

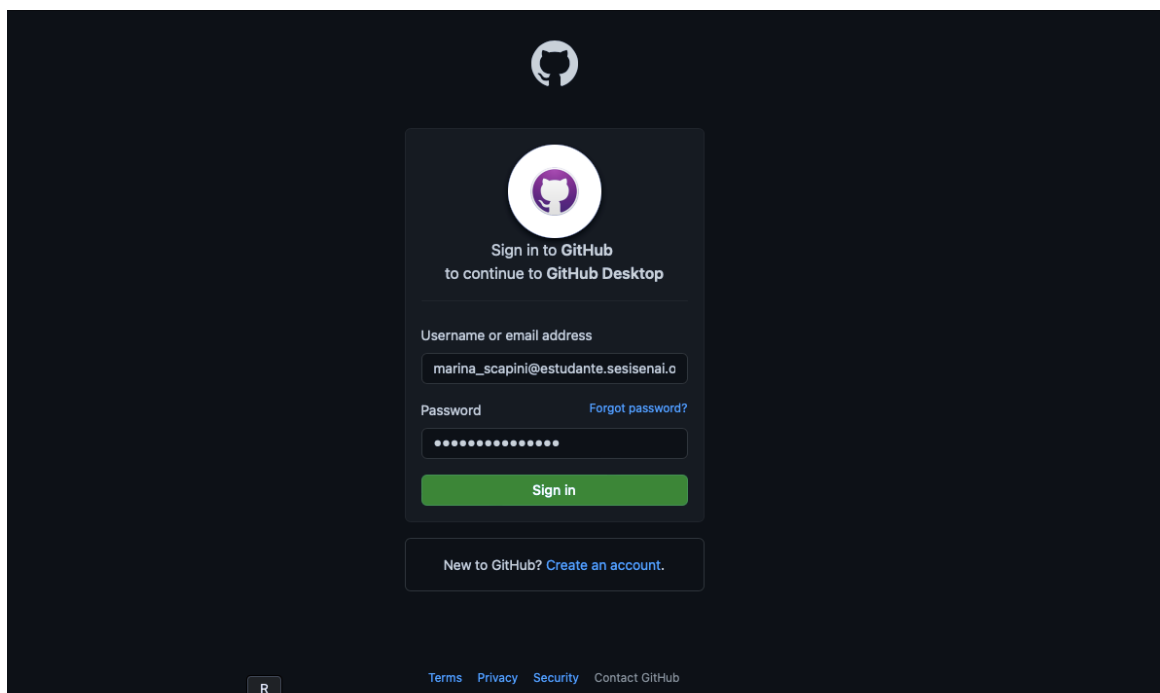
## Instalação e criação de repositórios do GitHub no macOS

Em primeiro lugar, é preciso compreender que tanto o Git quanto o GitHub são importantes ferramentas para o trabalho do Desenvolvedor, porém, não são a mesma coisa:

**Git:** utilizado para fazer o controle de versões existentes de um projeto (versionamento) e também utilizado para a criação de repositórios, que irão conter todos os arquivos de um determinado projeto. O Git é acessado a partir de linhas de comando no terminal da máquina.

**GitHub:** é uma plataforma que guarda os repositórios criados pelo Git. Nela, qualquer usuário cadastrado pode contribuir em diversos projetos.

Em primeiro lugar, vamos criar uma conta no GitHub, como exemplifica a imagem abaixo:



Agora que você já tem uma conta no GitHub, pode criar um repositório e começar a fazer o upload de seus códigos.


Vamos criar um novo repositório. Escolha um nome para o seu repositório, adicione um arquivo “README” e clique em *Create repository*:

**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 \***

 marinascapini /

Great repository names are short and memorable. Need inspiration? How about [congenial-carnival](#)?

**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](#).

**.gitignore template:** None ▾

☐ **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.

