




Tutorial GitHub 1.0



12/2023

Índice

1 Repositório local e remoto	3
2 Git 	4
2.1 Conceito	4
2.2 Características do GIT	4
2.3 Alguns termos do Git	4
2.4 Download e instalação do Git	5
2.5 Verificar se o Git está instalado (versão)	5
2.6 Comandos do Git	6
2.6.1 Criar o repositório local com o Git (init)	6
2.6.2 Verificar o status dos arquivos que estão no repositório local	6
2.6.3 Preparar para adicionar no repositório local ('add')	7
2.6.4 Adicionar arquivo no repositório local ('commit')	7
2.6.5 Adicionar arquivo no repositório remoto com o Git (push)	7
3 GitHub 	8
3.1 Alguns termos do GitHub	8
3.2 Criar a conta no GitHub	9
3.3 Criar repositório remoto no GitHub	9
3.4 Visualizar o repositório remoto no GitHub	10
4 Github Desktop 	10
4.1 O que é o GitHub Desktop	10
4.2 Download e instalação do GitHub Desktop	10
4.3 Criar repositório local com GitHub Desktop	11
4.4 Criar repositório remoto com GitHub Desktop	11
4.5 Adicionar arquivos no repositório local com GitHub Desktop (Commit)	11
4.6 Adicionar arquivos no repositório remoto com GitHub Desktop (Push origin)	12
5 Excluir repositórios local e remoto	12
5.1 Excluir repositório local com o GitHub Desktop	12
5.2 Excluir repositório remoto no GitHub	13
5.3 Cuidado ao excluir	13
6 Mostrar seus sites para outras pessoas	14
6.1 GitHub Pages	14

Há duas formas de trabalhar com o GitHub.

- ✓ Usar o **Git** (por linha de comando no terminal) e o **GitHub**; ou
- ✓ Usar o **GitHub Desktop** (interface gráfica) e o **GitHub**.

Passo a passo para utilizar o GitHub Escolher uma das duas opções abaixo			
1ª GitHub com Git (linha de comando no terminal do VS Code)		2ª GitHub com GitHub Desktop (interface gráfica)	
1	Criar conta no Github, caso ainda não tenha (3.2)	1	Criar conta no Github, caso ainda não tenha (3.2)
2	Fazer download e instalar o Git, caso não tenha feito (2.4)	2	Fazer download e instalar o GitHub Desktop, caso não tenha feito (4.2)
3	Criar pasta para guardar o repositório local no Explorador de arquivos do seu computador, onde ficarão seus arquivos, projeto ou site.	3	Criar pasta para guardar o repositório local no Explorador de arquivos do seu computador, onde ficarão seus arquivos, projeto ou site.
4	Abrir o VS Code	4	Abrir o GitHub Desktop.
5	No VS Code, abrir a pasta criada anteriormente no seu computador.	5	Criar repositório local com GitHub Desktop (4.3)
6	Abrir o terminal do VS Code	6	Criar repositório remoto com GitHub Desktop (4.4)
7	Criar repositório local (2.6.1)	7	Adicionar arquivos no repositório local com GitHub Desktop (4.5)
8	Verificar status (2.6.2)	8	Adicionar arquivos no repositório remoto com GitHub Desktop (4.6)
9	Preparar para adicionar arquivo no repositório local (2.6.3 e 2.6.2)		
10	Adicionar arquivo no repositório local (2.6.4 e 2.6.2)		
11	Criar repositório remoto no GitHub (3.3)		
12	Adicionar arquivo no repositório remoto (2.6.5)		

1 Repositório local e remoto

Repositório local:

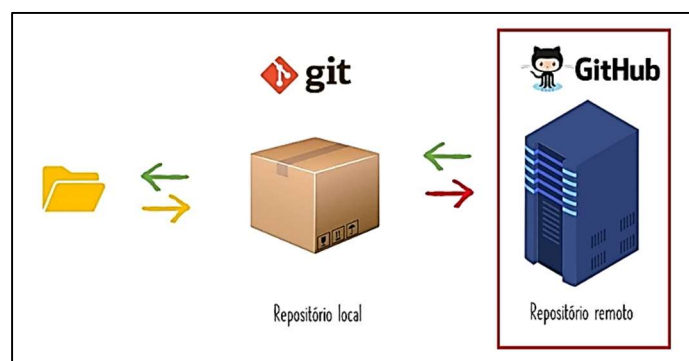
É um repositório isolado que está armazenado na sua máquina local onde você pode trabalhar em sua versão local de um projeto.

Pinto, Ivo. *Network Automation Made Easy* (Livro). Cisco Press. 2022

Repositório remoto:

“Um repositório usado para rastrear o mesmo projeto, mas que reside em outro lugar.”

<https://docs.github.com/en/get-started/quickstart/github-glossary#remote-repository>
<https://git-scm.com/docs/gitglossary#Documentation/gitglossary.txt-aiddefremoterepository>



2 Git

2.1 Conceito

"Git é um sistema de controle de versão distribuído de código aberto. Ele foi projetado para lidar com projetos de pequeno a grande porte com alta velocidade e eficiência. É desenvolvido para coordenar o trabalho entre os desenvolvedores. O controle de versão nos permite rastrear e trabalhar em conjunto com os membros da nossa equipe no mesmo espaço de trabalho."

