txt

PAULO DEYVID@DESKTOP-SUHF87S MINGW64 ~/Downloads/teste git (master)
$ git branch -M "main"

PAULO DEYVID@DESKTOP-SUHF87S MINGW64 ~/Downloads/teste git (main)
$ git remote add origin https://github.com/ScriptEquipe/PrimeiroCode.git

PAULO DEYVID@DESKTOP-SUHF87S MINGW64 ~/Downloads/teste git (main)
$ git push -u origin main
To https://github.com/ScriptEquipe/PrimeiroCode.git
! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/ScriptEquipe/PrimeiroCode.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

PAULO DEYVID@DESKTOP-SUHF87S MINGW64 ~/Downloads/teste git (main)
$
```

```
MINGW64:/c/Users/PAULO DEYVID/Downloads/teste git
PAULO DEYVID@DESKTOP-SUHF87S MINGW64 ~/Downloads/teste git (main)
$ git push -u origin main
To https://github.com/ScriptEquipe/PrimeiroCode.git
! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/ScriptEquipe/PrimeiroCode.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

PAULO DEYVID@DESKTOP-SUHF87S MINGW64 ~/Downloads/teste git (main)
$ git push --force origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 210 bytes | 210.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ScriptEquipe/PrimeiroCode.git
+ 5335345...a0c8ddf main -> main (forced update)
```

Prints do Passo a Passo

11º caso tiver algum impedimento você pode fazer isso
git push -force origin main

```
MINGW64:/c/Users/PAULO DEYVID/Downloads/teste git
[master (root-commit) a0c8ddf] primeiro
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 teste.txt

PAULO DEYVID@DESKTOP-SUHF87S MINGW64 ~/Downloads/teste git (master)
$ git branch -M "main"

PAULO DEYVID@DESKTOP-SUHF87S MINGW64 ~/Downloads/teste git (main)
$ git remote add origin https://github.com/ScriptEquipe/PrimeiroCode.git

PAULO DEYVID@DESKTOP-SUHF87S MINGW64 ~/Downloads/teste git (main)
$ git push -u origin main
To https://github.com/ScriptEquipe/PrimeiroCode.git
! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/ScriptEquipe/PrimeiroCode.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

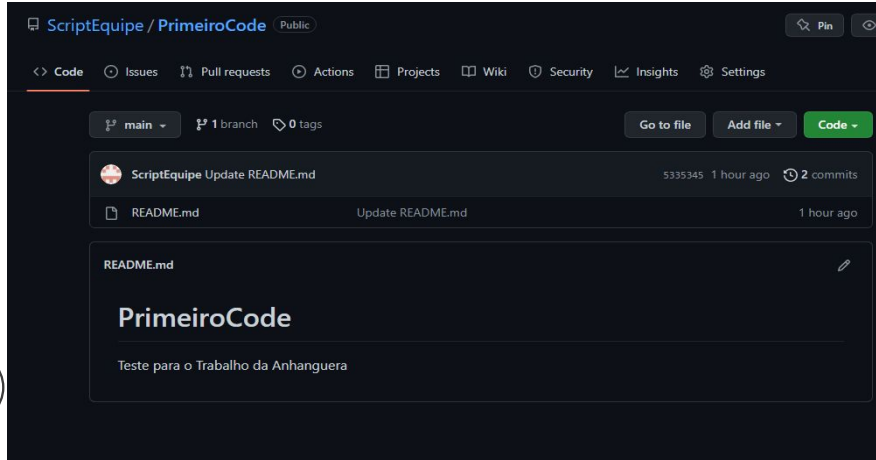
PAULO DEYVID@DESKTOP-SUHF87S MINGW64 ~/Downloads/teste git (main)
$
```

```
MINGW64:/c/Users/PAULO DEYVID/Downloads/teste git
PAULO DEYVID@DESKTOP-SUHF87S MINGW64 ~/Downloads/teste git (main)
$ git push -u origin main
To https://github.com/ScriptEquipe/PrimeiroCode.git
! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/ScriptEquipe/PrimeiroCode.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

