

Guia Detalhado para Criar um Repositório e Enviar Alterações para o GitHub

1. Inicializando um Repositório Git Local

Primeiro, você precisa criar um novo repositório Git no seu computador:

Abra o terminal ou prompt de comando

Navegue até a pasta do seu projeto usando `cd caminho/para/pasta`

Execute o comando:

- `git init`

Isso cria um novo subdiretório `.git` que contém todos os arquivos necessários do repositório.

2. Configurando o Repositório Remoto no GitHub

Acesse sua conta no GitHub (<https://github.com>)

Clique no botão "+" no canto superior direito e selecione "New repository"

Digite um nome para seu repositório

Escolha se será público ou privado

Podes inicializar com um README ou escolher outras opções, mas para este tutorial vamos começar vazio

Clique em "Create repository"

3. Conectando o Repositório Local ao Remoto

No terminal, ainda na pasta do seu projeto, execute:

`git remote add origin https://github.com/seu-usuario/nome-do-repositorio.git`

- Substitua "seu-usuario" e "nome-do-repositorio" pelos seus dados. Isso conecta seu repositório local ao remoto.

4. Adicionando e Confirmando Alterações

Passo 4.1: Verificando o status

Antes de adicionar arquivos, verifique o status:

- `git status`

Isso mostra quais arquivos foram modificados e quais estão prontos para commit.

Passo 4.2: Adicionando arquivos

Para adicionar todos os arquivos novos e modificados:

- `git add .`

Ou para adicionar arquivos específicos:

- `git add nome-do-arquivo.ext`

Passo 4.3: Fazendo o commit

Depois de adicionar os arquivos, faça um commit com uma mensagem descritiva:

- `git commit -m "Sua mensagem de commit aqui"`

5. Enviando as Alterações para o GitHub

Primeiro envio (quando o branch principal ainda não existe remotamente):

- `git push -u origin main`

O `-u` define o upstream, então nas próximas vezes você pode usar apenas `git push`.

Envios subsequentes:

- `git push`

6. Fluxo de Trabalho Básico

O ciclo normal de trabalho com Git segue estes passos:

Fazer alterações nos arquivos

`git add` para preparar as alterações

`git commit` para confirmar as alterações localmente

`git pull` para atualizar com alterações remotas (se trabalhar em equipe”nosso caso”)

Resolver conflitos se necessário

`git push` para enviar suas alterações para o GitHub

7. Trabalhando com Branches (Opcional)

Para criar um novo branch:

- `git checkout -b nome-do-novo-branch`

Para alternar entre branches:

- `git checkout nome-do-branch`

Para enviar um novo branch para o GitHub:

- `git push -u origin nome-do-branch`

8. Atualizando seu Repositório Local

Se outras pessoas fizerem alterações no repositório remoto, você pode atualizar seu local com:

- `git pull origin main`

Isso busca as alterações e as mescla com teu branch local.

Dicas Adicionais

Sempre verifique `git status` para entender o estado atual

Use `git log` para ver o histórico de commits

Para desfazer alterações não commitadas: `git checkout -- nome-do-arquivo`

Para desfazer um `git add`: `git reset nome-do-arquivo`

By:Diogo Oliveira