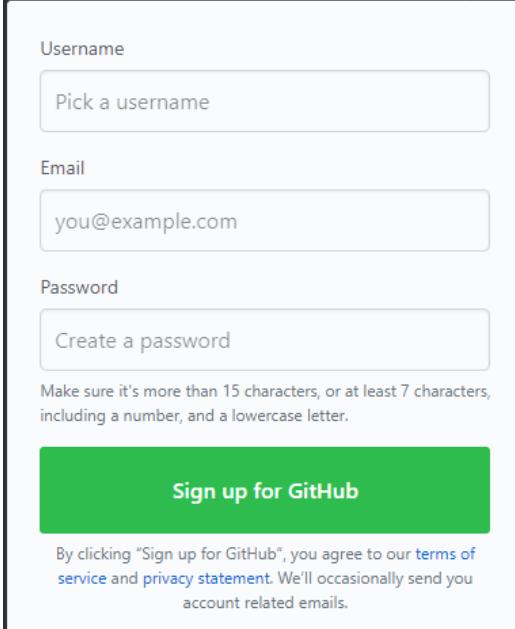


GitHub

Criando uma conta no GitHub

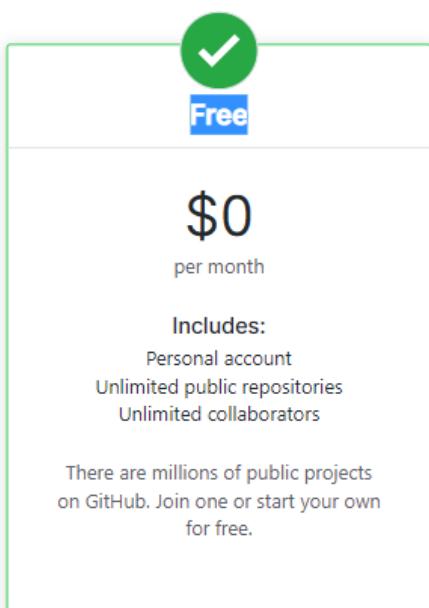
Acesse o site e clique em “Sign up for GitHub”



The screenshot shows the GitHub sign-up interface. It includes fields for Username (placeholder: "Pick a username"), Email (placeholder: "you@example.com"), and Password (placeholder: "Create a password"). Below the password field is a note: "Make sure it's more than 15 characters, or at least 7 characters, including a number, and a lowercase letter." A large green button labeled "Sign up for GitHub" is centered. At the bottom, a small note states: "By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails."

Preencha os dados da conta e resolva o “Captcha” proposto para validar que o acesso está sendo feito por um humano, depois clique em [Create an account](#).

No próximo passo devemos escolher um plano de uso, iremos optar pelo plano “Free” que irá nos atender em praticamente todas as necessidades.

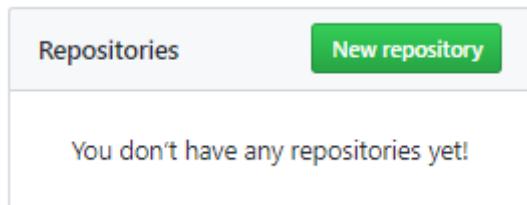


Para finalizar responda a um rápido questionário de perfil de usuário e clique em “Submit”. Se tudo estiver ok você será direcionado a uma página onde será possível criar um repositório.

Criando um Repositório.

Na janela de repositórios serão apresentados os repositórios que foram criados sem sua conta, caso não exista nenhum repositório crie em “New repository”.

Obs: caso não tenha acessado seu e-mail e validado a conta será solicitado que o faça.



Na tela que irá se abrir preencha os dados do novo projeto e ao terminar clique em “Create repository”.

Create a new repository

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

Owner Repository name

wagnerbarth / aulaTP ✓

Great repository names are short and memorable. Need inspiration? How about literate-happiness.

