

TUTORIAL: GITHUB PARA SUBIR PROJETOS

DevOps básico

GitHub para subir projetos

O presente documento tem por objetivo lhe apresentar a um passo a passo para a criação de projetos e o seu salvamento em um provedor de serviço de versionamento. No nosso caso, usaremos o Git como ferramenta de versionamento e o GitHub como provedor de serviço na nuvem.

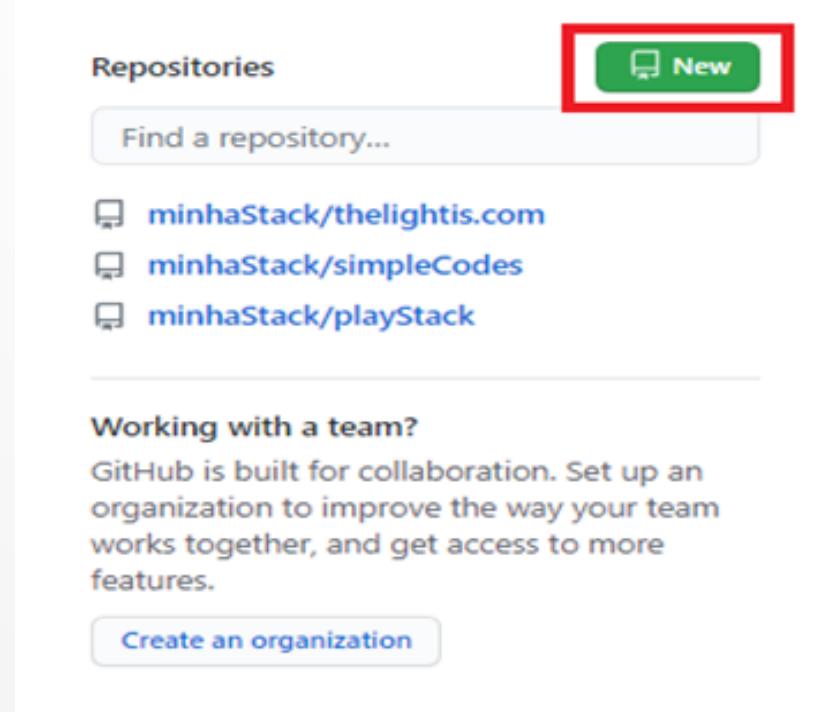
Para fazer uso da ferramenta Git, é importante que ela esteja instalada na sua máquina. Caso você ainda não tenha instalado, neste [link](#) você pode encontrar as versões para o seu sistema operacional.

Além da ferramenta instalada, é importante que você já tenha criado uma conta no GitHub. Para isso, acesse este [link](#) e se inscreva. O serviço é gratuito.

De posse da ferramenta e da inscrição no GitHub, siga o conjunto de passos que serão apresentados a seguir.

