

Exercício Prático de Git e GitHub

Atenção, se ainda não possuir o git em sua máquina, faça o download e a instalação do git no seu computador através deste [link](#)

Será de suma importância que você tenha o VSCODE, ou outro editor de código instalado, para facilitar a manipulação de arquivos, você pode instalar o VSCODE a partir deste [link](#)

Agora, é abrir o git **BASH** e o VSCODE e seguir as orientações abaixo:

Objetivo:

Aplicar os conceitos de controle de versão usando Git e GitHub, praticando a criação de repositórios, commits, branches, merges, e o uso de GitHub para colaboração.

Parte 1: Configuração Inicial

1. Configurar Git:

Verifique se o Git está instalado: ``git --version``

Configure seu nome e email (se ainda não fez isso):

```
git config --global user.name "Seu Nome"  
git config --global user.email "seuemail@dominio.com"
```

2. Criar um Repositório Local:

Crie uma pasta chamada ``meu_projeto`` e navegue até ela, **recomendo abrir o git bash pelo menu do botão direito do mouse, no gerenciador de arquivos, como demonstrado em sala de aula.**

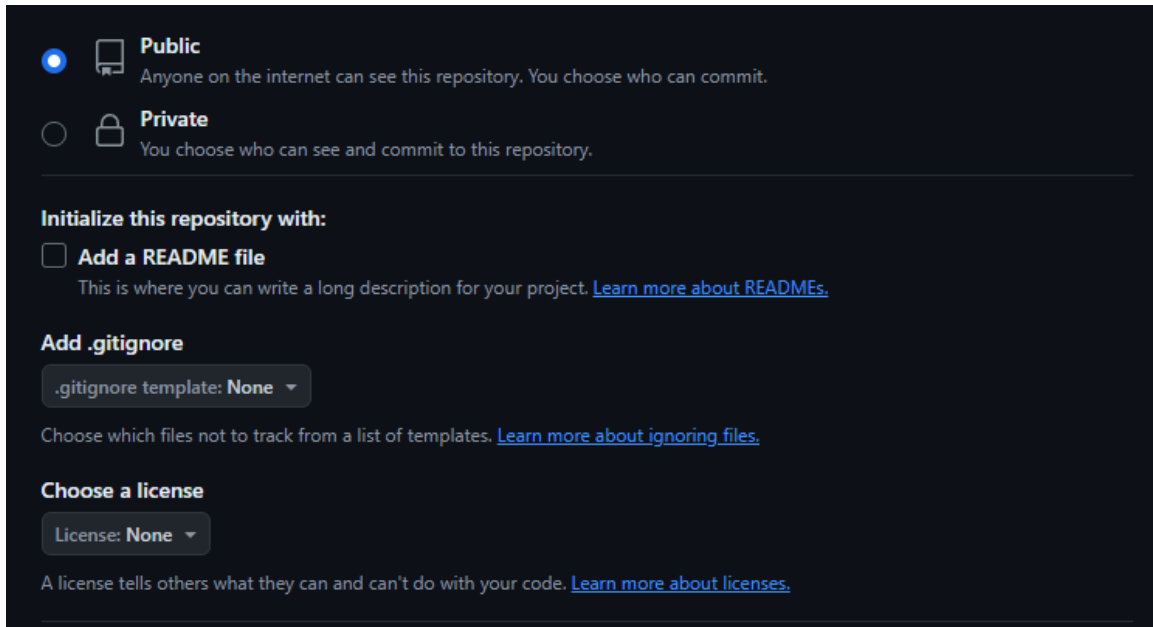
Inicialize um repositório Git:

```
git init
```

3. Criar um Repositório no GitHub:

No GitHub, crie um novo repositório chamado ``meu_projeto``. Não adicione README, .gitignore ou licença; vamos fazer isso localmente.

Você deve deixar as opções assim para criar um repositório limpo, ou seja, sem README e .gitignore



The screenshot shows the GitHub repository creation interface. At the top, there are two radio button options: 'Public' (selected) and 'Private'. Below these, there is a section 'Initialize this repository with:' containing a checkbox for 'Add a README file' (unchecked) and a dropdown for 'Add .gitignore' (set to 'None'). At the bottom, there is a 'Choose a license' section with a 'License: None' dropdown. All the options shown are the ones recommended in the text to create a 'clean' repository.