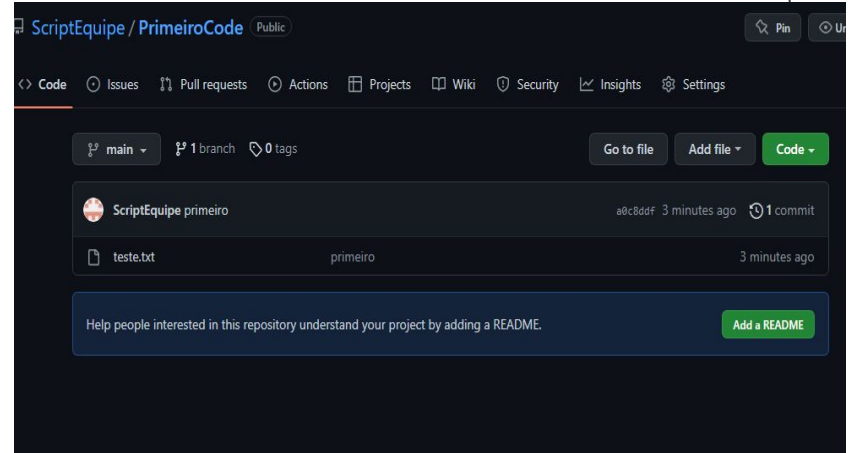
PAULO DEYVID@DESKTOP-SUHF87S MINGW64 ~/Downloads/teste git (main)
$ git push --force origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 210 bytes | 210.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ScriptEquipe/PrimeiroCode.git
+ 5335345...a0c8ddf main -> main (forced update)
```

Prints do Passo a Passo

12º Verificar se Realmente esta no Repositorio



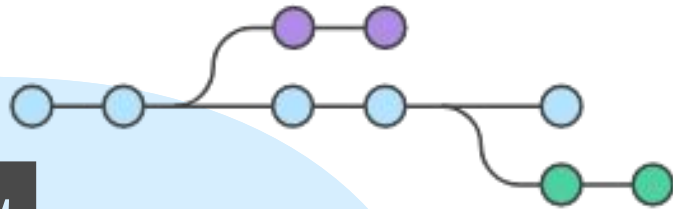
Sem o arquivo



Push realizado com sucesso

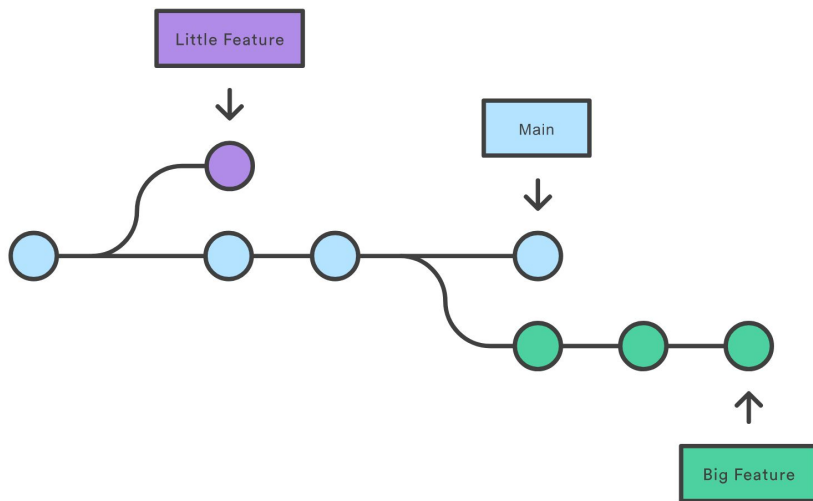
Branch Merge Clone Pull Fork Pull request

04





Branch

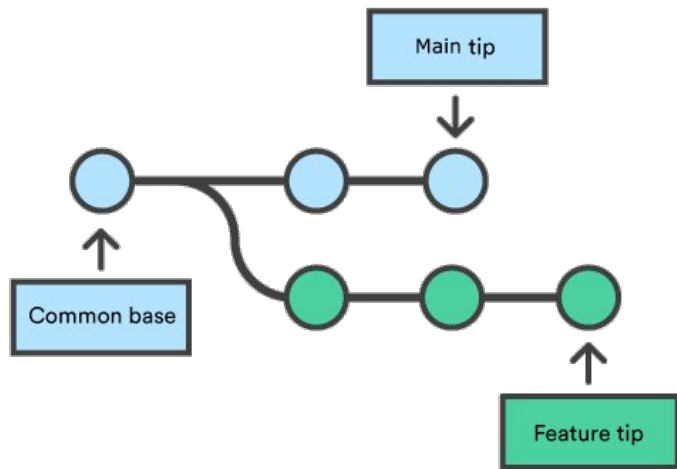


O comando **git branch** permite criar, listar, renomear e excluir ramificações. Ele não permite alternar entre as ramificações ou reunir um histórico bifurcado de novo. Por esse motivo, o comando **git branch** é muito integrado com os comandos **git checkout** e **git merge**

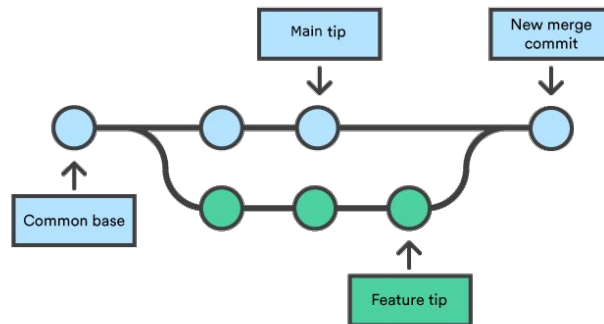




Merge

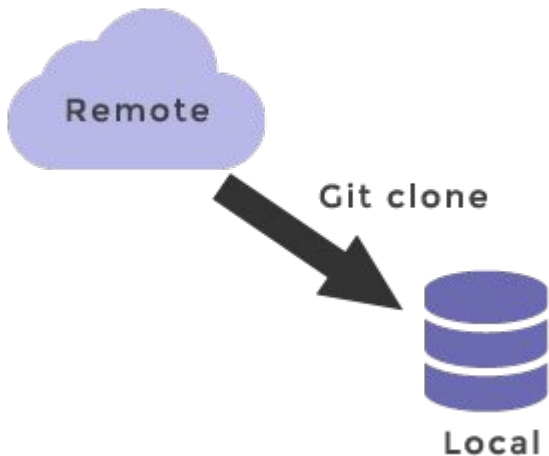


O Git Merge vai combinar várias sequências de commits em um histórico unificado. Nos casos de uso mais frequentes, o git merge é utilizado para combinar dois branches.





Clone

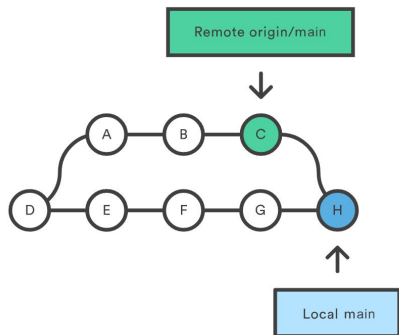
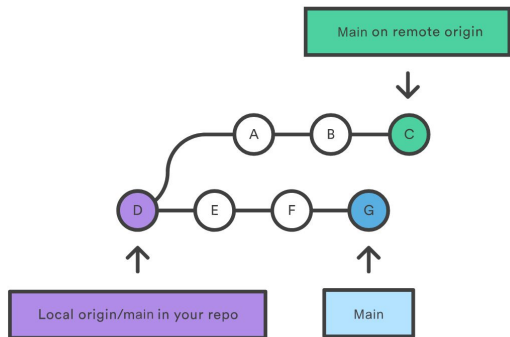