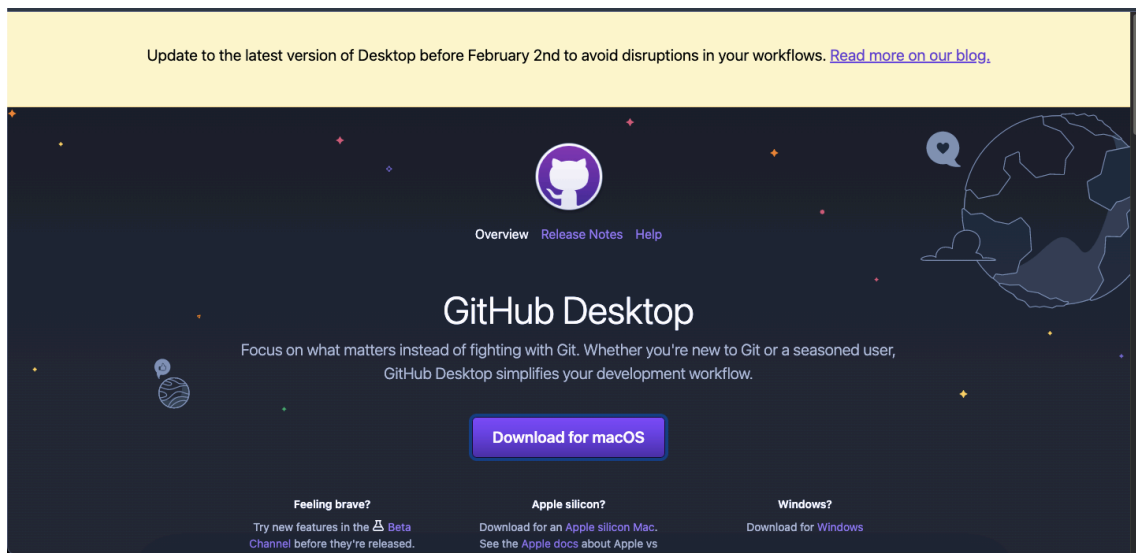
[Create repository](#)

---

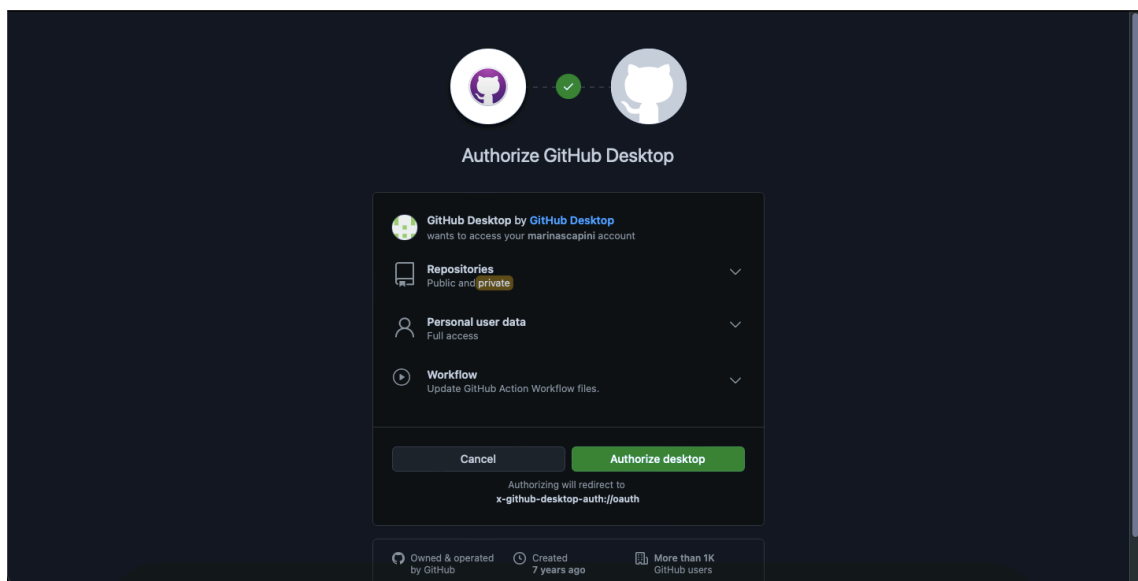
 © 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

Pronto, agora você já possui um repositório para organizar os códigos e projetos que você irá elaborar.

Você também pode fazer o download do GitHub Desktop, para facilitar o gerenciamento de seus repositórios. Para isso, é preciso efetuar o download do programa em <https://desktop.github.com/>.

