

## SUMÁRIO

<b>1. HOSPEDAGEM DE REPOSITÓRIOS COM GITHUB .....</b>	<b>1</b>
<b>2. INSTALANDO O GIT E CRIANDO UMA CONTA NO GITHUB .....</b>	<b>1</b>
2.1. INSTALAÇÃO DO GIT .....	1
2.2. CRIANDO UMA CONTA NO GITHUB .....	1
<b>3. GITHUB NA PRÁTICA .....</b>	<b>2</b>
3.1. CONFIGURANDO O GIT PARA A COMUNICAÇÃO COM O GITHUB .....	2
3.2. CRIANDO REPOSITÓRIO .....	2
3.3. CRIANDO A CONEXÃO COM O REPOSITÓRIO REMOTO .....	3
3.3.1. ESTABELECENDO A CONEXÃO COM O COMANDO REMOTE .....	3
3.4. ESTABELECENDO A CONEXÃO COM O COMANDO CLONE .....	3
3.5. REALIZANDO PUSH .....	3
<b>4. DESCOBRINDO E AVALIANDO PROJETOS .....</b>	<b>4</b>
4.1. SOBRE ISSUES .....	5
4.2. SOBRE FORKS .....	5
4.3. SOBRE PULL REQUEST .....	5
4.3.1. VERIFICANDO E ACEITANDO PULL REQUESTS .....	5
<b>FONTE .....</b>	<b>6</b>

## 1. HOSPEDAGEM DE REPOSITÓRIOS COM GITHUB

O GitHub é uma plataforma para hospedagem e controle de versão de código que utiliza o sistema Git como base. Também serve como uma comunidade para programadores do mundo todo colaborarem em projetos pessoais e de código-aberto.

O GitHub é gratuito, exigindo apenas um cadastro para utilizar todos os seus recursos. Os projetos são hospedados em “repositórios”, que podem ter acesso público ou privado.

**Todo o gerenciamento dos projetos, como criação de novos repositórios, é feito via comandos do sistema Git.** Você também pode gerenciá-los pelo site, mas normalmente os processos são executados via linha de comando do seu computador ou servidor.

Recentemente foi lançado, também, uma [versão desktop do Github](#), em que o gerenciamento, ao invés de se dar por linha de comando, se dá por uma interface gráfica.

## 2. INSTALANDO O GIT E CRIANDO UMA CONTA NO GITHUB

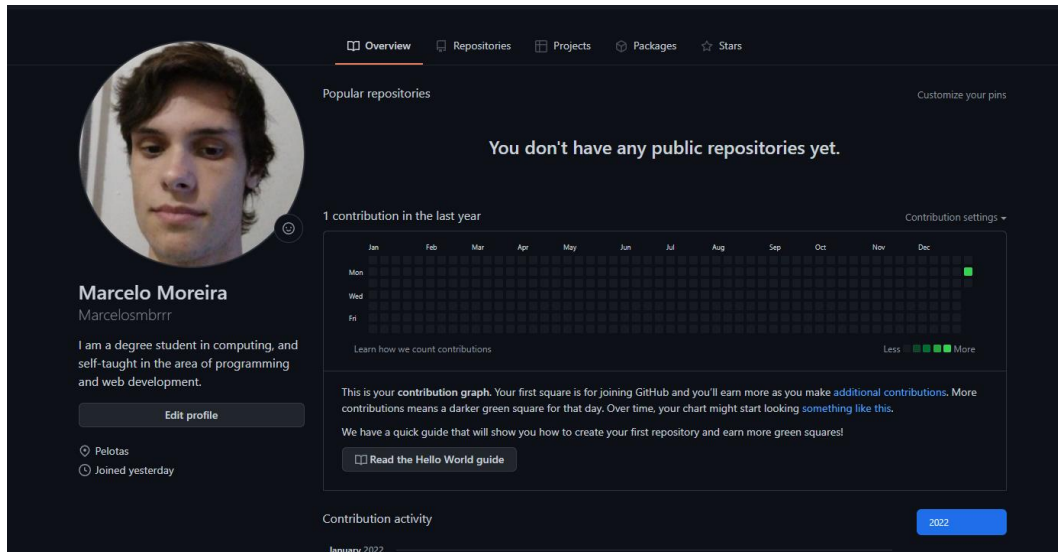
### 2.1. INSTALAÇÃO DO GIT

É preciso, então, instalar o Git em seu computador, de acordo com o seu sistema operacional. Para fazer isso acesse o site clicando [aqui](#).

### 2.2. CRIANDO UMA CONTA NO GITHUB

Para isso criar uma conta, basta acessar o site. Caso seja seu primeiro acesso, será apresentada um formulário de cadastro.

