

MANUAL DE INSTRUÇÕES: GIT E GITHUB

1. Introdução

1.1 O que é Git?

Git é um sistema de controle de versão distribuído que permite a desenvolvedores rastrear e trabalhar juntos em projetos, seja de pequena ou grande escala. Ele permite que você salve "versões" (ou "commits") de seu código ao longo do tempo, reverta para versões anteriores se necessário, e trabalhe em diferentes "ramificações" de um projeto ao mesmo tempo.

1.2 O que é GitHub?

GitHub é um serviço baseado na nuvem que utiliza o Git para controle de versão. Ele fornece um local centralizado para armazenar repositórios públicos e privados, facilita a colaboração em projetos e fornece ferramentas para ajudar a gerenciar e rastrear o trabalho.


2. Instalando o Git no Debian

Para instalar o Git em sistemas baseados no Debian, siga os seguintes passos:

1. Abra um terminal.


2. Atualize o sistema com o comando:

```
bash
sudo apt update
```

 Copy code


3. Instale o Git usando:

```
bash
sudo apt install git
```

 Copy code

4. Confirme a instalação verificando a versão do Git:

```
bash
git --version
```

 Copy code

3. Clonando e Atualizando um Repositório

3.1 Clonando um Repositório

Para clonar o repositório "<https://github.com/aXR6/utilitarios>" a partir do GitHub:

1. Navegue até o local desejado no seu sistema usando o terminal.
2. Utilize o comando:

```
bash
```

[Copy code](#)

```
git clone https://github.com/aXR6/utilitarios.git
```

Após a execução, uma pasta chamada "<https://github.com/aXR6/utilitarios>" será criada com todo o conteúdo do repositório.

3.2 Atualizando um Repositório

Se o repositório original (chamado "upstream") tiver novas atualizações e você deseja sincronizar essas atualizações com a sua versão local, siga os passos:

1. Entre no diretório do projeto:

```
bash
```

[Copy code](#)

```
cd utilitarios
```

2. Atualize o repositório usando:

```
bash
```

[Copy code](#)

```
git pull
```

O Git buscará as atualizações do repositório remoto e aplicará as mudanças ao seu repositório local.

4. Utilizando o GitHub

O GitHub não é apenas um lugar para armazenar código. Ele oferece ferramentas para:

- **Colaboração:** Os desenvolvedores podem colaborar em projetos sem interferir no trabalho uns dos outros usando "branches".
- **Rastreamento de Problemas:** O GitHub oferece um sistema de "issues" que permite aos usuários relatar bugs ou solicitar novos recursos.
- **Documentação:** Com o recurso "GitHub Pages", os usuários podem criar sites para seus projetos.
- **Integração Contínua:** Usando "GitHub Actions", os desenvolvedores podem automatizar testes e implantações para seus projetos.

Casos de Uso

1. **Trabalhando em Equipe:** Se você estiver trabalhando em uma equipe, poderá criar uma "branch" para um novo recurso, trabalhar nela e, quando estiver

pronto, criar um "pull request" para que a equipe reveja e adicione suas alterações ao projeto principal.

2. **Documentando seu Projeto:** Utilize o "README.md" para documentar seu projeto, explicar o que ele faz, como configurá-lo e outras informações relevantes.
3. **Automatizando Tarefas:** Com "GitHub Actions", você pode automatizar tarefas como testar seu código cada vez que você ou alguém faz uma alteração, garantindo que o código esteja sempre funcional.