

[ENTRAR](#)[MATRICULE-SE](#)[TODOS OS
CURSOS](#)[NOSSAS
FORMAÇÕES](#)[PARA
EMPRESAS](#)[DEV
EM <T>](#)[ARTIGOS > DEVOPS](#)

Nova exigência do Git de autenticação por token, o que é e o que devo fazer?

**Camila Fernanda Alves**

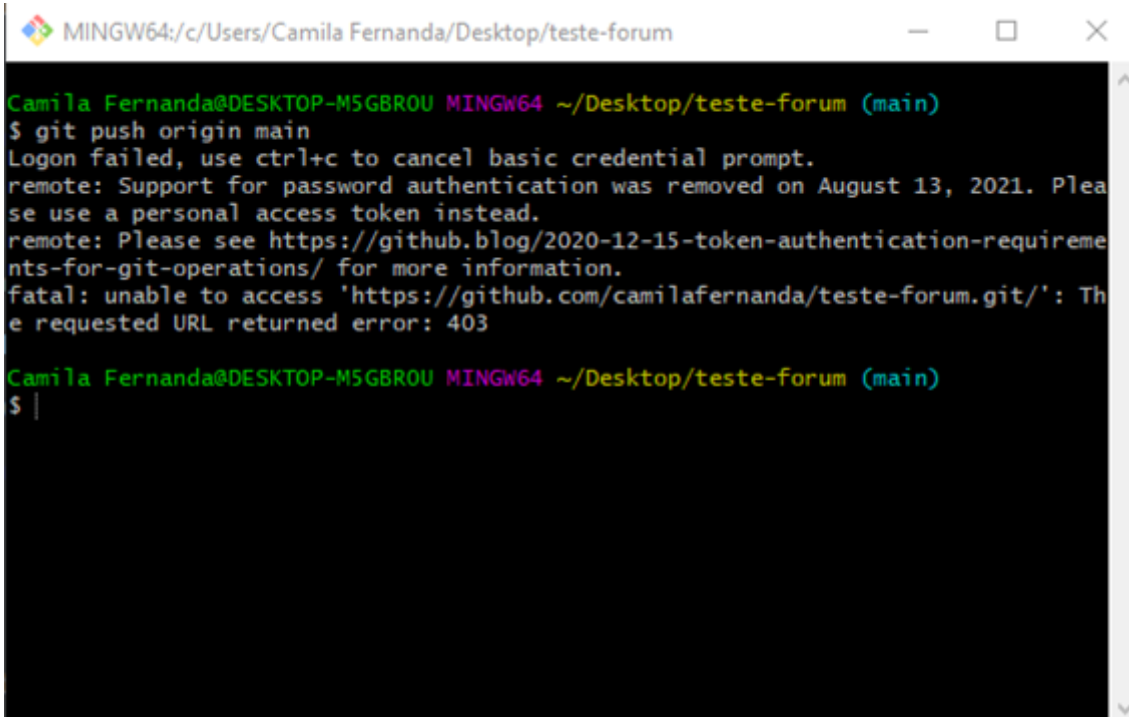
22/09/2021

[COMPARTILHE](#)

Nova exigência do Git de autenticação por token, o que é e o que devo fazer?

Ultimamente, você se deparou com essa mensagem ao fazer alguns comandos no Git?

```
Logon failed, use ctrl+c to cancel basic credential prompt.  
remote: Support for password authentication was removed on August 13,  
remote: Please see https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/ for more information.  
fatal: unable to access '<repositório Git>': The request URL returned
```

A screenshot of a terminal window titled 'MINGW64:/c/Users/Camila Fernanda/Desktop/teste-forum'. The terminal shows the command '\$ git push origin main' and the resulting error message: 'Logon failed, use ctrl+c to cancel basic credential prompt. remote: Support for password authentication was removed on August 13, 2021. Please use a personal access token instead. remote: Please see https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/ for more information. fatal: unable to access 'https://github.com/camilafernanda/teste-forum.git/': The requested URL returned error: 403'. The prompt '\$ |' is visible at the bottom.

```
MINGW64:/c/Users/Camila Fernanda/Desktop/teste-forum  
Camila Fernanda@DESKTOP-M5GBROU MINGW64 ~/Desktop/teste-forum (main)  
$ git push origin main  
Logon failed, use ctrl+c to cancel basic credential prompt.  
remote: Support for password authentication was removed on August 13, 2021. Please use a personal access token instead.  
remote: Please see https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/ for more information.  
fatal: unable to access 'https://github.com/camilafernanda/teste-forum.git/': The requested URL returned error: 403  
  
Camila Fernanda@DESKTOP-M5GBROU MINGW64 ~/Desktop/teste-forum (main)  
$ |
```

