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](https://git-scm.com/downloads)

**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**](https://code.visualstudio.com/download)

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.

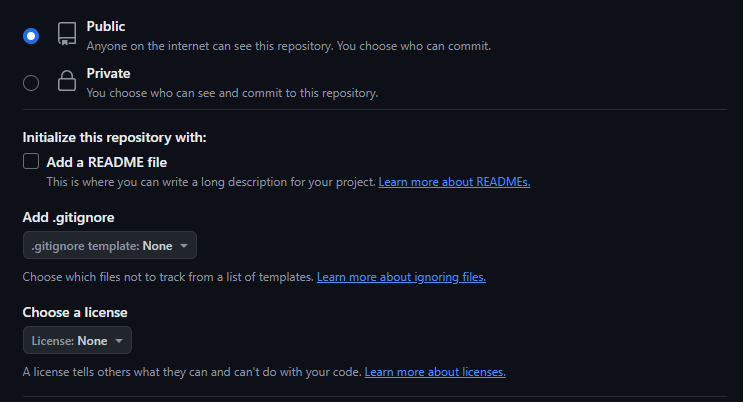
Inicialize um repositório Git:

*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



## 4. Conectar Repositório Local ao GitHub:

Adicione o repositório remoto:

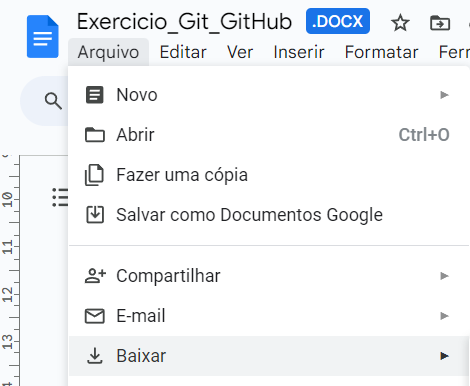
Abaixo, em vermelho, estão valores que deverão ser mudados no seu link, de acordo com o seu nome de usuário e o nome do repositório Remoto.

*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.