<https://www.javatpoint.com/git>

2.2 Características do GIT

- **Código aberto (open source):** é lançado sob a licença **GPL** (General Public License).
- **Escalabilidade:** quando o número de usuários aumenta, o Git pode lidar facilmente com tais situações.
- **Segurança:** ele usa **SHA1** (Secure Hash Function) para nomear e identificar objetos em seu repositório. Ele armazena seu histórico de tal forma que o ID de commits específicos depende do histórico completo de desenvolvimento que leva a esse commit. Uma vez publicado, não é possível fazer alterações em sua versão antiga.
- **Velocidade:** ele conclui as tarefas em pouco tempo. A maioria das operações git são feitas no repositório local, proporcionando uma velocidade enorme. Além disso, um sistema centralizado de controle de versão se comunica continuamente com um servidor em algum lugar.
- **Distribuído:** significa que em vez de mudar o projeto para outra máquina, podemos criar um "clone" de todo o repositório. Além disso, em vez de ter apenas um repositório central para o qual você envia alterações, cada usuário tem seu próprio repositório que contém todo o histórico de commits do projeto. Não precisamos nos conectar ao repositório remoto; a mudança é armazenada apenas em nosso repositório local. Se necessário, podemos enviar essas alterações para um repositório remoto.

<https://www.javatpoint.com/git>

2.3 Alguns termos do Git

- **SCM:** Source Code Management.
- **Commit:** um commit, ou "revisão", é uma alteração individual em um arquivo (ou conjunto de arquivos). Quando você faz um commit para salvar seu trabalho, o Git cria um ID exclusivo (também conhecido como "SHA" ou "hash") que permite manter o registro das alterações específicas confirmadas junto com quem as fez e quando. Os commits geralmente contêm uma mensagem de commit que é uma breve descrição de quais alterações foram feitas.

<https://docs.github.com/en/get-started/quickstart/github-glossary#commit>

- **Push:** enviar suas alterações confirmadas para um repositório remoto em GitHub.com. Por exemplo, se você alterar algo localmente, poderá enviar essas alterações para que outras pessoas possam acessá-las.

<https://docs.github.com/en/get-started/quickstart/github-glossary#push>



<https://www.toolsqa.com/git/git-terminologies/>

- **Pull:** quando você está buscando alterações e mesclando-as. Por exemplo, se alguém editou o arquivo remoto em que você e outra pessoa estão trabalhando, você desejará inserir essas alterações em sua cópia local para que fique atualizado.

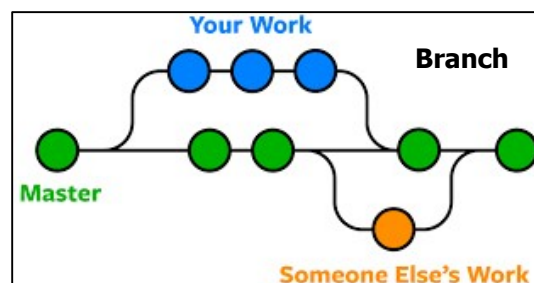
<https://docs.github.com/en/get-started/quickstart/github-glossary#pull>



- **Clone:** é uma cópia de um repositório que fica no seu computador, em vez de em algum servidor de um site, ou o ato de fazer essa cópia. Ao fazer um clone, você pode editar os arquivos em seu editor preferido e usar o Git para acompanhar suas alterações sem precisar estar online. O repositório que você clonou ainda está conectado à versão remota para que você possa enviar suas alterações locais para o remoto para mantê-las sincronizadas quando estiver online.

<https://docs.github.com/en/get-started/quickstart/github-glossary#clone>

- **Main:** o branch de desenvolvimento padrão. Sempre que você cria um repositório Git, um branch chamado main é criado e se torna o branch ativo.
<https://docs.github.com/en/get-started/quickstart/github-glossary#main>
- **Merge:** a mesclagem pega as alterações de uma ramificação/branch (no mesmo repositório ou de uma bifurcação) e as aplica em outra. Isso geralmente acontece como uma "solicitação pull" (que pode ser considerada uma solicitação de mesclagem) ou por meio da linha de comando. Uma mesclagem pode ser feita por meio de uma solicitação pull por meio da interface da web do GitHub.com, se não houver alterações conflitantes, ou sempre pode ser feita por meio da linha de comando.
<https://docs.github.com/en/get-started/quickstart/github-glossary#merge>
- **Checkout:** você pode usar git checkout na linha de comando para criar um novo branch, alterar seu branch de trabalho atual para um branch diferente ou até mesmo mudar para uma versão diferente de um arquivo de um branch diferente com `git checkout [branchname] [caminho para o arquivo]`.
<https://docs.github.com/en/get-started/quickstart/github-glossary#checkout>
- **Branch:** é uma versão paralela de um repositório. Ele está contido no repositório, mas não afeta o branch primário ou principal (main), permitindo que você trabalhe livremente sem interromper a versão "ao vivo". Depois de fazer as alterações desejadas, você pode mesclar seu branch novamente no branch principal para publicar suas alterações.
<https://docs.github.com/en/get-started/quickstart/github-glossary#branch>



<https://www.nobledesktop.com/learn/git/git-branches>

2.4 Download e instalação do Git

<https://git-scm.com/download>

<https://github.com/git-for-windows/git/releases/download/v2.43.0.windows.1/Git-2.43.0-64-bit.exe>



Após o download, executar o arquivo 'Git-2.43.0-64-bit.exe' para efetuar a **instalação**.