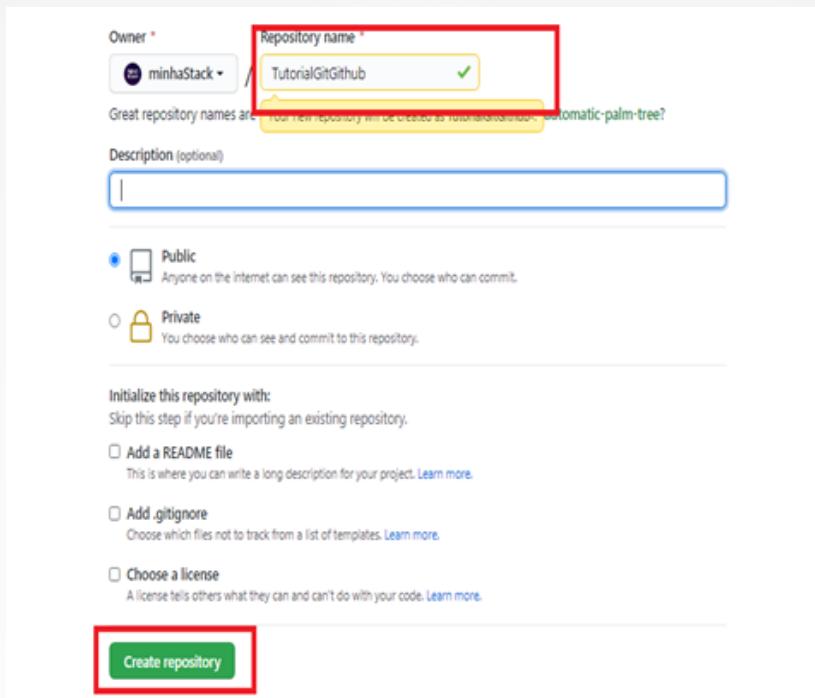
Crie seu repositório no GitHub

Faça o login em sua conta do GitHub. Uma vez logado, no lado esquerdo da página principal, deverá aparecer a sua lista de repositórios, conforme a figura abaixo.



Caso o pedaço de tela referido na figura não esteja aparecendo, confirme se você está logado. Se estiver logado, clique na logo do GitHub, no canto superior esquerdo da tela.

Ao clicar em New, é apresentada a tela da figura abaixo. O campo **repository name** é obrigatório e será posteriormente utilizado como link no GitHub para ter acesso direto ao código lá versionado/armazenado.



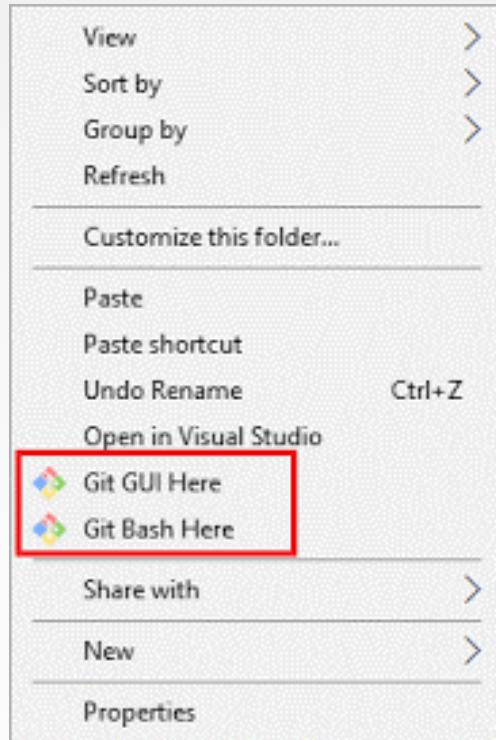
De forma básica e direta, recomenda-se a criação de um nome de repositório (**repository name**) curto e sugestivo. Defina seu projeto como público. Todas as demais opções podem ser melhor exploradas com o tempo. Lembrando que, neste tutorial, o objetivo é rapidamente a criação de um repositório e o armazenamento do seu projeto no GitHub.

Ao clicar em **create repository**, seu espaço de armazenamento na nuvem está garantido, porém ele ainda está vazio. Para preenchê-lo, vamos criar um projeto Git na sua máquina local.

Iniciando o Git no seu computador

Como ponto de partida, escolha uma pasta na qual armazenará o seu projeto em seu computador. Lembre que ele não precisa ser salvo em uma pasta que já esteja com o serviço de nuvem embutido (e.g. Dropbox, OneDrive, Google Drive etc.), pois todo o código será salvo no GitHub. Pessoalmente, possuo uma pasta “Projetos”, na qual salvo os diferentes projetos que desenvolvo. Assim, no ambiente Windows tenho a pasta C:\Projetos e, no Linux, tenho a pasta dentro da HOME do meu usuário, no subdiretório Projetos. No restante deste tutorial, consideraremos o Windows como ambiente de desenvolvimento.

Dentro da pasta escolhida, clique com o botão direito do mouse. O menu de opções aparecerá, conforme a próxima figura.