Após criada a conta, o painel principal poderá ser acessado e será igual ou semelhante ao mostrado abaixo. Um dos principais recursos do painel é a marcação de atividades realizadas na plataforma.

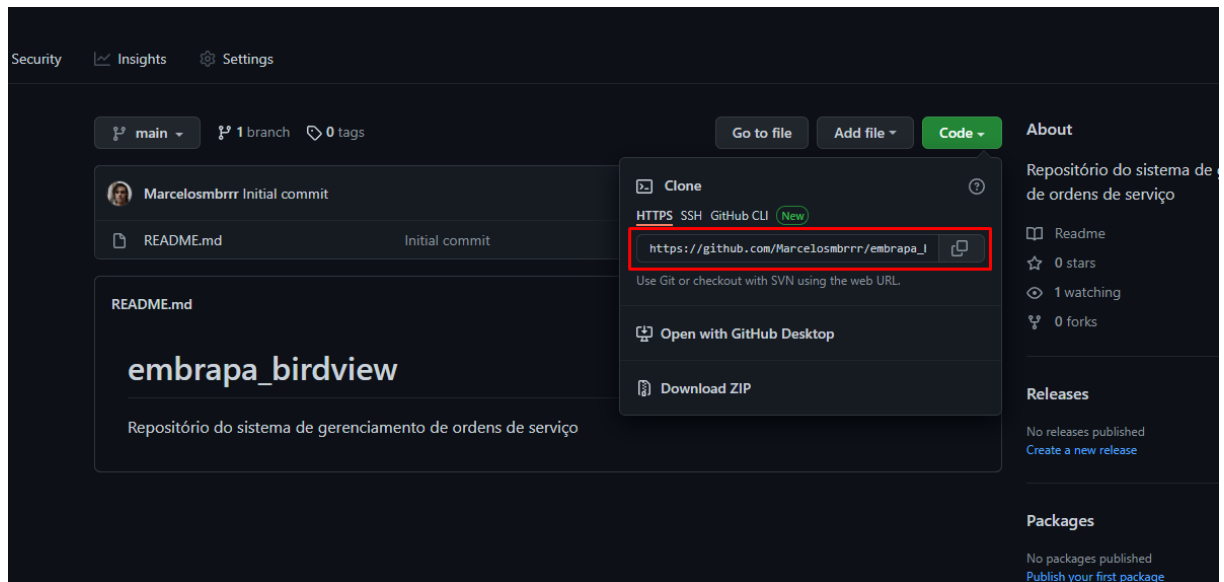


### 3. GITHUB NA PRÁTICA

#### 3.1. CONFIGURANDO O GIT PARA A COMUNICAÇÃO COM O GITHUB

#### 3.2. CRIANDO REPOSITÓRIO

Para criar um repositório, basta clicar na aba “repositórios”, e depois em “New”. O nome do repositório não deve conter acentos ou espaços, assim como são os nomes de programas, e deve ter, geralmente, um “Readme”, então a caixa para gerar um arquivo desse tipo deve ser marcada. Veja abaixo um exemplo de repositório criado:



O repositório criado é um do tipo “bare repository”, assunto que foi abordado no documento sobre o “Git”. Esse tipo de repositório serve para armazenamento, e permite que seja “empurrado” conteúdo para ele, o que faz também ser considerado um repositório “pushable”.

### 3.3. CRIANDO A CONEXÃO COM O REPOSITÓRIO REMOTO

**Existem duas formas de conectar o ambiente local com o repositório** criado na hospedagem do Github: ou o repositório criado é clonado para a máquina local, ou pode ser criado um repositório local e depois estabelecida a conexão com o repositório remoto.

Independente da forma que é escolhida, **todo repositório criado no Github tem um endereço**, como o marcado na imagem anterior, e é a partir dele que podemos obter acesso ao repositório remoto na máquina local. **O formato do endereço é:**

`https://github.com/<nome_do_usuario>/<nome_do_repositório>.git.`

Em ambos os casos, clonando o repositório, ou estabelecendo uma conexão entre um local e o remoto, é utilizado o endereço do repositório remoto.

#### 3.3.1. ESTABELECENDO A CONEXÃO COM O COMANDO REMOTE

Desta forma, um repositório local deve ser criado com o comando “git init”, e depois conectado com o remoto com o comando “git remote add <apelido> <endereço do repositório>”. Normalmente, por mera convenção, o atributo “apelido” é nomeado como “origin”.

**git remote add origin <endereço do repositório>**

### 3.4. ESTABELECENDO A CONEXÃO COM O COMANDO CLONE

No terminal do Git Bash, você irá digitar o comando abaixo, já inserindo o endereço copiado:

**git clone <endereço do repositório remoto>**

Clicando “enter”, então, o repositório remoto será clonado, e será impresso um aviso de que ele está vazio – naturalmente, pois será um recém-criado, e que será preenchido, ao longo do tempo, com comandos “push”.