2.5 Verificar se o Git está instalado (versão)

No Visual Code:

- Abrir o terminal.
- Digitar `git --version`

```
PS E:\testedogit> git --version
git version 2.42.0.windows.2
```

2.6 Comandos do Git

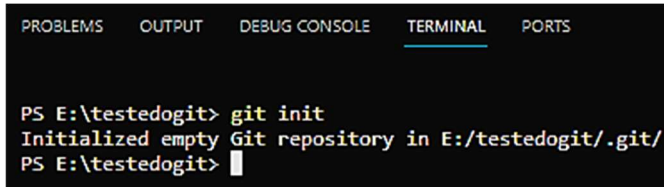
Matheus Battisti – Hora de Codar

Aprenda GIT em 30 minutos – Os principais comandos de GIT

<https://www.youtube.com/watch?v=Zwv9gRyVeU4>, acessado em 15/12/2023.

2.6.1 Criar o repositório local com o Git (init)

- Abrir o VS Code.
- Abrir a pasta onde você quer criar o repositório local.
- Abrir o terminal.
- Digitar no terminal: `git init`



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

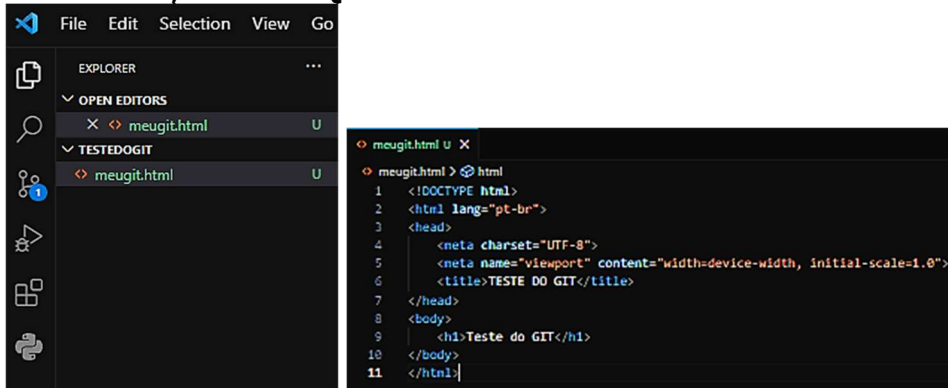
PS E:\testedogit> git init
Initialized empty Git repository in E:/testedogit/.git/
PS E:\testedogit>
```

Após o `'git init'`, abrir o Explorador de Arquivos na pasta do projeto para ver se a pasta `'.git'` (repositório local) foi criada.

Pasta do projeto e repositório local:



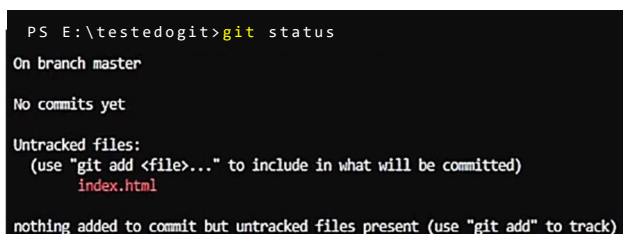
NO VS CODE, CRIAR ARQUIVO HTML



2.6.2 Verificar o status dos arquivos que estão no repositório local

- No VS Code, abrir o terminal.
- Digitar no terminal: `git status`

Se aparecer a mensagem: "Untracked files" significa que os arquivos do repositório local listados no terminal ainda não estão no repositório remoto, então é preciso adicioná-los.



```
PS E:\testedogit> git status

On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
```

2.6.3 Preparar para adicionar no repositório local ('add')

Os arquivos serão adicionados para serem rastreados pelo GIT.

Se for para adicionar somente um arquivo específico, colocar o seu nome (neste caso é 'index.html').

No VS Code, digitar no terminal: `git add .\index.html`

Se for para adicionar todo o repositório (incluir no comando o **ponto** após o 'add'):

No VS Code, digitar no terminal: `git add .`

Após executar 'git add', verificar novamente o status, para verificar as mudanças (`git status`)

```
PS E:\testedogit>git status
On branch master

No commits yet

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

2.6.4 Adicionar arquivo no repositório local ('commit')

No VS Code, digitar no terminal: `git commit -a -m "Criação do arquivo meugit.html"`

-a: para adicionar todos os tipos de arquivos ao commit.

-m: para adicionar uma mensagem dizendo o que foi feito no commit, que na imagem abaixo é "Criação do arquivo meugit.html".