Description (optional)

Aula de uso de GitHub

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

Instale o aplicativo GitHub no seu computador, caso ele seja Windows ou Mac.

Para isso siga as instruções do documento “instalar-aplicativo-git-windows”.

Configurando o Git.

Existem 2 pequenos passos para configurar o seu GIT para ter um acesso mais simplificado ao github. Aqui estaremos estabelecendo que, sempre que necessitar, você irá fornecer o seu login e senha ao GitHub.

Então, com o seu terminal git aberto, vamos digitar:

```
$ git config --global user.name "YOUR NAME"  
$ git config --global user.email "YOUR EMAIL ADDRESS"
```

Exemplo:

```
Barth@DESKTOP-2OLVI6S MINGW64 ~  
$ git config --global user.name "wagnerbarth"  
  
Barth@DESKTOP-2OLVI6S MINGW64 ~  
$ git config --global user.email "wagner@barth.com.br"
```

Para verificar se tudo deu certo use o comando “**less .gitconfig**”

Obs: Você deve estar no diretório “**home**” representado por um “**~**” (til), caso não esteja no diretório home execute o comando “**cd ~**”

```
MINGW64:/c/Users/Barth  
[user]  
    name = wagnerbarth  
    email = wagner@barth.com.br  
~  
~
```

Clonando um projeto para o seu computador.

Este processo traz para a sua máquina o projeto que está disponível no repositório, assim podemos utilizá-lo.

Para realizar o processo de clonagem utilizamos o comando:

```
git clone https://github.com/<username>/site.git
```

```
Barth@DESKTOP-2OLVI6S MINGW64 ~  
$ git clone https://github.com/wagnerbarth/aulaTP/  
Cloning into 'aulaTP'...  
warning: You appear to have cloned an empty repository.
```

Obs: Como não havia nenhum arquivo em nosso projeto a mensagem de “Warning” foi apresentada.

Subindo arquivos / diretórios para o repositório

Copie um arquivo ou um diretório contendo seus arquivos e, no diretório do seu projeto, digite o comando:

`git status`

```
Barth@DESKTOP-2OLVI6S MINGW64 ~
$ cd aulaTP/

Barth@DESKTOP-2OLVI6S MINGW64 ~/aulaTP (master)
$ git status
On branch master

No commits yet

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

    decisao-e-loop/
nothing added to commit but untracked files present (use "git add" to track)
```

Como podemos observar foi percebido que o seu diretório local possui uma pasta ou arquivo que não está no repositório, “**Untracked files**”.

Para subirmos os arquivos usamos o comando:

`git add <files>`

```
Barth@DESKTOP-2OLVI6S MINGW64 ~/aulaTP (master)
$ git add decisao-e-loop

Barth@DESKTOP-2OLVI6S MINGW64 ~/aulaTP (master)
$ git status
On branch master

No commits yet

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

    new file:  decisao-e-loop/decisao-e-loop.sln
    new file:  decisao-e-loop/decisao-e-loop.v11.suo
    new file:  decisao-e-loop/decisao-e-loop/App.config
    new file:  decisao-e-loop/decisao-e-loop/Program.cs
    new file:  decisao-e-loop/Properties/AssemblyInfo.cs
    new file:  decisao-e-loop/decisao-e-loop/bin/Debug/decisao-e-loop.exe
    new file:  decisao-e-loop/decisao-e-loop/bin/Debug/decisao-e-loop.exe.config
    new file:  decisao-e-loop/decisao-e-loop/bin/Debug/decisao-e-loop.pdb
    new file:  decisao-e-loop/decisao-e-loop/bin/Debug/decisao-e-loop.vshost.exe
    new file:  decisao-e-loop/decisao-e-loop/bin/Debug/decisao-e-loop.vshost.exe.config
    new file:  decisao-e-loop/decisao-e-loop/bin/Debug/decisao-e-loop.vshost.exe.manifest
    new file:  decisao-e-loop/decisao-e-loop/decisao-e-loop.csproj
    new file:  decisao-e-loop/decisao-e-loop/decisao-e-loop.csproj.user
    new file:  decisao-e-loop/decisao-e-loop/obj/Debug/DesignTimeResolveAssemblyReferencesInput.cache
    new file:  decisao-e-loop/decisao-e-loop/obj/Debug/TemporaryGeneratedFile_036C085B-1481-4323-8D20-8F5ADC23D92.cs
    new file:  decisao-e-loop/decisao-e-loop/obj/Debug/TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dd1.cs
    new file:  decisao-e-loop/decisao-e-loop/obj/Debug/TemporaryGeneratedFile_E7A71F73-0F8D-489B-B56E-8E70B10BC5D3.cs
    new file:  decisao-e-loop/decisao-e-loop/obj/Debug/decisao-e-loop.csproj.FileListAbsolute.txt
    new file:  decisao-e-loop/decisao-e-loop/obj/Debug/decisao-e-loop.exe
    new file:  decisao-e-loop/decisao-e-loop/obj/Debug/decisao-e-loop.pdb
```