Isso ocorreu, porque o GitHub [anunciou](https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/) uma nova exigência de autenticação baseada em **token** para o Git. A partir do dia 13 de Agosto de 2021, a senha da conta do GitHub passou a não ser aceita mais para autenticar operações no Git.

Essa mudança se deve por **motivos de segurança**, pois os tokens oferecem alguns benefícios, como ter a possibilidade de salvar as informações do usuário no token e não no servidor, bem como o fato de ser configurado para ter as permissões que você quiser e ter um período de expiração. Ou seja, ele para de funcionar após determinado tempo, como se fosse uma senha única e descartável.

Assim, caso outra pessoa tenha acesso ao seu token, ela estará restrita apenas às permissões que você configurou para o mesmo. Assim você não irá correr o risco de perder sua conta do GitHub, podendo descartar o token perdido e gerar outro novamente.

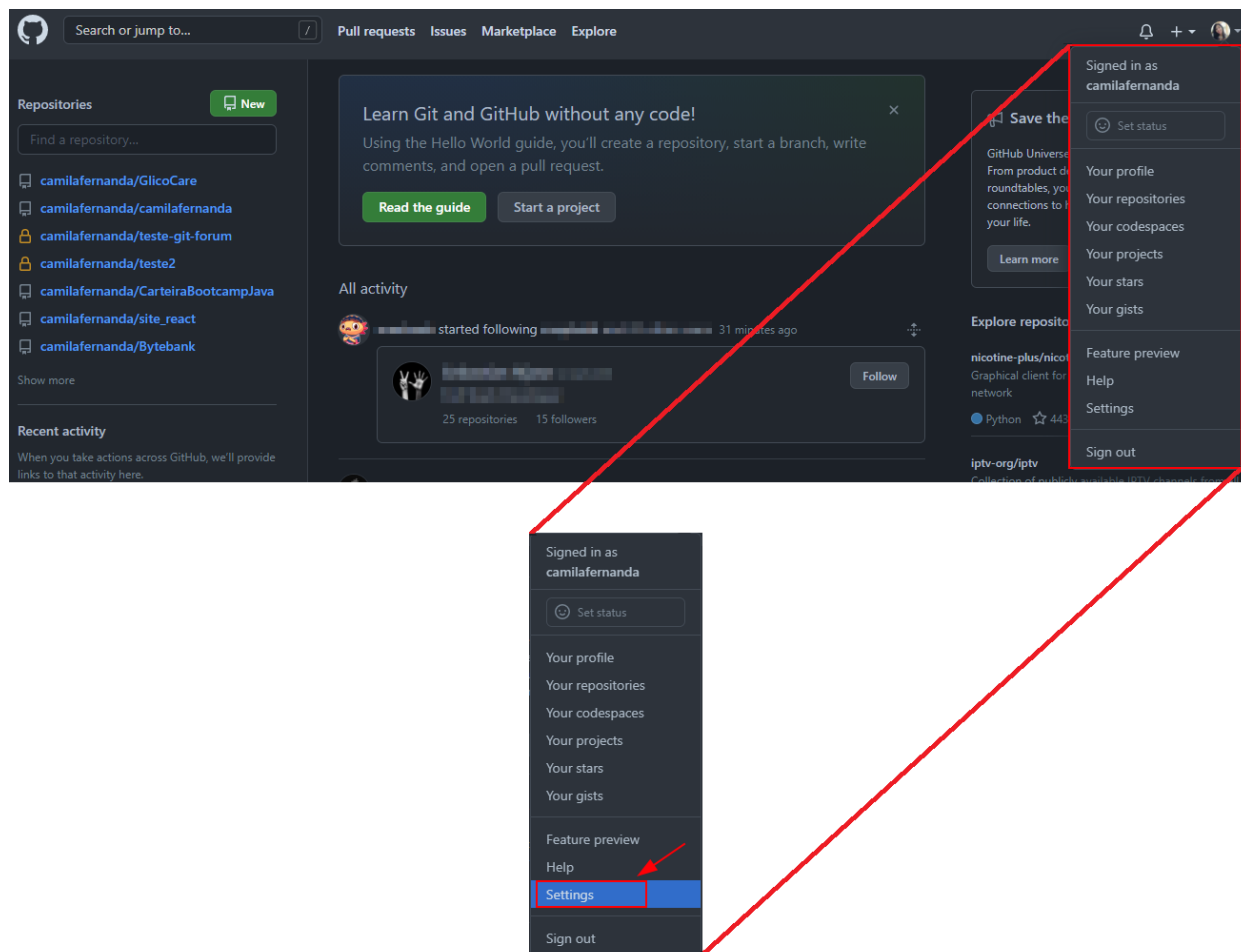
Mas então, o que é preciso fazer para continuar usando as operações Git?