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\testedogit> git commit -a -m "Criação do arquivo meugit.html"
[master (root-commit) 165e0b5] Criação do arquivo meugit.html
 1 file changed, 11 insertions(+)
 create mode 100644 meugit.html
```

Fazer novamente a verificação do status para confirmar que já foram feitas todas as adições (`git status`)

```
PS E:\testedogit> git status
On branch master
nothing to commit, working tree clean
```

CRIAR REPOSITÓRIO NO GITHUB (Ver item 3.3)

2.6.5 Adicionar arquivo no repositório remoto com o Git (push)

Faz a **ligação** entre os repositórios local e remoto.

No terminal do VS Code, digitar:

1º Adicionar uma origem remota ao repositório

`git remote add origin https://github.com/nome_da_sua_conta_no_github/nome_do_repositorio.git`

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\testedogit> git remote add origin https://github.com/carmencabral/testedogit.git
PS E:\testedogit>
```

2º Branch principal da aplicação

`git branch -M main`

```
PS E:\testedogit> git branch -M main
PS E:\testedogit>
```

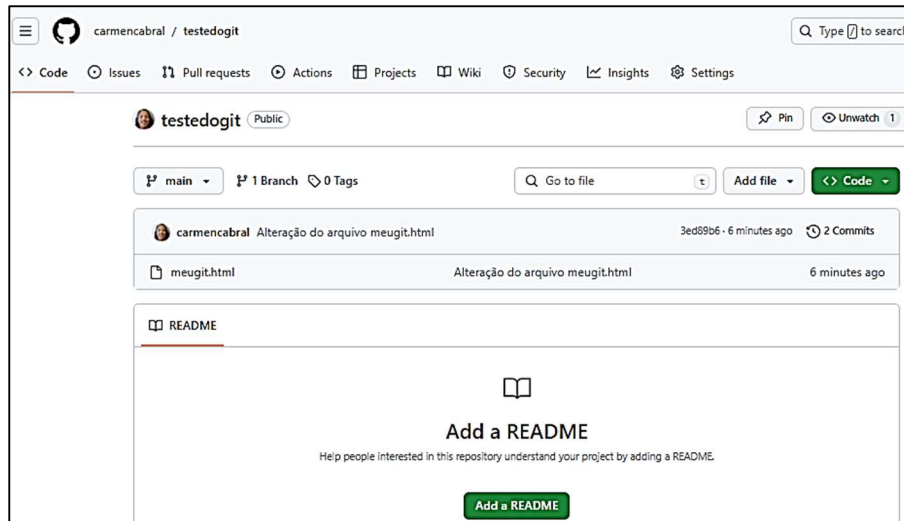
3º Enviar o arquivo para o repositório remoto