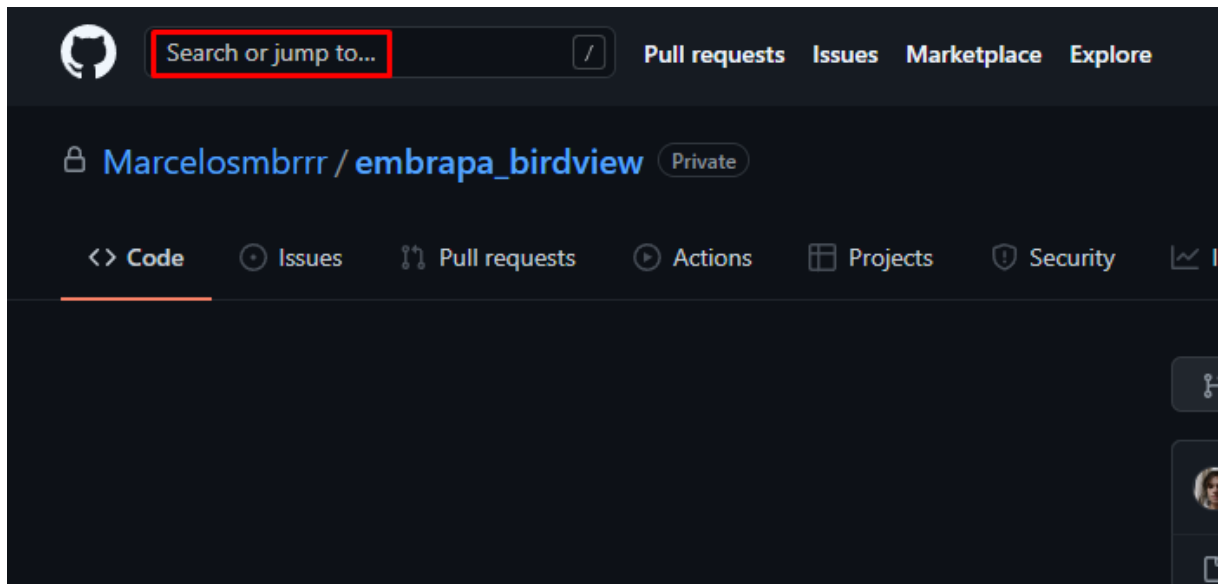
### 3.5. REALIZANDO PUSH

Para subir conteúdo para o repositório remoto, é necessário realizar o comando “git push -u <apelido> <branch>”, ou seja, um comando que empurra o conteúdo especificamente para onde, que seria o apelido do repositório, e para que branch dele.

