

# Instrução para a Prática – Controle de Versão com Git + GitHub

## **Objetivo:**

Demonstrar como criar e controlar versões de um projeto usando Git e GitHub.

## **PASSO A PASSO**

### ◆ **1. Criar uma conta no GitHub**

Acesse: <https://github.com>

- Clique em **Sign up** (Cadastrar)
- Siga as etapas de criação (nome de usuário, email, senha etc.)
- Após o cadastro, confirme o email e faça login

### **2. Criar um repositório no GitHub**

1. No canto superior direito, clique no botão **+** e depois em **"New repository"**
2. Preencha os dados:
  - **Repository name:** engenharia-software
  - **Description (opcional):** Repositório da disciplina de Engenharia de Software
  - Escolha **Public**
  - Marque **Add a README file**
3. Clique em **Create repository**

### 3. Clonar o repositório no computador

⚠ certifique-se que o Git está instalado:

`git --version` (verifica a versão do git)

Clone o repositório:

`git clone https://github.com/SEU\_USUARIO/engenharia-software.git`

Substitua SEU\_USUARIO pelo nome de usuário real do GitHub.

### 4. Criar um arquivo chamado README.md

Acesse a pasta clonada:

`cd engenharia-software`

Se o arquivo não existir, crie/edit:

`echo "# Projeto Engenharia de Software" > README.md`

### 5- Fazer commit com mensagem descritiva

```
git add README.md
```

```
git commit -m "RF01 - Tela de login criada"
```

```
git push origin main
```

**Cada commit é um ponto no tempo com histórico. A mensagem deve seguir um padrão como RF01.**

## **6- Criar uma branch chamada versao-1.0**

```
git checkout -b versao-1.0
```

```
git push origin versao-1.0
```

**A Branch serve para isolar uma nova versão ou funcionalidade sem afetar a principal (main ou master).**

## **7 - Gerar um release no GitHub**

1. Vá até o repositório no GitHub
2. Clique na aba "Releases"
3. Clique em "Draft a new release"
4. Em Tag version, digite: v1.0
5. Em Release title, coloque: Versão 1.0 estável
6. Descreva as mudanças feitas (ex: tela de login, cadastro etc.)
7. Clique em "Publish release"

**Releases são usados para empacotar versões estáveis do software.**

**Padrão de Labels (para mensagens de commit)**

<b>Código</b>	<b>Descrição</b>
<b>RF01</b>	<b>Cadastro de usuários</b>
<b>RF02</b>	<b>Tela de login com autenticação</b>
<b>BUG01</b>	<b>Corrigir erro de redirecionamento</b>
<b>REL01</b>	<b>Primeiro release para o cliente</b>