O GitHub recomenda começar a utilizar um token de acesso pessoal, mas caso queira, pode optar por utilizar uma chave SSH.

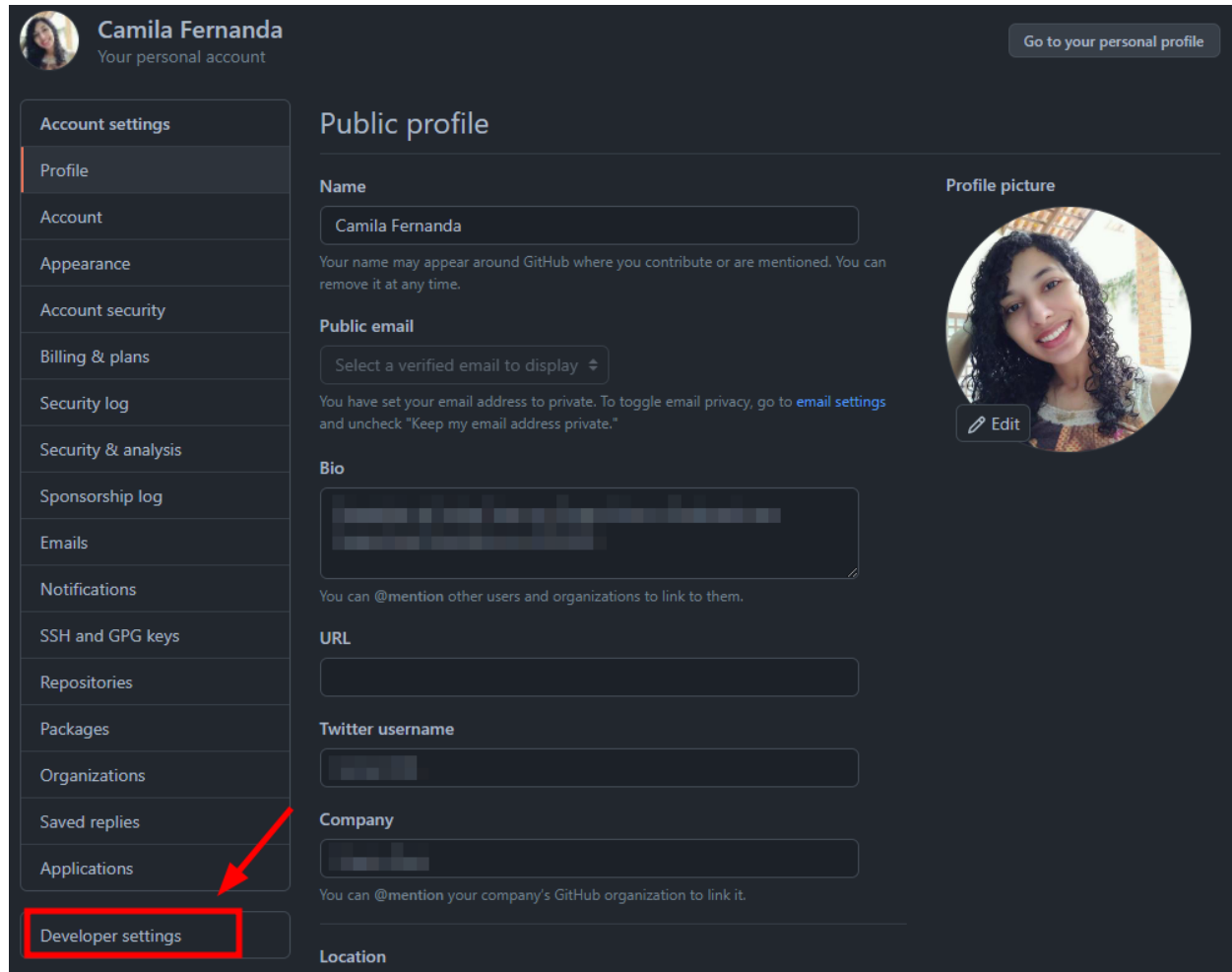
Dessa forma, segue um passo a passo de como gerar seu token pessoal:

1) Faça o login na página do [GitHub](#);

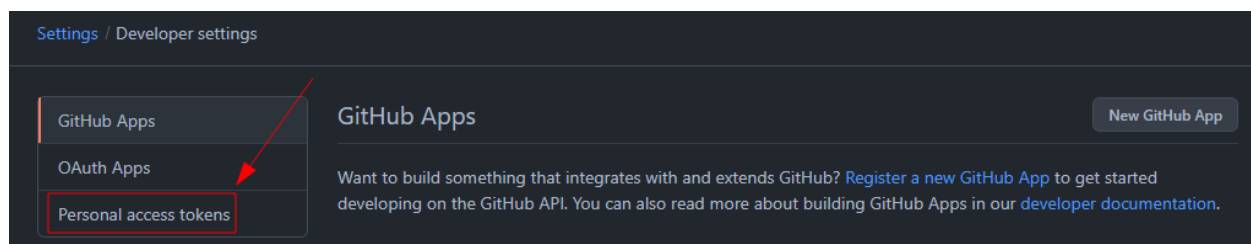
2) Na página principal, selecione seu usuário no canto superior direito e clique na opção **Settings**:



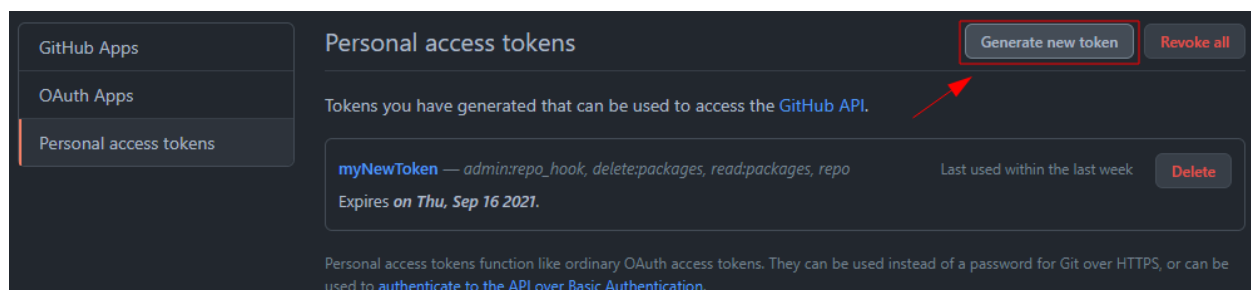
3) Na página que irá abrir, clique na opção **Developer settings**, 18ª opção da lista à esquerda.