`git push -u origin main`


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\testedogit> git push -u origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 646 bytes | 215.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/carmencabral/testedogit.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS E:\testedogit>
```

Após esses comandos, ir no **GitHub** para visualizar o repositório remoto.



3 GitHub

3.1 Alguns termos do GitHub

- **Readme:** é um arquivo de texto contendo informações sobre os arquivos em um repositório que normalmente é o **primeiro arquivo** que um visitante do seu repositório verá.
<https://docs.github.com/en/get-started/quickstart/github-glossary#readme>
- **Github Pages:** é também conhecido como páginas. Um serviço de **hospedagem de site estático** projetado para hospedar páginas pessoais, de organização ou de projeto diretamente de um repositório GitHub.
<https://docs.github.com/en/get-started/quickstart/github-glossary#github-pages>
- **License:** é um **documento** que você pode incluir em seu projeto para que as pessoas saibam o que podem ou não fazer com seu código-fonte.
<https://docs.github.com/en/get-started/quickstart/github-glossary#license>
- **Repositório:** é o elemento mais básico do GitHub. Eles são mais fáceis de imaginar como uma **pasta de projeto**. Um repositório contém todos os arquivos do projeto (incluindo documentação) e armazena o **histórico de revisão** de cada arquivo. Os repositórios podem ter vários colaboradores e podem ser públicos ou privados.
<https://docs.github.com/en/get-started/quickstart/github-glossary#repository>
- **Repositório público:** pode ser visualizado por **qualquer pessoa**, incluindo pessoas que não são usuários do GitHub.
<https://docs.github.com/en/get-started/quickstart/github-glossary#public-repository>
- **Repositório privado:** são visíveis apenas para o **proprietário** do repositório e para os **colaboradores especificados** pelo proprietário.
<https://docs.github.com/en/get-started/quickstart/github-glossary#private-repository>

3.2 Criar a conta no GitHub

<https://github.com/signup>

Harish Rajora (autor); Lakshay Sharma (revisor). <https://www.toolsqa.com/git/how-to-create-github-account/>