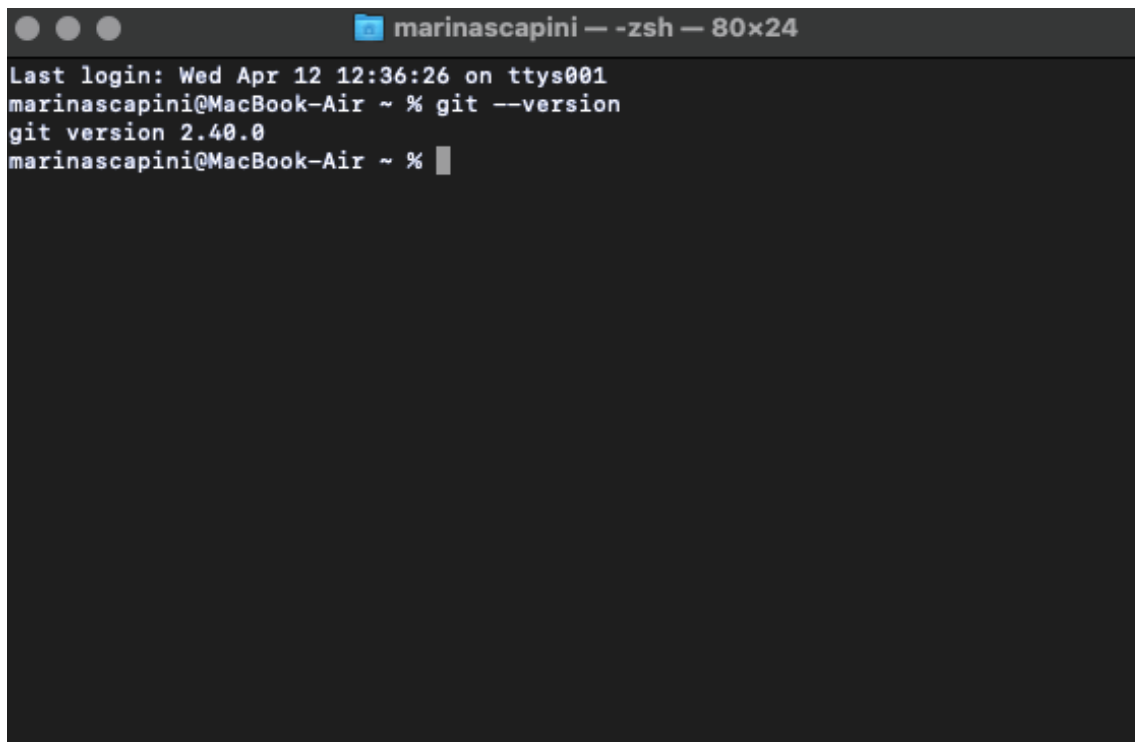


Depois, é preciso autorizar o programa e fazer o login em sua conta GitHub:



Com o passar do tempo, é natural que os projetos e códigos se acumulem, e que recebam vários updates com o passar do tempo, o que pode dificultar o controle de versões e atualizações. Para auxiliá-lo nisso, pode-se utilizar o Git para manter os projetos sempre atualizados.

Primeiramente, é preciso verificar se sua máquina já possui alguma versão do Git. Para isso, no Terminal, digite o seguinte comando:

A screenshot of a macOS terminal window. The title bar at the top reads "marinascapini — -zsh — 80x24". The terminal content shows the last login time as "Wed Apr 12 12:36:26 on ttys001". The user "marinascapini" is at the "MacBook-Air" prompt. They have entered the command "git --version", and the terminal has responded with "git version 2.40.0". The prompt is now waiting for the next command.

```
marinascapini@MacBook-Air ~ % git --version
git version 2.40.0
marinascapini@MacBook-Air ~ %
```

Se seu computador não possuir ainda nenhuma versão do Git, você poderá instalado em um macOS usando o Homebrew. Para instalá-lo, digite o seguinte comando no terminal:

```
/bin/bash -c "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

O terminal irá requisitar sua senha, e, a seguir, irá prosseguir com a instalação do Homebrew:

```
marinascapini@MacBook-Air ~ % git --version
git version 2.40.0
marinascapini@MacBook-Air ~ % /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
Checking for 'sudo' access (which may request your password)...
Password:
This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew
Press RETURN/ENTER to continue or any other key to abort:
/usr/bin/sudo /usr/sbin/chown -R marinascapini:admin /usr/local/Homebrew
Downloading and installing Homebrew...
remote: Enumerating objects: 2790, done.
remote: Counting objects: 100% (1680/1680), done.
remote: Compressing objects: 100% (280/280), done.
remote: Total 2790 (delta 1392), reused 1643 (delta 1369), pack-reused 1110
Receiving objects: 100% (2790/2790), 1.30 MiB | 4.64 MiB/s, done.
Resolving deltas: 100% (1735/1735), completed with 338 local objects.
From https://github.com/Homebrew/brew
 * [new tag]         4.0.11   -> origin/master
 * [new tag]         4.0.12   -> 4.0.11
Updating files: 100% (3286/3286), done.
HEAD is now at 8c69c954 Merge pull request #15206 from 8098/portable-ruby-2.6.10.1
Updated 2 taps (homebrew/core and homebrew/cask).
Installation successful!
Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).
Homebrew is run entirely by unpaid volunteers. Please consider donating:
https://github.com/Homebrew/brew#donations
Next steps:
- Run these two commands in your terminal to add Homebrew to your PATH:
  (echo; echo 'eval "$(brew shellenv)"') >> /Users/marinascapini/.zprofile
  eval "$(brew shellenv)"
- Run brew help to get started
- Further documentation:
  https://docs.brew.sh
marinascapini@MacBook-Air ~ %
```

Depois de instalar o Homebrew, você pode instalar o Git digitando o seguinte comando no terminal:

```
brew install git
```

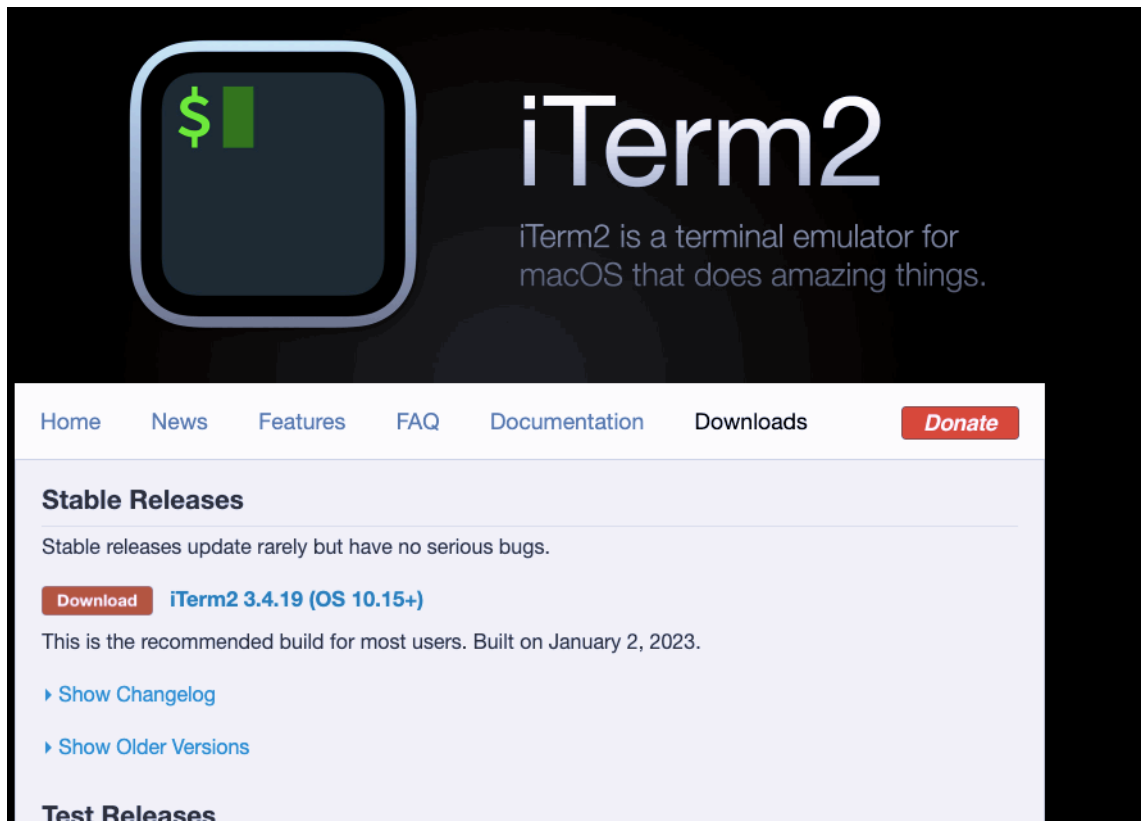
Esse comando irá permitir que o Homebrew faça o download do Git para seu computador. Depois da instalação, você irá conseguir verificar a versão git instalada a partir do comando:

```
git --version
```

Você deverá ver a versão do Git sendo exibida no terminal, como exemplificado abaixo:

```
marinascapini@MacBook-Air ~ % git --version
git version 2.40.0
marinascapini@MacBook-Air ~ %
```

Para utilizar o Git Bash em um Mac, é necessário a instalação de um emulador de terminal, como o iTerm2, disponível em <https://iterm2.com/downloads.html>.



Depois de baixar e instalar o iTerm2, você irá poder abrir uma janela do terminal e executar comandos do Git Bash normalmente. Com o Git, você poderá subir repositórios, arquivos e atualizações de projetos a partir do terminal.

Após a configuração do Git, você poderá utilizar comandos no terminal para realizar modificações em seu repositório do GitHub.

O comando `git add .` irá mandar os seus arquivos locais para uma espécie de host intermediário (até você fazer o *commit*!).

```

    new file:   "M\303\263dulo 01/Semana 02/.DS_Store"

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:   .DS_Store

marinascapini@MacBook-Air floripamaistec % git add .
marinascapini@MacBook-Air floripamaistec % git branch -r
origin/HEAD -> origin/main
origin/main
marinascapini@MacBook-Air floripamaistec % git branch
* main
marinascapini@MacBook-Air floripamaistec % git branch -a
* main
remotes/origin/HEAD -> origin/main
remotes/origin/main
marinascapini@MacBook-Air floripamaistec % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .DS_Store
    new file:   .Rhistory
    new file:   "M\303\263dulo 01/.DS_Store"
    new file:   "M\303\263dulo 01/Semana 01/.DS_Store"
    new file:   "M\303\263dulo 01/Semana 02/.DS_Store"

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:   .DS_Store
    modified:   "M\303\263dulo 01/Semana 01/.DS_Store"

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    "M\303\263dulo 01/Semana 01/Exerc\303\255cio 01/"

marinascapini@MacBook-Air floripamaistec % git add .
marinascapini@MacBook-Air floripamaistec % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .DS_Store
    new file:   .Rhistory
    new file:   "M\303\263dulo 01/.DS_Store"
    new file:   "M\303\263dulo 01/Semana 01/.DS_Store"
    new file:   "M\303\263dulo 01/Semana 01/Exerc\303\255cio 01/operadores.sln"
    new file:   "M\303\263dulo 01/Semana 02/.DS_Store"

marinascapini@MacBook-Air floripamaistec %

```

git add .

Após você adicionar as pastas/arquivos que quiser mandar para o repositório, é preciso digitar o comando git commit para garantir que as alterações foram salvas.

```

floripamaistec — -zsh — 121x30

Last login: Tue Feb  7 22:00:43 on ttys000
marinascapini@MacBook-Air floripamaistec % ls
Módulo 01      Módulo 02      Módulo 03      README.md      e
marinascapini@MacBook-Air floripamaistec % git add .
marinascapini@MacBook-Air floripamaistec % git commit -m
error: switch 'm' requires a value
marinascapini@MacBook-Air floripamaistec % git commit -m "datatypes"
[main b8f20a6] datatypes
 6 files changed, 25 insertions(+)
 create mode 100644 .DS_Store
 create mode 100644 "M\303\263dulo 01/.DS_Store"
 create mode 100644 "M\303\263dulo 01/Semana 01/.DS_Store"
 create mode 100644 "M\303\263dulo 01/Semana 01/Exerc\303\255cio 01/operadores.sln"
 create mode 100644 "M\303\263dulo 01/Semana 02/.DS_Store"
 create mode 100644 e
marinascapini@MacBook-Air floripamaistec % git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (12/12), 2.37 KiB | 1.18 MiB/s, done.
Total 12 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), done.
To github.com:marinascapini/floripamaistec.git
   1b02b57..b8f20a6  main -> main
marinascapini@MacBook-Air floripamaistec %

```