A partir do menu, clique em **git bash here**. Um terminal equivalente ao da figura abaixo aparecerá. Dentro deste terminal, digite **git init** e pressione ENTER no teclado. Ao realizar este passo, você terá criado um projeto Git localmente, mas ele ainda não estará vinculado ao serviço do GitHub. Esta vinculação será vista mais adiante.

```
jose@Neeto MINGW64 ~/Desktop/fictionalProject
$ git init
Initialized empty Git repository in C:/Users/jose/Desktop/fictionalProject/.git/
jose@Neeto MINGW64 ~/Desktop/fictionalProject (master)
$ |
```

Incluindo arquivos em nosso repositório local

Faça o commit no projeto. Para isso, digite **git commit -m “...”**. Dentro destas aspas, vamos colocar qualquer mensagem. Recomenda-se que a mensagem do comando commit contenha detalhes sobre a modificação realizada.

```
jose@Neeto MINGW64 ~/Desktop/fictionalProject (master)
$ git commit -m "envio do arquivo .txt"
```

A seguir, informamos ao Git os arquivos que serão adicionados. Para isso, adicione um novo arquivo ao repositório. Pode ser um arquivo de teste, que contenha um texto qualquer. Após criar este arquivo, digite o comando **git add nomeDoArquivo** no diretório em que o arquivo está, ou adicione todos os arquivos da sua pasta, com o comando **git add .**, no qual o símbolo **.** (ponto) representa a inclusão de todos os arquivos para serem salvos no repositório local. A Figura 6 ilustra o uso daquele comando.

```
jose@Neeto MINGW64 ~/Desktop/fictionalProject (master)
$ git add .
```

Verificando se os arquivos foram adicionados

Após adicionar um arquivo, verifique se está tudo certo com o comando **git status**. O resultado do uso deste comando pode ser observado na figura a seguir.

```
jose@Neeto MINGW64 ~/Desktop/fictionalProject (master)
$ git status
On branch master

No commits yet

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

Os recursos listados em verde demonstram que foram “adicionados” para o versionamento. Recursos que estiverem na cor vermelha não farão parte do versionamento. Caso queria incluir um recurso para versionamento, mas este está em vermelho, lembre-se de revisar o comando **git add**, garantindo que o símbolo de ponto foi usado (isto incluirá todos recursos elaborados) ou que o nome de cada recurso foi informado corretamente logo após o comando **git add**.

Após a inclusão dos arquivos para versionamento, utilize o comando **git commit -m “mensagem apropriada”** novamente.

Adequando no nome da branch

Quando criamos um repositório, alguns nomes distintos podem ser atribuídos. Para garantir uniformidade com o repositório remoto, deve-se renomear a branch usando o comando **git branch -M main**. Ao final, tecle ENTER. A figura abaixo demonstra esta ação.

```
jose@Neeto MINGW64 ~/Desktop/fictionalProject (master)
$ git branch -M main
```

Vinculando com o repositório no GitHub

Para este passo, você precisa saber o link do seu repositório no GitHub. A figura abaixo apresenta os detalhes de um repositório que foi criado. Acesse o GitHub e o seu repositório alvo. Copie o link do seu repositório para referenciar onde salvará os arquivos no GitHub. Esse vínculo será feito usando o comando **git remote add origin <<link>>**, sendo que o link pode ser melhor entendido na segunda figura abaixo.



```
jose@Neeto MINGW64 ~/Desktop/fictionalProject (main)
$ git remote add origin git@github.com:minhaStack/TutorialGitGithub-.git
```

Enviando arquivo para o GitHub

Uma vez criado o repositório local e feito o vínculo com o projeto remoto, cabe enviá-los para o GitHub. Os comandos **git add** e **git commit** afetam tão somente o repositório local. Estes comandos são importantes para preparar para envio ao GitHub. Para enviar ao GitHub, deve-se utilizar o comando **git push -u origin main**. O comando é demonstrado na figura abaixo.

```
jose@Neeto MINGW64 ~/Desktop/fictionalProject (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 249 bytes | 83.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:minhaStack/TutorialGitGithub-.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