The first screenshot shows the GitHub sign-up form with the username 'harishrajora' entered. A red error message states: 'Username harishrajora is already taken. harishrajora-ui, harishrajora-eng, or harishrajora805 are available.' The second screenshot shows the form with 'harishrajora805' as the username, 'harish@toolsqa.com' as the email, and a password. The third screenshot shows the 'What is your level of programming experience?' section with 'New to programming' selected, and the 'What do you plan to use GitHub for?' section with 'Learning Git and GitHub', 'Host a project (repository)', 'Collaborating with my team', and 'Finding a project to contribute to' selected.

3.3 Criar repositório remoto no GitHub

Pré-requisito: Criar a conta no GitHub

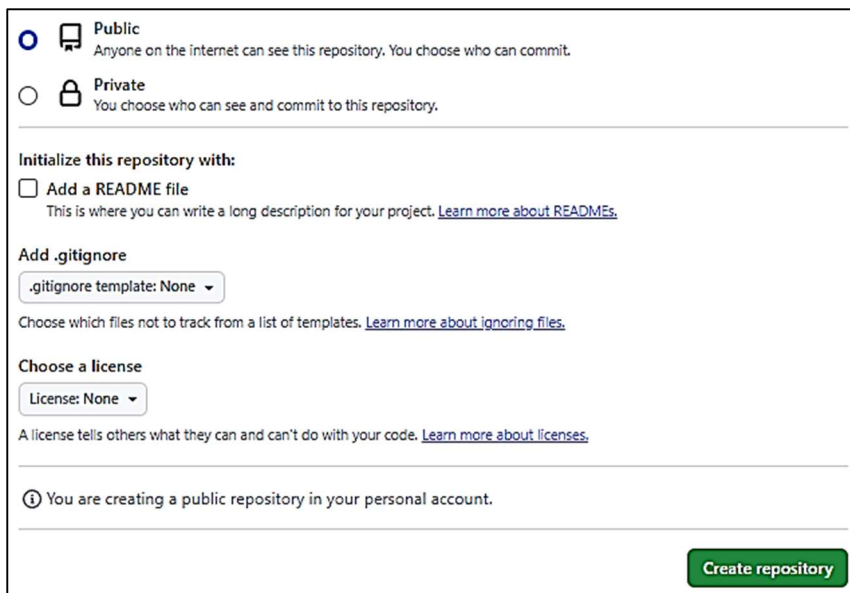
Na sua conta do GitHub, clicar no ícone '+' no canto superior direito e depois clicar em 'New repository'.

The screenshot shows the GitHub top navigation bar with a search bar and a '+ Create new...' button.

The screenshot shows the dropdown menu that appears after clicking the '+' button, with 'New repository' selected.

- Colocar o nome do repositório (**Repository name**) remoto, que neste caso é 'testedogit'.
- A descrição (**description**) é opcional, mas é aconselhável colocar para informar sobre algum detalhe.
- Deixar marcada a opção 'Public' para outras pessoas verem seus repositórios.
- Clicar no botão 'Create Repository'

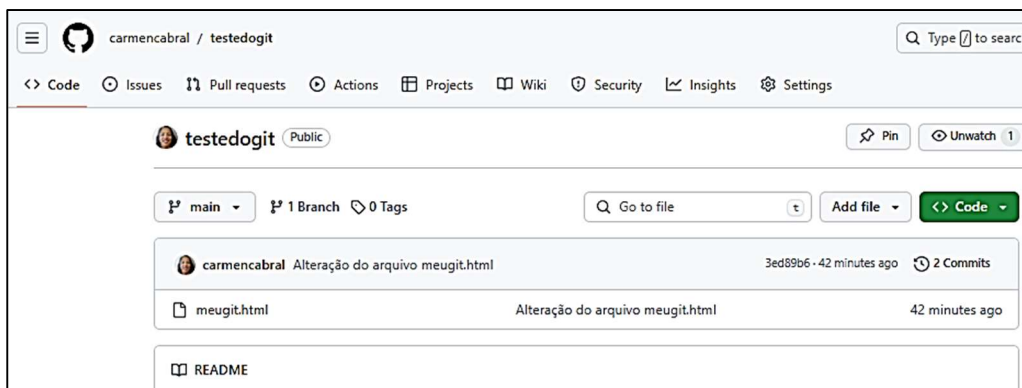
The screenshot shows the 'Create a new repository' form. The 'Owner' is 'carmencabral' and the 'Repository name' is 'testedogit'. A message states: 'Your new repository will be created as testedogit-. The repository name can only contain ASCII letters, digits, and the characters -, ., and _.' The 'Description (optional)' field is empty.



The screenshot shows the 'Create repository' form on GitHub. It has two radio buttons: 'Public' (selected) and 'Private'. Below them is a section 'Initialize this repository with:' with a checkbox 'Add a README file'. There is a dropdown for 'Add .gitignore' (set to 'None') and another for 'Choose a license' (set to 'None'). A green 'Create repository' button is at the bottom right.

3.4 Visualizar o repositório remoto no GitHub

