

Guia Completo para Instalar e Configurar o Git e o GitHub no Windows

Este guia prático foi feito para iniciantes. Vamos cobrir a instalação do **Git** no Windows, a configuração do **GitHub**, a autenticação segura com tokens, e os comandos essenciais para você começar a versionar seus projetos.

1. Instalando o Git no Windows

1. **Baixe o instalador:** Acesse o [site oficial do Git](#) e o download do arquivo .exe deve começar automaticamente.
2. **Execute o instalador:** Dê um duplo clique no arquivo baixado.
3. **Siga o assistente de instalação:**
 - Clique em **"Next"** em todas as telas até chegar em **"Adjusting your PATH environment"**.
 - **Importante:** Mantenha a opção padrão **"Git from the command line and also from 3rd-party software"**. Isso permite que você use o Git em qualquer terminal.
 - Nas telas seguintes, as opções padrão já são as mais recomendadas. Pode clicar em **"Next"** até o final e, depois, em **"Install"**.
4. **Conclua a instalação:** Ao final, desmarque a opção **"View Release Notes"** e clique em **"Finish"**.

Verificando a instalação:

Abra o Prompt de Comando (ou Terminal) e digite:

```
Bash
```

```
git --version
```

Se a versão do Git for exibida, a instalação foi um sucesso!

2. Configurando sua Conta do GitHub

Agora, configure seu nome de usuário e e-mail para que o Git saiba quem está fazendo as alterações.

Bash

```
# Define seu nome de usuário (use o mesmo nome do seu perfil do GitHub)
git config --global user.name "Seu Nome de Usuário"
```

```
# Define seu e-mail (use o mesmo e-mail do seu perfil do GitHub)
git config --global user.email "seu.email@exemplo.com"
```

3. Criando e Usando um Token de Acesso Pessoal (PAT)

O GitHub descontinuou o uso de senhas para operações Git. Agora, usamos um **Token de Acesso Pessoal (PAT)**, uma senha especial e mais segura.

Passo a Passo para Gerar o PAT

1. **Acesse as Configurações:** No seu perfil do GitHub, clique na sua foto no canto superior direito e selecione **"Settings"** (Configurações).
2. **Vá para as Configurações de Desenvolvedor:** No menu lateral esquerdo, clique em **"Developer settings"** (Configurações de desenvolvedor).
3. **Encontre a Seção de Tokens:** Clique em **"Personal access tokens"** e depois em **"Tokens (classic)"**.
4. **Crie um Novo Token:** Clique no botão verde **"Generate new token (classic)"**.
5. **Preencha os Detalhes do Token:**
 - **Note:** Dê um nome para o seu token (Ex: "Meu PC Pessoal").
 - **Expiration:** Defina um tempo de expiração (Ex: 90 dias).
 - **Select scopes:** Marque a caixa **repo**. Isso dá ao token permissão para ler e escrever em seus repositórios.
6. **Gere e Copie o Token:**
 - Clique em **"Generate token"**.

- **ATENÇÃO:** O GitHub mostrará o token **apenas uma vez**. **Copie-o imediatamente** e salve-o em um local seguro antes de sair da página.

Como Usar o Token

A primeira vez que você executar um comando que precisa de permissão (como git push), o **Gerenciador de Credenciais do Windows** abrirá uma janela. No campo "Password", **cole o Token de Acesso Pessoal (PAT)** que você acabou de copiar. O Gerenciador de Credenciais salvará o token com segurança.

4. Primeiros Passos com o Git: Comandos Essenciais

Vamos criar um novo projeto e usar os comandos básicos do Git.

git init

Este comando inicializa um novo repositório Git em uma pasta. É o ponto de partida para o Git começar a rastrear as alterações.

Bash

```
# Navegue até a pasta do seu projeto  
cd C:\caminho\para\seu\projeto
```

```
# Inicialize o repositório Git  
git init
```

git status

Este é seu melhor amigo. O git status mostra o estado atual do seu repositório: quais arquivos foram alterados, quais estão prontos para serem salvos (commitados) e quais ainda não estão sendo rastreados.

```
Bash
```

```
git status
```

git add

Quando você cria ou altera um arquivo, ele fica na "área de trabalho". Use o git add para incluir essa alteração no próximo "salvamento".

```
Bash
```

```
# Adiciona um arquivo específico para ser "salvo"  
git add nome_do_arquivo.txt
```

```
# Adiciona todos os arquivos alterados  
git add .
```

git commit

O commit é o ato de salvar suas alterações de forma definitiva no histórico do Git. É como um "ponto de salvamento".

```
Bash
```

```
git commit -m "Mensagem clara sobre as alterações feitas"
```

5. Conectando o Repositório Local ao GitHub

Após criar seu repositório no GitHub, você precisa dizer ao seu projeto local "onde" o repositório online está.

1. **Crie o repositório no GitHub:** Siga os passos do item 3 para criar um novo repositório e copie a URL HTTPS dele.
2. **Conecte seu repositório local:**

Bash

```
# Exemplo: conectando seu projeto local ao repositório no GitHub
```

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

- git remote: Comando para gerenciar repositórios online.
- add origin: Adiciona um novo repositório com o apelido origin.
- [URL]: A URL do seu repositório no GitHub.

git push

O push envia seus commits do seu computador local para o repositório no GitHub.

Bash

```
# Envia os commits da branch principal para o GitHub
```

```
git push -u origin main
```

A flag -u é usada apenas na primeira vez para "ligar" a branch local main à branch remota main.

git pull

O pull baixa as alterações que foram feitas no repositório remoto para o seu computador. Use-o sempre antes de começar a trabalhar para garantir que você tenha a versão mais recente.

Bash

```
git pull origin main
```

6. Trabalhando com Branches

Branches são como "ramificações" do seu projeto. Elas permitem que você desenvolva novos recursos sem afetar o código principal.

git branch

- `git branch`: Lista todas as branches do seu projeto. A branch atual é marcada com um `*`.
- `git branch nome-da-branch`: Cria uma nova branch.

git checkout

O checkout permite que você alterne entre as branches.

Bash

Altera para a nova branch
`git checkout nome-da-branch`

Ou, de forma mais rápida, crie e alterne para a nova branch em um só comando
`git checkout -b nova-branch`

git merge

Quando terminar de trabalhar em uma branch, você pode mesclá-la (merge) com a branch principal.

1. **Vá para a branch principal:** `git checkout main`
 2. **Mescle a sua branch:** `git merge nome-da-branch`
-

7. Como Apagar as Credenciais do GitHub no Windows

Se você precisar remover suas informações de acesso de um computador, o Windows as armazena no **Gerenciador de Credenciais**.

1. **Abra o Gerenciador de Credenciais:**
 - Clique no menu Iniciar e pesquise por "**Gerenciador de Credenciais**".
2. **Acesse as Credenciais do Windows:**
 - Na janela do gerenciador, clique em "**Credenciais do Windows**".
3. **Localize e remova as credenciais do GitHub:**
 - Procure por uma entrada que comece com **git:** ou **github.com**.
 - Clique na entrada para expandir os detalhes e, em seguida, em "**Remover**".