4. Conectar Repositório Local ao GitHub:

Adicione o repositório remoto:

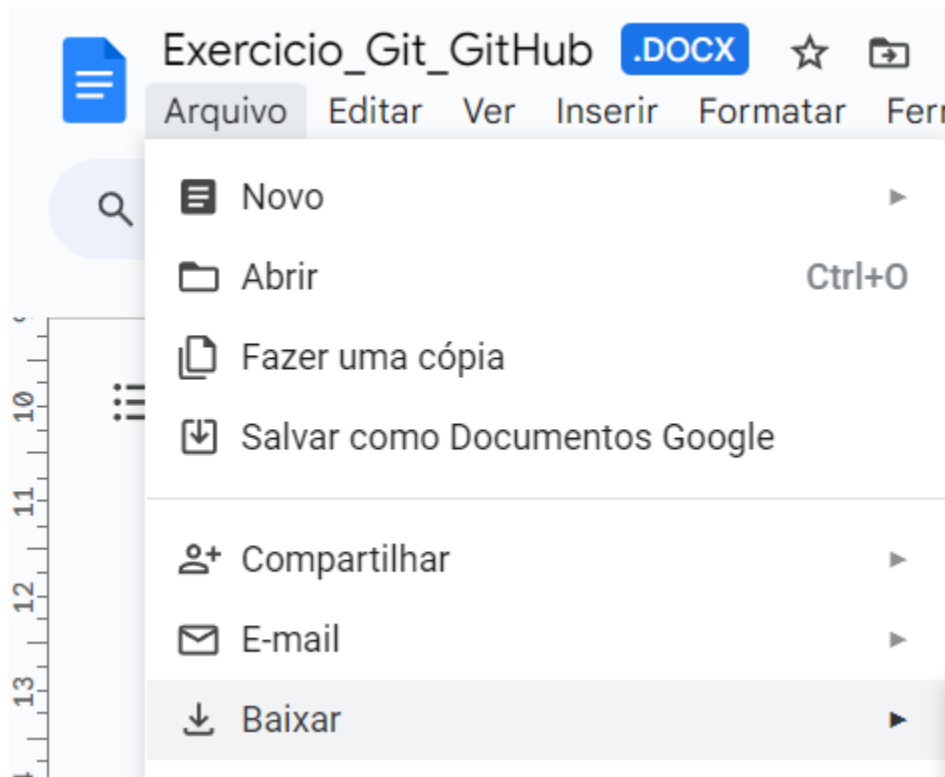
Abaixo, **em vermelho**, estão valores que deverão ser mudados no seu link, de acordo com o seu nome de usuário e o nome do repositório Remoto.

```
git remote add origin https://github.com/seuusuario/meu_projeto.git
```

Parte 2: Controle de Versão

1. Criar e Comitar Arquivos:

Baixe esse arquivo do google drive através do menu **Arquivo** -> **Baixar**.



Adicione e commite o arquivo:

```
git add Exercicio_Git_GitHub.docx  
git commit -m "Adicionar arquivo Exercicio_Git_GitHub.docx"
```

2. Criar uma Branch para uma Nova Funcionalidade:

Crie e mude para uma nova branch chamada `nova_funcionalidade`:

```
git checkout -b nova_funcionalidade
```

Adicione um arquivo `funcionalidade.txt`, e commite:

git add .

git commit -m "Adiciona novo arquivo funcionalidade.txt"

3. Fazer Merge de Branches:

Volte para a branch principal:

git checkout main

Faça o merge da branch `nova_funcionalidade`:

git merge nova_funcionalidade

Parte 3: Colaboração com GitHub

1. Subir o Projeto para o GitHub:

Faça o push das mudanças para o repositório remoto:

git push origin main

3. Usar o `.gitignore`:

Crie um arquivo com o nome: **.gitignore**

Atenção para o ponto no início do nome.

Crie um arquivo com o nome: arquivo-ignorado

digite, dentro do .gitignore: arquivo-ignorado

Adicione e commite o `.gitignore`.

Faça o push para o seu repositório e entregue a atividade conforme combinado.