Fazer refresh na página da sua conta do GitHub, caso já esteja nela ou abrir o GitHub com suas credenciais.



4 Github Desktop

4.1 O que é o GitHub Desktop

É um aplicativo gratuito e de código aberto que ajuda você a trabalhar com arquivos hospedados no GitHub ou em outros serviços de hospedagem Git.

<https://docs.github.com/en/desktop/overview/about-github-desktop#about-github-desktop>

4.2 Download e instalação do GitHub Desktop

<https://desktop.github.com/>

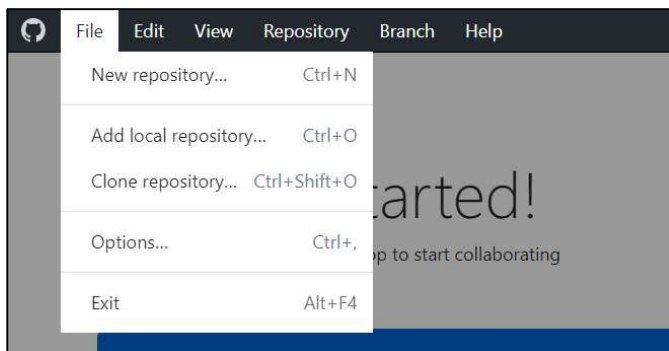
<https://central.github.com/deployments/desktop/desktop/latest/win32>

Após o **download** do arquivo 'GitHubDesktopSetup-x64.exe', fazer a sua **instalação**.



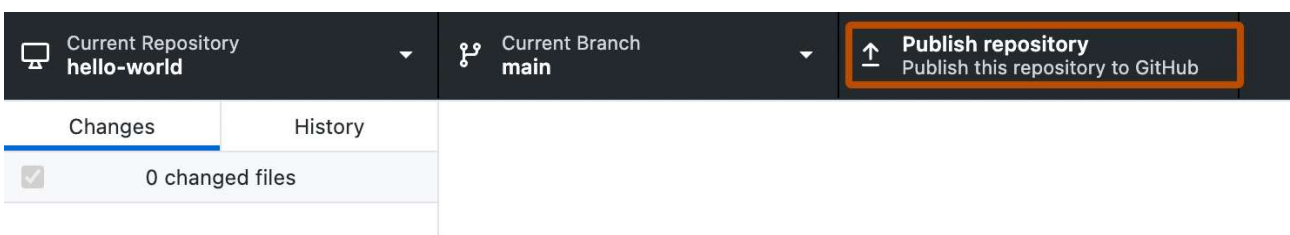
4.3 Criar repositório local com GitHub Desktop

Abrir o GitHub Desktop, clicar no menu 'File' e depois em 'New repository...'



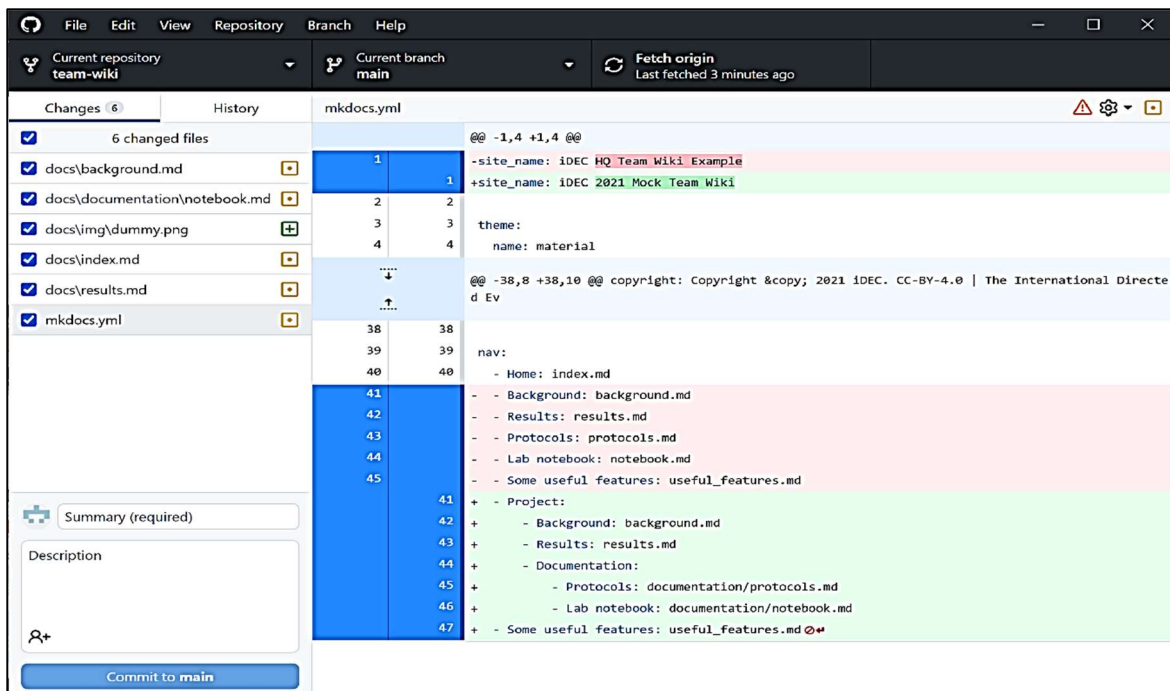
4.4 Criar repositório remoto com GitHub Desktop

Observar se o repositório atual/'Current Repository' está correto antes de continuar.



Ao clicar em '**Publish repository**', será criado o repositório remoto 'hello-world'.

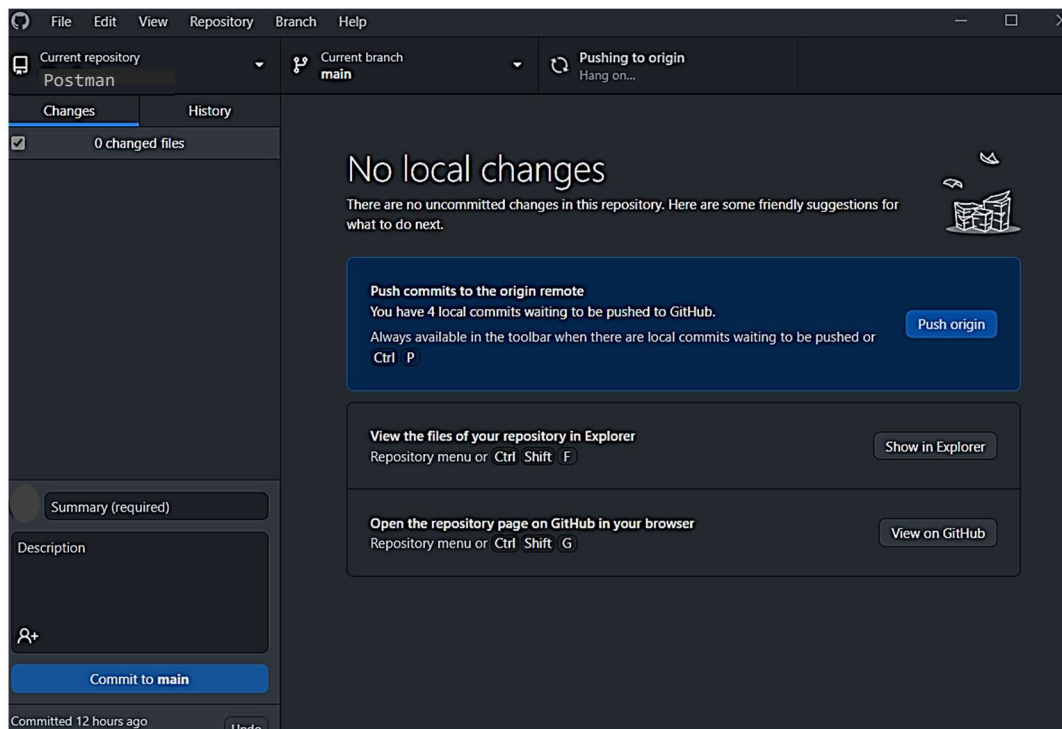
4.5 Adicionar arquivos no repositório local com GitHub Desktop (Commit)



Em 'Summary (required)', colocar uma descrição do que foi alterado ou incluído no repositório atual / Current repository, como um novo arquivo, uma alteração feita em algum arquivo. Ex. Inclusão do arquivo contatos.html; Alteração do index.html; Inclusão de exercício do Postman.

4.6 Adicionar arquivos no repositório remoto com GitHub Desktop (Push origin)

Após fazer 'Push origin', serão adicionados os arquivos no repositório remoto (GitHub).



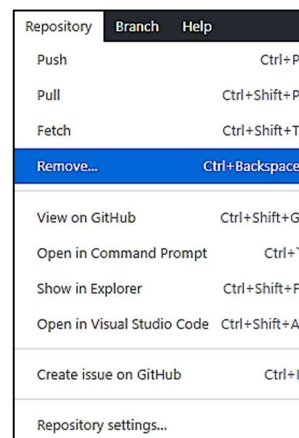
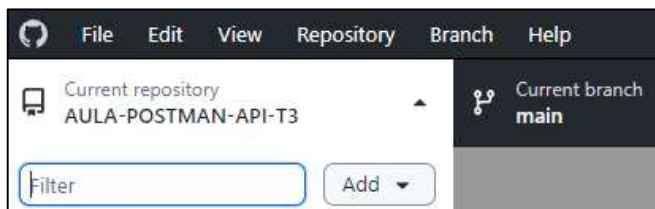
Após clicar em 'commit to main', clicar em 'Push origin' para adicionar os arquivos no repositório remoto.

5 Excluir repositórios local e remoto

Excluir somente quando tiver certeza absoluta, pois não o repositório não poderá ser recuperado. Em ambas exclusões, é necessário também fazer a exclusão manual da pasta local do repositório do seu computador, abrindo o 'Explorador de Arquivos'/'File Explorer'.

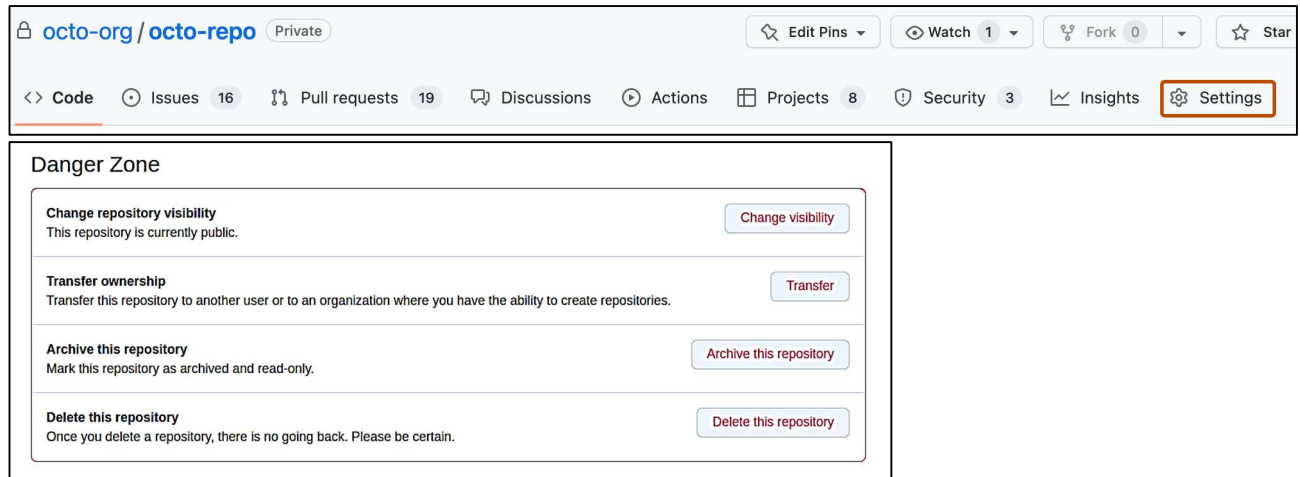
5.1 Excluir repositório local com o GitHub Desktop