Podemos observar que novos arquivos foram adicionados ao repositório, porém, eles ainda não estão por definitivo, este processo apenas adicionou os arquivos a um “index” do Git.

Commit (adicionar ao repositório)

Ao aplicarmos um “**commit**” os arquivos serão adicionados ao repositório local e ao Git.

Devemos sempre informar o que está sendo feito neste “**commit**” pois estas informações serão úteis para levantarmos os dados necessários sobre as versões. Para realizar um “**commit**” utilize o comando:

```
git commit -m "Comentários...."
```

```
Barth@DESKTOP-2OLVI6S MINGW64 ~/aulaTP (master)
$ git commit -m "Inicio do projeto -decisão-e-loop- primeira versão 1.0"
[master (root-commit) 66dfb48] Inicio do projeto -decisão-e-loop- primeira versão 1.0
 20 files changed, 351 insertions(+)
 create mode 100644 decisao-e-loop/decisao-e-loop.sln
 create mode 100644 decisao-e-loop/decisao-e-loop.v11.suo
 create mode 100644 decisao-e-loop/decisao-e-loop/App.config
 create mode 100644 decisao-e-loop/decisao-e-loop/Program.cs
 create mode 100644 decisao-e-loop/decisao-e-loop/Properties/AssemblyInfo.cs
 create mode 100644 decisao-e-loop/decisao-e-loop/bin/Debug/decisao-e-loop.exe
 create mode 100644 decisao-e-loop/decisao-e-loop/bin/Debug/decisao-e-loop.exe.config
 create mode 100644 decisao-e-loop/decisao-e-loop/bin/Debug/decisao-e-loop.pdb
 create mode 100644 decisao-e-loop/decisao-e-loop/bin/Debug/decisao-e-loop.vshost.exe
 create mode 100644 decisao-e-loop/decisao-e-loop/bin/Debug/decisao-e-loop.vshost.exe.config
 create mode 100644 decisao-e-loop/decisao-e-loop/bin/Debug/decisao-e-loop.vshost.exe.manifest
 create mode 100644 decisao-e-loop/decisao-e-loop/decisao-e-loop.csproj
 create mode 100644 decisao-e-loop/decisao-e-loop/decisao-e-loop.csproj.user
 create mode 100644 decisao-e-loop/decisao-e-loop/obj/Debug/DesignTimeResolveAssemblyReferencesInput.cache
 create mode 100644 decisao-e-loop/decisao-e-loop/obj/Debug/TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs
 create mode 100644 decisao-e-loop/decisao-e-loop/obj/Debug/TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dd1.cs
 create mode 100644 decisao-e-loop/decisao-e-loop/obj/Debug/TemporaryGeneratedFile_E7A71F73-0F8D-489B-B56E-8E70B10BC5D3.cs
 create mode 100644 decisao-e-loop/decisao-e-loop/obj/Debug/decisao-e-loop.csproj.FileListAbsolute.txt
 create mode 100644 decisao-e-loop/decisao-e-loop/obj/Debug/decisao-e-loop.exe
 create mode 100644 decisao-e-loop/decisao-e-loop/obj/Debug/decisao-e-loop.pdb
```