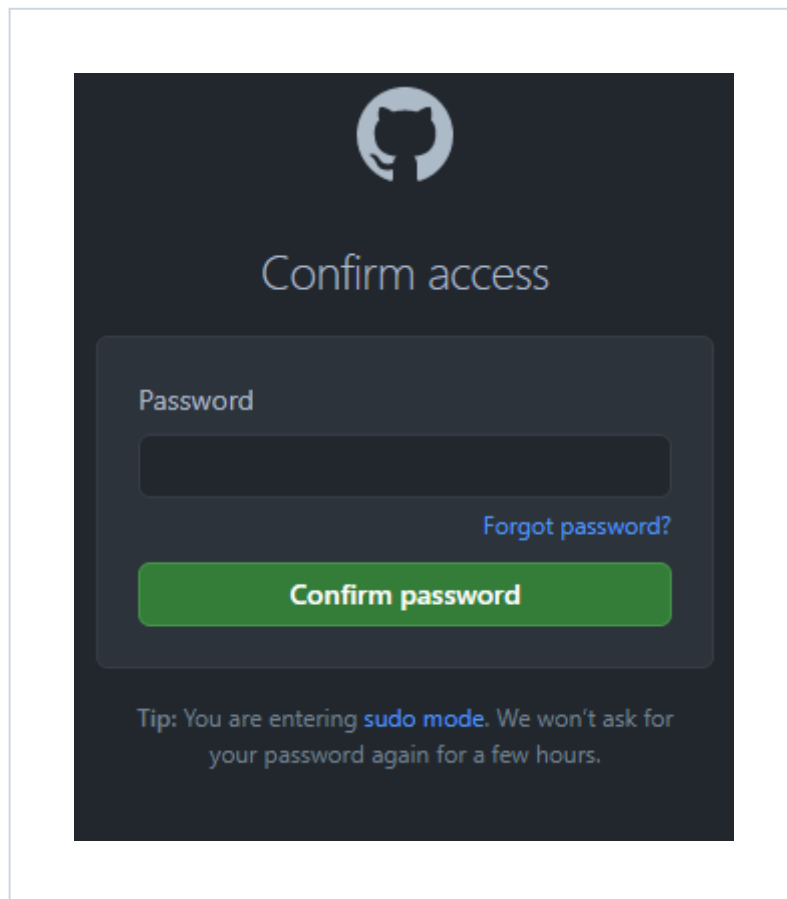
4) Selecione a opção **Personal access tokens**, 3ª opção da lista à esquerda.



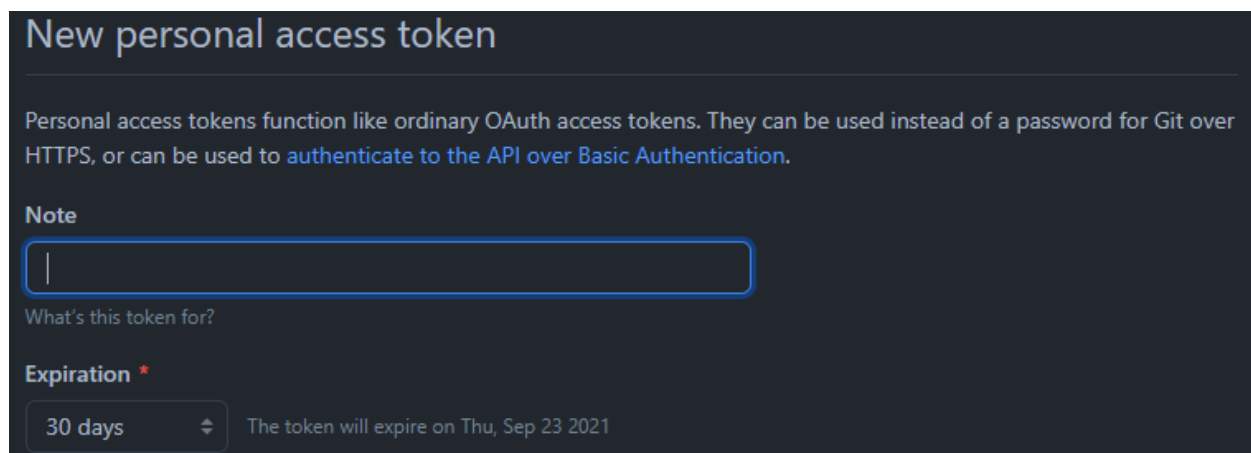
5) Clique na opção **Generate new token**.



6) Confirme com sua senha do GitHub.



7) Dê um nome para seu token e defina seu tempo de expiração.



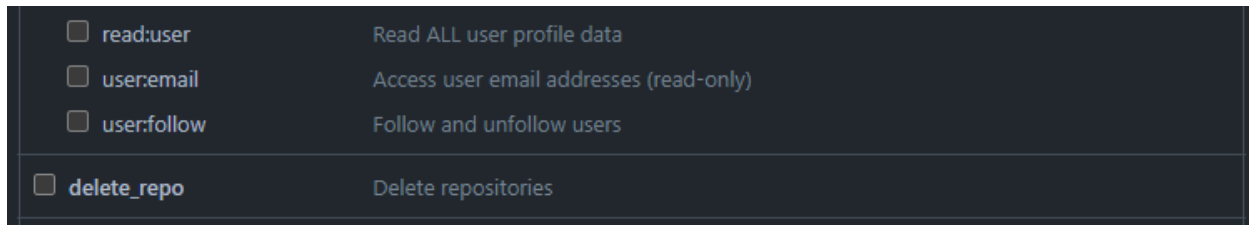
8) Marque as permissões que você deseja conceder ao token. Detalhe para user:all

8) Marque as permissões que você deseja conceder ao token. Defina, para usar seu token para acessar repositórios, selecione a opção **repo**.