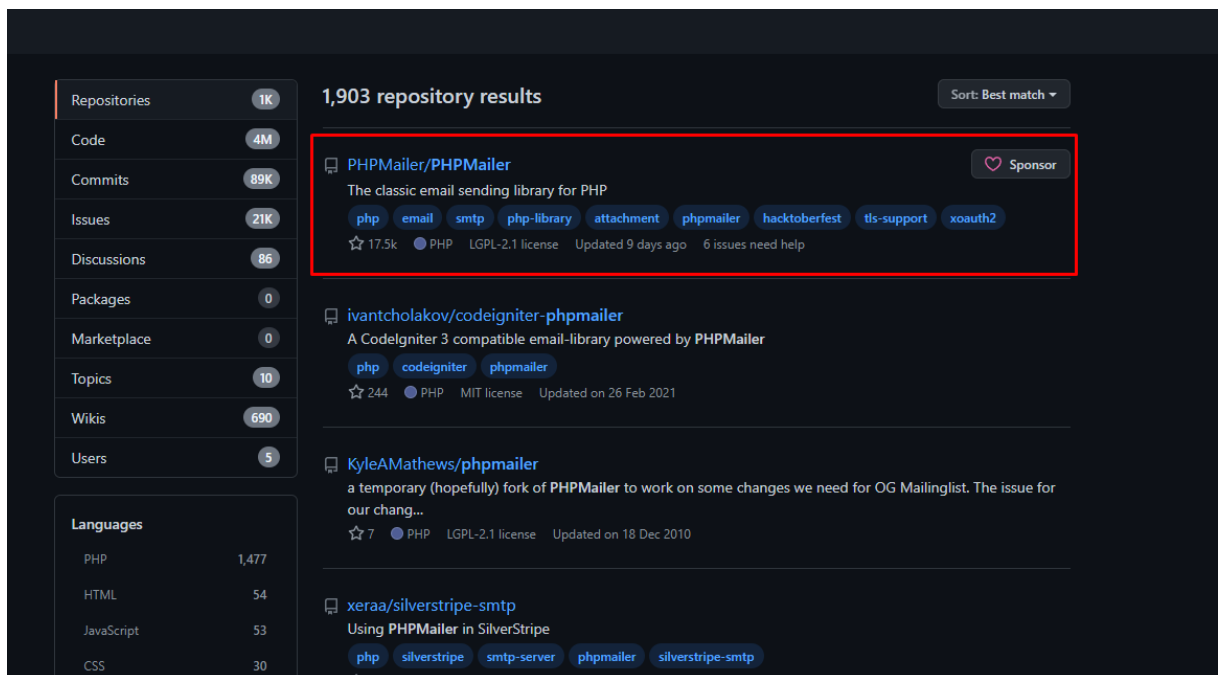
## 4. DESCOBRINDO E AVALIANDO PROJETOS

Além de ter seus próprios repositórios, de sua inteira autoria, você pode trabalhar com repositórios criados por outros usuários da plataforma.

Para isso, vá na página inicial da sua conta, clicando no ícone do Github, existente no canto superior esquerdo, e encontre o input de pesquisa “Search”.



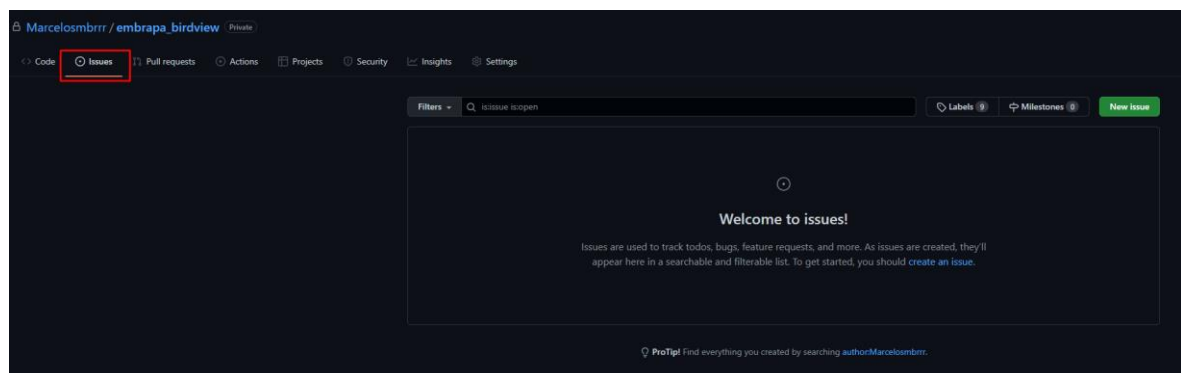
Pesquisando por “phpmailer”, por exemplo, que é uma biblioteca PHP para envio de e-mails, esse é o resultado:



#### 4.1. SOBRE ISSUES

Os problemas permitem que você acompanhe seu trabalho no GitHub e de outros colaboradores. Se trata de um formulário em que podem ser descritos erros identificados e que devem ser tratados no programa.

É uma ótima maneira para colaboração nos projetos. É possível fazer isso para um projeto próprio, ou em outros, de outros desenvolvedores. Também é possível, nessa mesma área, propor melhorias para o programa.



#### 4.2. SOBRE FORKS

Criar um “fork” de um repositório significa produzir uma cópia pessoal do projeto de outra pessoa no próprio Github. Além disso, o processo estabelece uma ponte entre o repositório original, e a cópia pessoal, permitindo que modificações sejam realizadas na cópia e inseridas no projeto original - mas claro, sob autorização do autor do projeto.

A diferença entre um fork e um clone é que o primeiro permanece no próprio Github, e passa a ser contabilizado como um dos “repositórios” do usuário. Já o clone é uma cópia de um repositório remoto para a máquina local do usuário.

#### 4.3. SOBRE PULL REQUEST

**A requisição para a assimilação no projeto original das modificações realizadas em cópias de um fork, chama-se “Pull Request”.** Ou seja, aqueles que desejam contribuir para um sistema, trabalham com suas cópias, e realizam requisições para que as alterações sejam consideradas e inseridas no projeto original.

##### 4.3.1. VERIFICANDO E ACEITANDO PULL REQUESTS

Na aba “Pull Request” é possível ver as requisições realizadas para assimilar atualizações para o repositório. Embora exista nessa seção a opção de criar um “Pull Request” mesmo para o criador do repositório, não faz sentido utilizar a funcionalidade, pois seria uma requisição de assimilação do conteúdo do criador para ele mesmo.

**FONTE**

<https://medium.com/reprogramabr/git-e-github-por-onde-come%C3%A7ar-ca88a783c223>

