

Documento Explicativo - Github

Parte 2: GitHub


O que é o GitHub?

O **GitHub** é uma plataforma online que hospeda repositórios Git e permite colaboração entre desenvolvedores. Ele facilita o trabalho em equipe, versionamento, revisão de código e distribuição de software.

 Embora o Git funcione localmente, o GitHub permite que você compartilhe seus repositórios com o mundo.


1. Criando sua Conta no GitHub

1. Acesse <https://github.com>
2. Clique em **Sign up**.
3. Preencha com um nome de usuário, e-mail e senha.
4. Escolha o plano gratuito e siga as instruções de verificação.

 Use um e-mail válido e de fácil acesso, pois você precisará confirmá-lo para usar todas as funcionalidades.

2. Criando seu Primeiro Repositório

1. Após logar, clique no botão **+** no canto superior direito.
2. Selecione **New repository**.
3. Preencha:
 - Nome do repositório
 - Descrição (opcional)
 - Visibilidade: pública ou privada
4. (Opcional) Marque a opção **Initialize this repository with a README**.
5. Clique em **Create repository**.

 Repositórios públicos são visíveis a todos, ideais para projetos open source. Privados são ótimos para projetos em desenvolvimento ou pessoais.

3. Fazendo Push de Outras Branches

Depois de criar uma nova branch no Git e realizar commits nela, você pode enviar essa branch ao GitHub com:


```
1 git push origin nome-da-branch
2
```

Por exemplo:

```
1 git push origin AddMenu
2
```

Por que isso é importante?


Permite que outros colaboradores vejam seu progresso e revisem sua implementação antes de mesclar com a branch principal.

 Use branches separadas para cada funcionalidade para facilitar a organização e revisão do projeto.

4. Criando uma Issue

As **issues** são utilizadas para relatar bugs, sugerir melhorias ou acompanhar tarefas.

1. Vá até o repositório no GitHub.
2. Clique em **Issues**.
3. Clique em **New issue**.
4. Adicione um título e uma descrição clara.
5. (Opcional) Atribua pessoas, marque com labels ou conecte a um projeto.
6. Clique em **Submit new issue**.

 Use as issues como ferramenta de comunicação com a equipe ou para documentar ideias e pendências.


5. Criando um Release

Releases são versões "congeladas" de um projeto, geralmente associadas a marcos como lançamentos de produto ou novas funcionalidades.

1. No repositório, clique em **Releases > Draft a new release**.
2. Escolha a tag (ex: `v1.0.0`).
3. Preencha o título e notas da versão.
4. (Opcional) Anexe arquivos binários, se for o caso.
5. Clique em **Publish release**.

Por que criar um release?

Ajuda a organizar versões e facilita a instalação ou uso do seu software por outras pessoas.

 Siga convenções como o [SemVer](#) (ex: `v1.2.3`) para facilitar o entendimento das alterações entre versões.

6. Realizando o Pull das Alterações

Se alguém fizer mudanças no repositório (ou você em outro computador), é preciso trazê-las para seu ambiente local:

```
1 git pull origin main
2
```

Substitua `main` pelo nome da branch, se necessário.

Por que isso é necessário?

Mantém seu repositório local atualizado com as alterações feitas no repositório remoto.

 Sempre dê `git pull` antes de começar a trabalhar para evitar conflitos e garantir que você está na versão mais atual.

7. Gerando e Utilizando um Token de Acesso Pessoal (PAT - Personal Access Token)

Desde agosto de 2021, o GitHub não aceita mais autenticação com senha via linha de comando. É necessário usar um token de acesso.

Passo a Passo para Gerar o Token:

1. Acesse seu perfil do GitHub e clique na sua foto (canto superior direito).
2. Vá em **Settings**.
3. No menu lateral, clique em **Developer settings**.
4. Em seguida, clique em **Personal access tokens > Tokens (classic)**.
5. Clique em **Generate new token (classic)**.
6. Dê um nome ao token e defina sua validade.
7. Marque os escopos de permissão (ex: `repo`, `workflow`, etc.) dependendo do que você precisa fazer.
8. Clique em **Generate token**.
9. Copie o token gerado e guarde-o em local seguro (você não poderá vê-lo novamente).

Como Usar o Token no Terminal:

Ao executar um comando como `git push` ou `git pull`, o terminal solicitará sua autenticação:

- **Username:** coloque seu nome de usuário do GitHub.
- **Password:** cole o token gerado (não a sua senha do GitHub).

```
1 git push origin main
2 Username: seu-usuario
3 Password: <cole aqui o token gerado>
```

Dica: Use um gerenciador de credenciais como o Git Credential Manager para não precisar digitar o token toda vez.

 Você também pode usar autenticação via SSH como alternativa segura ao uso do token.