git clone é um utilitário de linha de comando que é usado para selecionar um repositório existente e criar um clone ou cópia do repositório de destino.

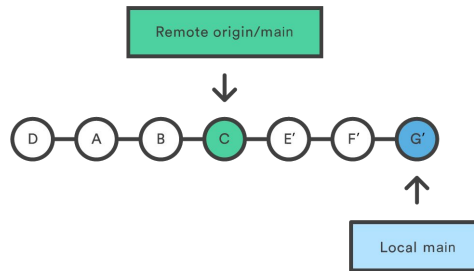




Pull

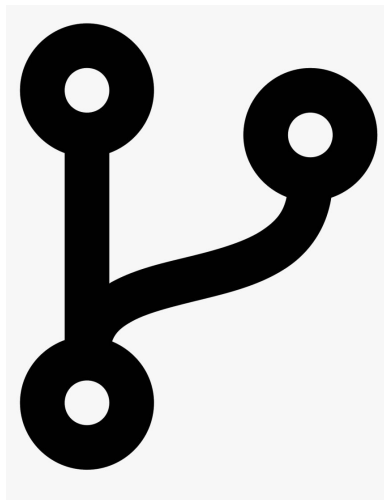
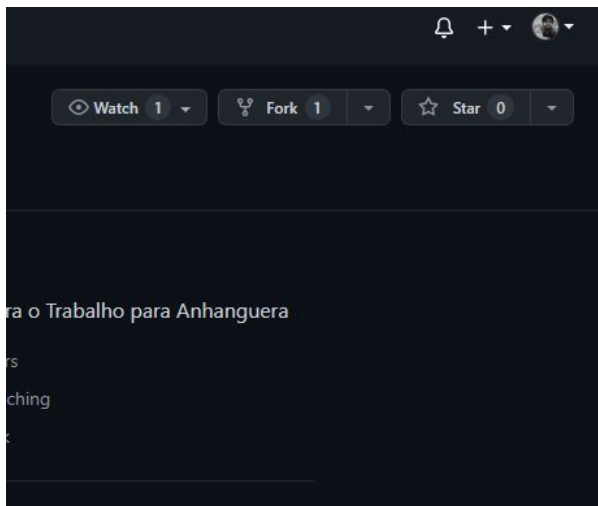


O comando **git pull** é usado para buscar e baixar conteúdo de repositórios remotos e fazer a atualização imediata ao repositório local para que os conteúdos sejam iguais.





Fork



Fork nada mais é que **uma cópia de um repositório**. Esta cópia vira um clone do estado atual do repositório, fazendo assim com que você possa experimentar mudanças futuras e novas sem precisar utilizar o repositório principal.





Pull Request



The screenshot shows the GitHub interface for a repository named 'kedark3 / Demo'. At the top, there are buttons for 'Unwatch', 'Star', and 'Fork'. Below this is a navigation bar with links for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The main heading is 'Open a pull request', followed by a subtext: 'Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).' Below this is a comparison bar showing 'base: master' and 'compare: new_branch', with a green checkmark and the text 'Able to merge. These branches can be automatically merged.' The main content area is titled 'Adding a test file to new_branch' and includes a 'Write' tab, a 'Preview' tab, and