

# Três passos para publicar projetos React no GitHub



Com o passar dos anos, muitas ferramentas se tornaram essenciais para pessoas recrutadoras de cargos relacionados ao mundo da programação. Da mesma forma, muita coisa da parte de quem busca um emprego também mudou: quanto mais seu trabalho puder ser visto e mensurado, mais fácil você será encontrado e contratado para um determinado cargo. Para preencher esta lacuna temos diversas plataformas de hospedagem de códigos disponibilizadas por aí, sendo uma das mais conhecidas o [GitHub](#).

Se você já está familiarizado com programação, sem dúvida já deve ter algum perfil no GitHub e até mesmo algum projeto publicado por lá (e se você não tem, sinceramente, chegou a hora de correr atrás do prejuízo). Todavia, quando fazemos o upload do nosso código para a conta do GitHub, só é possível executá-lo caso realizemos alguns passos, como clonar o repositório e executar na nossa máquina local. Convenhamos que com o advento da tecnologia e da busca por praticidade, não seria possível que o GitHub só nos proporcionasse essa maneira de acessar os projetos.

Visando a facilidade na exibição das nossas aplicações, neste artigo vamos ensinar em três passos como publicar projetos no GitHub sem gastar muito tempo para fazer isto.

## Como publicar projetos no GitHub

### 1 - Instale o Github na sua máquina e crie um projeto React

Para realizarmos o upload e download de nossos projetos por meio do GitHub, é necessário instalá-lo em nosso terminal. Basta executar os comandos abaixo, caso você utilize o Linux:

```
sudo apt-get install git-all
```

```
git config --global user.name "Seu nome"
```

```
git config --global user.email "seuemail@exemplo.br"
```

Caso você use o macOS, execute o comando `brew install git` (necessário possuir o `brew` instalado). Para o sistema operacional Windows, [clique aqui](#) para ser direcionado para a página oficial de download do GitHub.

Se tudo deu certo, vamos para a próxima etapa. Ao executar os comandos abaixo, você receberá como resposta a versão do git instalada e os dados que você cadastrou acima, respectivamente:

```
git --version
```

```
git config --list
```

Para um funcionamento de excelência do GitHub em sua máquina local, também é necessário criar e cadastrar uma chave SSH. O [passo a passo](#) é disponibilizado pelo próprio GitHub para facilitar nossas vidas.

Se você ainda não criou um projeto React, execute os comandos listados abaixo (se você já o fez, vá direto para o passo 2. Caso tudo seja feito corretamente, uma página com a logo do React será carregada em seu Browser:

```
npx create-react-app publicando-meu-projeto
```

```
cd project-react-redux
```

```
npm install
```

```
npm start
```

**OBS** - Foi utilizado o *npm* como ferramenta de gerenciamento de pacotes, mas sintam-se à vontade para utilizar o que lhe for mais confortável.

## 2 - Crie um repositório no GitHub e conecte-o com seu projeto

Na tela inicial do GitHub, clique no botão verde escrito “New” localizado no canto superior esquerdo da tela. Você será encaminhado para uma página de criação de projeto, onde deverá informar o nome de criação, o responsável e se ele será público ou privado. Já que estamos publicando na plataforma com o intuito de apresentar nosso trabalho, projetos como este sempre serão públicos:

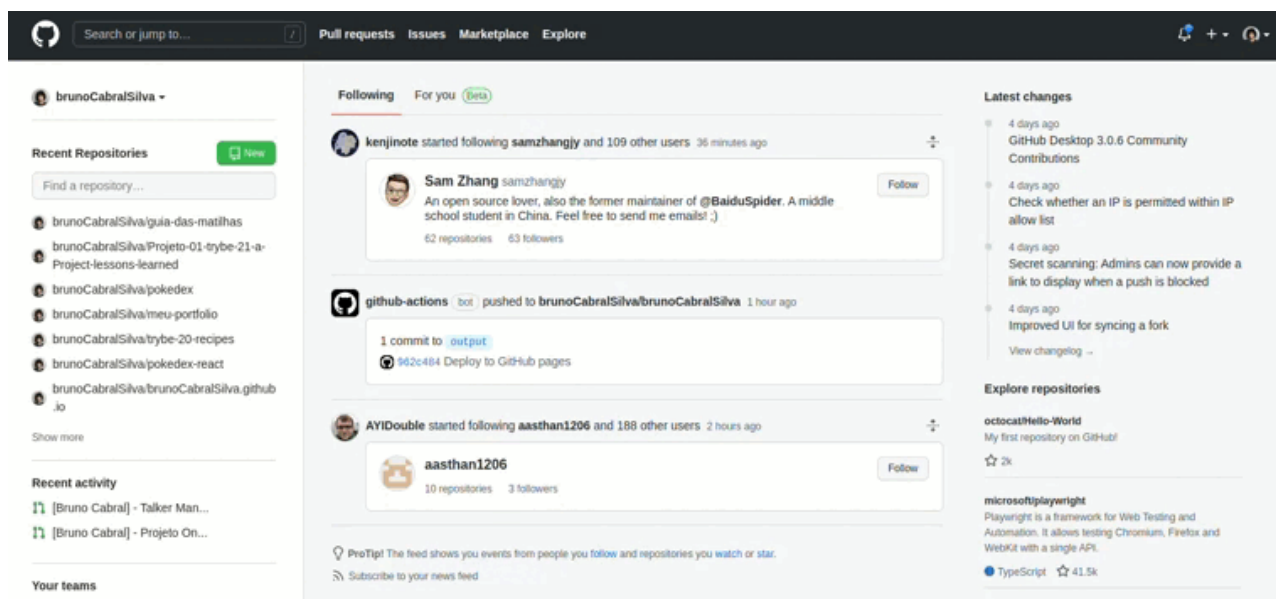


Figura 1 - Criação de Repositório GitHub

Quando criarmos o repositório, um conjunto de recomendações será apresentado em tela. Iremos utilizá-las para conectar nosso projeto local com o GitHub.

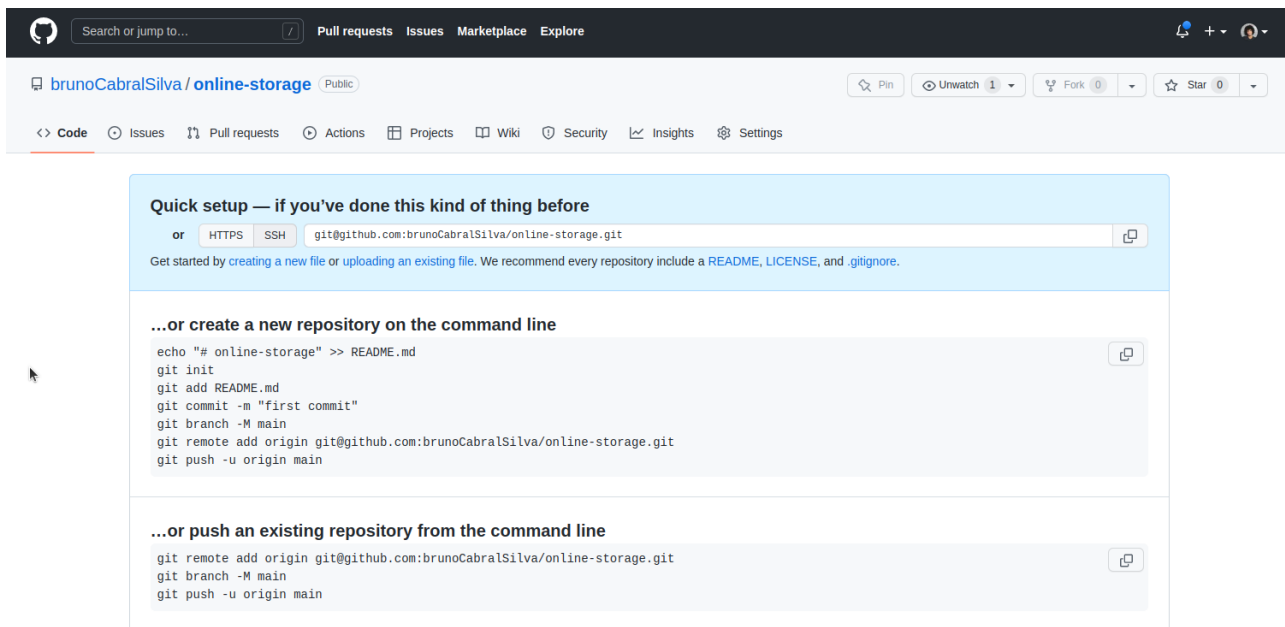


Figura 2 - Passo a passo para conexão com o repositório local

Não é necessário iniciar o git (*git init*) no nosso repositório React, pois quando executamos o comando do “*create-react-app*” já é realizada esta ação. Desta forma, execute apenas os comandos listados, lembrando da necessidade de trocar o campo “sua-chave-SSH” pela chave que o GitHub nos proporcionou ao criar o projeto:

*git add* .