Se utilizarmos o comando “**git status**” poderemos ver que não há mais diferenças sobre arquivos que não estão presentes ao índice do Git, ou seja, as mudanças até agora são locais e ainda não foram publicadas no GitHub.

```
Barth@DESKTOP-2OLVI6S MINGW64 ~/aulaTP (master)
$ git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

nothing to commit, working tree clean
```

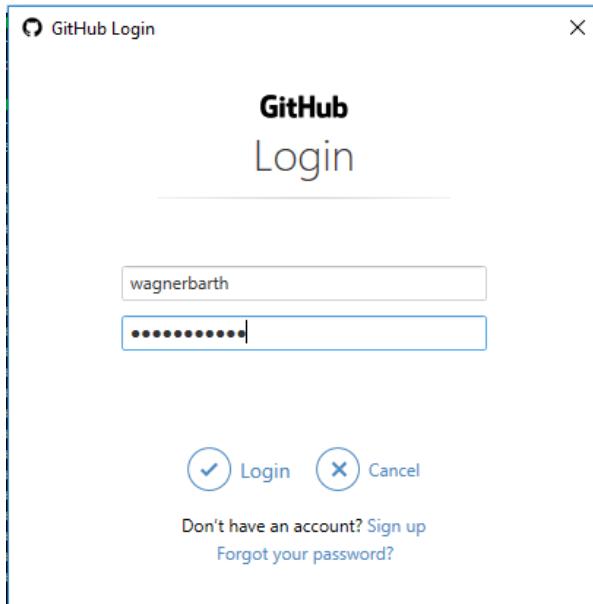
Publicando o projeto no GitHub

Para publicarmos o projeto no GitHub e deixarmos disponível para que todos os usuários utilizem o projeto devemos utilizar o comando:

```
git push
```

```
Barth@DESKTOP-2OLVI6S MINGW64 ~/aulaTP (master)
$ git push
```

Será solicitado o usuário e senha para que o processo possa ser realizado.



Após o processo de “push” serão apresentadas as informações de carregamento (publicação) no GitHub.

```
Barth@DESKTOP-2OLVI6S MINGW64 ~/aulaTP (master)
$ git push
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 4 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (23/23), 26.49 KiB | 3.78 MiB/s, done.
Total 23 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/wagnerbarth/aulaTP/pull/new/master
remote:
To https://github.com/wagnerbarth/aulaTP/
 * [new branch]      master -> master
```

Agora ao acessarmos novamente o site do GitHub com a sua conta podemos observar o projeto disponível para download e utilização.

The screenshot shows a GitHub repository page for the user 'wagnerbarth' with the repository name 'aulaTP'. The page includes navigation links for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, and Settings. At the top right, there are Watch (0), Star (0), Fork (0) buttons. Below the header, the repository name 'Aula de uso de GitHub' is displayed, along with an 'Edit' button and a 'Manage topics' link. A summary bar shows 1 commit, 1 branch, 0 releases, and 1 contributor. Below this, a 'Branch: master ▾' dropdown and a 'New pull request' button are visible. A green 'Clone or download' button is prominently displayed. The main content area lists a single commit by 'wagnerbarth' titled 'Inicio do projeto -decisão-e-loop- primeira versão 1.0', made 9 minutes ago. A file named 'decisao-e-loop' is also listed, also from 9 minutes ago. At the bottom, a note encourages adding a README, with a green 'Add a README' button.

Observe o botão “Clone or download” disponível ao lado direito do projeto. Neste botão podemos baixar os arquivos ou clonar o projeto para a nossa máquina.