a large text area for comments. To the right of the main content are sections for 'Reviewers', 'Assignees', 'Labels', 'Projects', and 'Milestone'. At the bottom, there is a green 'Create pull request' button. Below the main content area, there is a summary bar showing '1 commit', '1 file changed', '0 commit comments', and '1 contributor'. The bottom section shows a commit by 'kedark3' on Jul 17, 2019, with the message 'Adding a test file to new_branch'.

Pull Request quer dizer **solicitação de puxar**, em uma tradução livre. E isso está diretamente ligado ao fato de que, ao enviar a notificação, às demais pessoas desenvolvedoras saberão que precisam fazer o merge do código na branch principal.





05

Extras





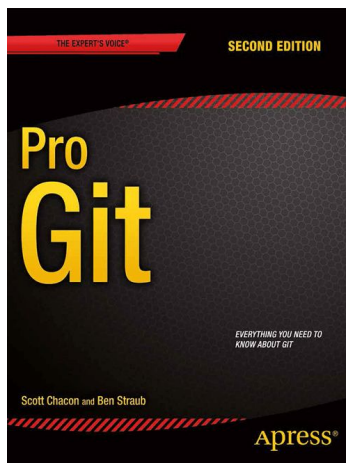
Links e Livros

<https://github.com/PDEYVID/GitTutorial>

<https://git-scm.com/book/en/v2>

<https://www.udemy.com/course/git-e-github-para-iniciantes/learn/lecture/8852726?start=105#overview>

<https://www.atlassian.com/br/git/tutorials>



Controlando versões com
Git e GitHub



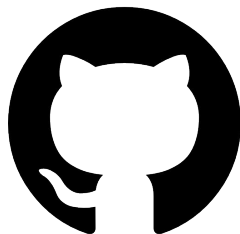
Casa do Código | alura

ALEXANDRE AQUILAS
RODRIGO FERREIRA

OBRIGADO



" Seja você o Github, o controle de versões da sua vida. E a cada atualização mal sucedida, retorne a sua melhor versão! "



<https://github.com/ScriptEquipe>