- Abrir o GitHub Desktop.
- Verificar se o repositório atual/'Current repository' é o repositório a ser apagado, senão, alterar.
- Clicar no menu 'Repository'
- Clicar em 'Remove'.
- Fechar o GitHub Desktop

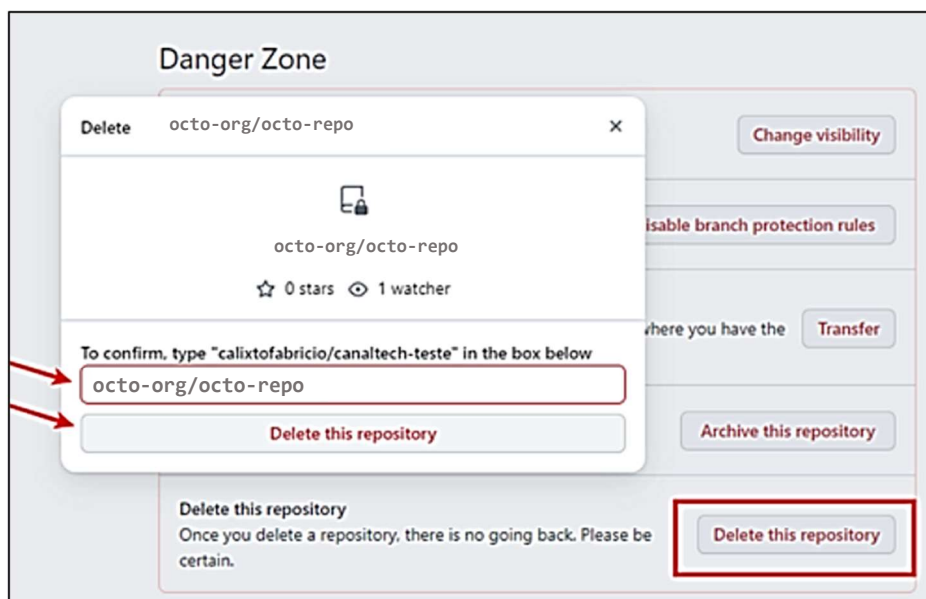


5.2 Excluir repositório remoto no GitHub

- Abrir a página do GitHub com suas credenciais válidas;
- Clicar no repositório que será excluído;
- Clicar em 'Settings/Configurações';
- Vá até a 'Danger Zone/Zona de Perigo';
- Clique em 'Delete this repository'.



<https://docs.github.com/pt/repositories/creating-and-managing-repositories/deleting-a-repository>



5.3 Cuidado ao excluir

Are you ABSOLUTELY sure?

Unexpected bad things will happen if you don't read this!

This action **CANNOT** be undone. This will delete the repository, wiki, issues, and comments permanently.

Please type in the name of the repository to confirm.

I understand the consequences, delete this repository

Você tem certeza ABSOLUTA?

Coisas ruins inesperadas acontecerão se você não ler isto!

Esta ação **NÃO PODE** ser desfeita. Isto apagará o repositório, *wiki*, *issues* e *comments* permanentemente.

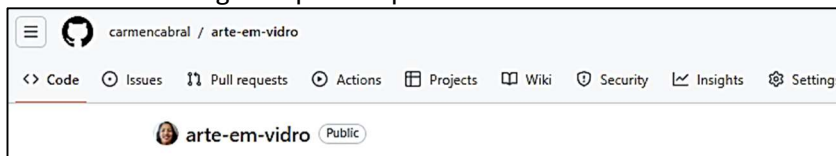
Por favor digite o nome do repositório para confirmar.

Eu entendo as consequências, apague este repositório.

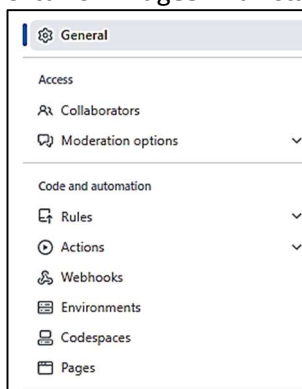
6 Mostrar seus sites para outras pessoas

6.1 GitHub Pages

- No Explorador de arquivos, criar a pasta onde ficaram os arquivos do site.
- No VS Code, criar todos arquivos do site (arquivos HTML, CSS, JavaScript) na pasta criada anteriormente, incluindo também o arquivo `index.html`, que será a página inicial do site.
- Colocar nessa pasta também os arquivos de imagem, áudio e vídeo, se for o caso.
- Criar um repositório local e remoto, conforme itens 4.3 e 4.4.
- Entrar no GitHub com suas credenciais válidas.
- Entrar no repositório que tem o site a ser publicado. Na imagem abaixo é `'arte-em-vidro'`.
- Clicar em `'Settings'` na parte superior direita.



- Clicar em `'Pages'` na lista de opções que aparece à esquerda.



- Escolher `'main'` e depois clicar em `'Save'`.



- Após salvar, será gerado um link com `'https://'` para abrir o site. Ao clicar em `'Visit site'`, o site será aberto.