*git commit -m "first commit"*

*git branch -M main*

*git remote add origin sua-chave-SSH*

*git push -u origin main*

### 3 - Instale e configure o GitHub Pages

O GitHub Pages é um serviço disponibilizado pela própria plataforma para que possamos hospedar sites que utilizam HTML, CSS e JavaScript, de forma que seja possível executá-los e publicá-los para acesso de qualquer pessoa (no caso de projetos públicos). Todas as aplicações criadas por meio do GitHub Pages são hospedadas no domínio *github.io* e o que faremos nos passos seguintes será realizar uma build para nosso repositório do GitHub em uma branch chamada *gh-pages*, que é reconhecida automaticamente pela plataforma como uma página web do repositório.

Para isto, digite o seguinte comando por meio do terminal na pasta raiz do seu projeto:

*npm add -D gh-pages*

Uma vez instalado o *gh pages*, iremos criar o script para que possamos realizar o upload do nosso conteúdo para esta branch *gh-pages*. Faremos isto adicionando algumas sentenças ao documento *package.json*, localizado na pasta raiz da aplicação criada. Neste documento *JSON*, adicionaremos uma nova chave chamada “homepage”, que terá como valor uma string contendo “*https://nome-da-conta-github.github.io/nome-do-projeto*” onde o “*nome-da-conta-github*” deve ser substituído pelo nome do seu usuário GitHub e “*nome-do-projeto*” será o nome que demos ao criar o repositório na plataforma. Além disso, adicionaremos na chave *scripts* as linhas abaixo:

`"predeploy": "npm run build",`

`"deploy": "gh-pages -d build"`

```
    "react-router-dom": "5.2.0",
    "react-scripts": "3.4.1"
  },
  "homepage": "https://brunoCabralSilva.github.io/online-storage",
  "scripts": {
    "predeploy": "npm run build",
    "deploy": "gh-pages -d build",
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject",
```

Figura 3 - Inserindo a chave homepage e atualizando a chave scripts

O último passo que devemos fazer para publicar nosso site é digitar o comando abaixo:

`npm run deploy`

Se você está utilizando o gerenciamento de rotas (super recomendado para publicação e organização de projetos), é necessário alterar a rota criada principal de “/” para “/nome-do-projeto”. Isso evitará que haja algum conflito com projetos futuros publicados no GH Pages. Neste exemplo, alteramos para ‘/online-storage’.

Para acessar a publicação do seu projeto, basta apenas copiar este endereço existente na chave “homepage” e colar na URL do seu browser. Se não aparecer nada, aguarde alguns minutos para que o projeto seja copiado de forma efetiva para a branch responsável pela exibição e depois tente novamente. Se não der certo mesmo assim, releia o passo a passo para garantir que você não se esqueceu de nenhuma ação.

Sempre que realizarmos alguma alteração em nosso projeto, precisamos atualizar a branch gh-pages através do comando `npm run deploy`. Lembre-se também de manter a branch main atualizada, realizando regularmente novos commits e encaminhando as alterações para o repositório através do comando `git push`.

## Considerações finais

É de extrema importância para o programador possuir um portfólio com os seus projetos que comprovem as suas capacidades. Deixamos aqui nossas recomendações para que você publique todas as suas aplicações no GitHub e consequentemente também no GH Pages, até mesmo para que você acompanhe sua própria evolução.

Além disso, manter o perfil atualizado com seus projetos evita que você perca-os por algum problema local na sua máquina que possa ocorrer. O que você está esperando para começar?



As opiniões e comentários expressos neste artigo são de propriedade exclusiva de seu autor e não representam necessariamente o ponto de vista da Revelo.

A **Revelo Content Network** acolhe todas as raças, etnias, nacionalidades, credos, gêneros, orientações, pontos de vista e ideologias, desde que promovam diversidade, equidade, inclusão e crescimento na carreira dos profissionais de tecnologia.

