

# Pequeno tutorial sobre Git

## Enviar projeto para o GitHub com Git

Acessar o diretório do projeto. As duas primeiras linhas são obrigatórias apenas na primeira vez

```
git config --global user.name "Ribamar FS"
git config --global user.email "ribafs2@gmail.com"
git init
git add .
git commit -m "Primeiro Commit"
git remote add origin git@github.com:ribafs/backup-github.git
git push -f origin master
```

## Para enviar novamente após alterações:

remova a pasta .git

e repita os passos acima

O projeto está agora hospedado no GitHub e pronto para ser abrigado pelo Packagist.

## Criar uma conta no GitHub para o SSH

Em seu desktop execute:

```
cd
ssh-keygen -t rsa -b 4096 -C "ribafs@gmail.com"
Enter
Enter
Enter
```

## Capturando o hash da senha

```
cat ~/.ssh/id_rsa.pub
```

Copie a chave mostrada desde ssh-rsa ... até seu e-mail, inclusive.

Vá até o GitHub, faça login e clique acima em seu avatar e Settings

SSH and GPG keys

New SSH key

Entre com um Title (Residência)

E cole na caixa Key abaixo

E clique em Add SSH key

## Apontar domínio para o GitHub

No repositório criar um arquivo CNAME no raiz contendo o domínio

CNAME

ribafs.org

```
ribafs.org.      3592  IN    CNAME  ribafs.github.io.
ribafs.github.io. 43192 IN    CNAME  GITHUB-PAGES-SERVER      (inválido)
GITHUB-PAGES-SERVER . 22  IN    A      192.0.2.1
```

```
ribafs.org 3600 IN A 185.199.108.153
```

```
ribafs.org 3600 IN A 185.199.110.153
```

```
ribafs.org 3600 IN A 185.199.110.153
```

```
ribafs.org 3600 IN A 185.199.111.153
```

<https://help.github.com/en/articles/managing-a-custom-domain-for-your-github-pages-site>

Na administração do domínio criar dois registros tipo A

A - 192.30.252.153

A - 192.30.252.154

CNAME www ribafs.github.io

E um tipo CNAME apontando para www e para ribafs.github.io

- Produto: Domínio Internacional (ribafs.org)
- Solicitação: Alteração de DNS
- Informe as entradas de DNS ou entradas de zona de DNS avançada desejadas: Favor apontar

A - 192.30.252.153

A - 192.30.252.154

CNAME www ribafs.github.io

- ID do produto: 24722

## Como adicionanr eu sua Github page Fork me on GitHub

<head>

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/github-fork-ribbon-css/0.2.3/gh-fork-ribbon.min.css" />
```

</head>

Na tag <body> da página:

```
<a class="github-fork-ribbon" href="https://github.com/ribafs/laravel" data-ribbon="Fork me on GitHub" title="Fork me on GitHub">Fork me on GitHub</a>
```

<https://simonwhitaker.github.io/github-fork-ribbon-css/>

## **Criando um site ou Github page no GitHub**

Exemplo: <http://ribafs.github.io>

Visitar - <https://pages.github.com/>

- Crie um repositório no GitHub chamado seuuser.github.io

Settings - Repository name - seuuser.github.io

Faça um clone do seu repositório para seu desktop

Acesse o terminal e execute:

```
git clone https://github.com/login/login.github.io
```

```
cd login.github.io
```

Crie a index.html

Agora é o momento de copiar para a pasta acima tudo que deseja em seu novo site, mas lembre que somente conteúdo frontend, HTML, CSS e JavaScript

Então envie tudo para o GitHub

```
git add --all
```

```
git commit -m "Initial commit"
```

```
git push -u origin master
```

Para criar outros sites de outros repositórios use o mesmo nome mas o endereço será, por exemplo:

<http://ribafs.github.io/laravel-acl>

## **Então pretendo enviar uma pasta para o repositório**

Tenho um repositório no GitHub com vários arquivos e pastas

Como proceder?

Exemplo

- Clonar o repositório localmente
- Acessar o terminal
- `git clone git@github.com:ribafs/tutoriais.git`
- `cd tutoriais`
- `git add .`
- `git commit 'Add nome pasta'`

- git push origin master

### **Sincronizando repositório local com o remoto**

De forma simples e quando temos apenas uma branch, a master.

Em seu desktop mantenha um repositório sincronizado com o repositório do GitHub.

```
git clone git@github.com:ribafs/teste.git teste
cd teste
# Criei um arquivo chamado local.txt apenas com uma string
git add .
git commit -m "Teste de sincronização"
git pull # Aqui avisa que já está atualizado
git push
```

Isso enviou o local.txt para o remoto

Agora eu criei um arquivo remoto.txt no repositório do GitHub e joguei uma string qualquer nele

```
git add .
git commit -m "Teste de sincronização"
git pull # Agora ele traz o arquivo remoto.txt para cá
git push # Aqui avisa que já está atualizado
```

Removendo os arquivos local e remoto do repositório local

```
git add .
git commit -m "Teste de sincronização"
git pull # Already up to date
git push # Aqui avisa que já está atualizado
```

Removeu também do remoto

### **Resumindo a Sincronização**

- Somente na primeira vez fazemos um clone do repositório remoto para uma pasta local
- Entramos na pasta e daqui pra frente para sincronizar usamos basicamente os comandos:
  - Após as alterações feitas, que podem ser locais ou remotas
  - git add .
  - git commit
  - git pull
  - git push

Beleza de dica Eduardo :)

### **Checando alterações locais**

Após efetuar alterações no repositório local podemos receber informações sobre elas usando o comando

```
git status
```

Então nos é sugerido usar git add para guardar as alterações. Geralmente usamos "git add ." para isso, mas podemos usar git add nomearquivo.

### **Checendo alterações remotas**

git pull

Estes comandos abaixo basta a primeira vez em que executou o git

```
git config --global user.name "Ribamar FS"
```

```
git config --global user.email "ribafs@gmail.com"
```

### **Boas referências**

Ótima dica recebida do colega Eduardo Alano do grupo PHP Brasil do Facebook

<https://woliveiras.com.br/posts/comandos-mais-utilizados-no-git/>

<https://dev.to/jkvyff/reset-or-sync-local-branch-to-a-remote-repository-41c4>