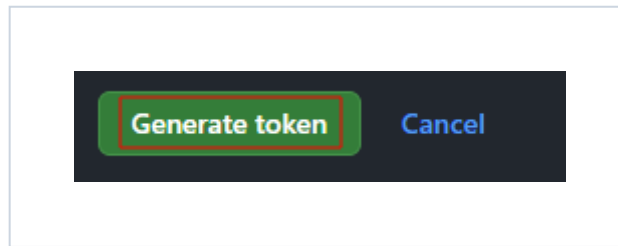
Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

- | | |
|---|---|
| <input checked="" type="checkbox"/> repo | Full control of private repositories |
| <input checked="" type="checkbox"/> repo:status | Access commit status |
| <input checked="" type="checkbox"/> repo_deployment | Access deployment status |
| <input checked="" type="checkbox"/> public_repo | Access public repositories |
| <input checked="" type="checkbox"/> repo:invite | Access repository invitations |
| <input checked="" type="checkbox"/> security_events | Read and write security events |
| <input type="checkbox"/> workflow | Update GitHub Action workflows |
| <input type="checkbox"/> write:packages | Upload packages to GitHub Package Registry |
| <input type="checkbox"/> read:packages | Download packages from GitHub Package Registry |
| <input type="checkbox"/> delete:packages | Delete packages from GitHub Package Registry |
| <input type="checkbox"/> admin:org | Full control of orgs and teams, read and write org projects |
| <input type="checkbox"/> write:org | Read and write org and team membership, read and write org projects |
| <input type="checkbox"/> read:org | Read org and team membership, read org projects |
| <input type="checkbox"/> admin:public_key | Full control of user public keys |
| <input type="checkbox"/> write:public_key | Write user public keys |
| <input type="checkbox"/> read:public_key | Read user public keys |
| <input type="checkbox"/> admin:repo_hook | Full control of repository hooks |
| <input type="checkbox"/> write:repo_hook | Write repository hooks |
| <input type="checkbox"/> read:repo_hook | Read repository hooks |
| <input type="checkbox"/> admin:org_hook | Full control of organization hooks |
| <input type="checkbox"/> gist | Create gists |
| <input type="checkbox"/> notifications | Access notifications |
| <input type="checkbox"/> user | Update ALL user data |



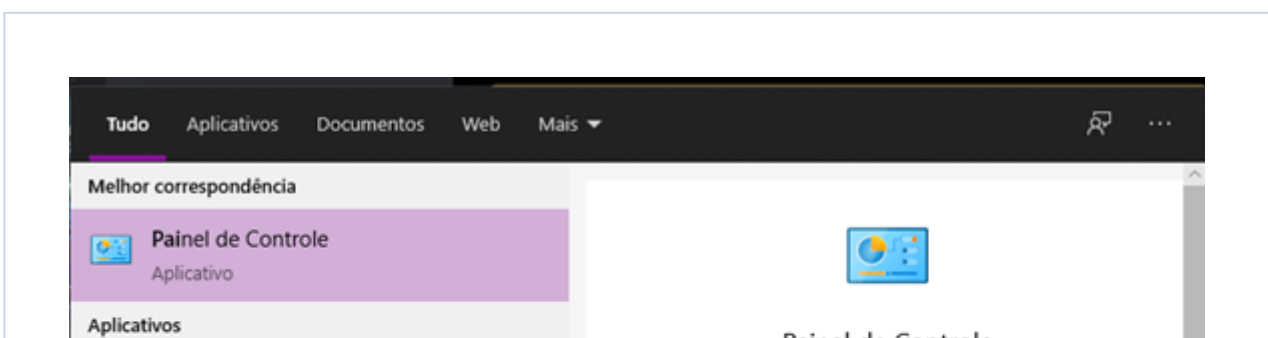
9) Por fim, ao selecionar todas as opções escolhidas, clique em Generate token ao final da tela.

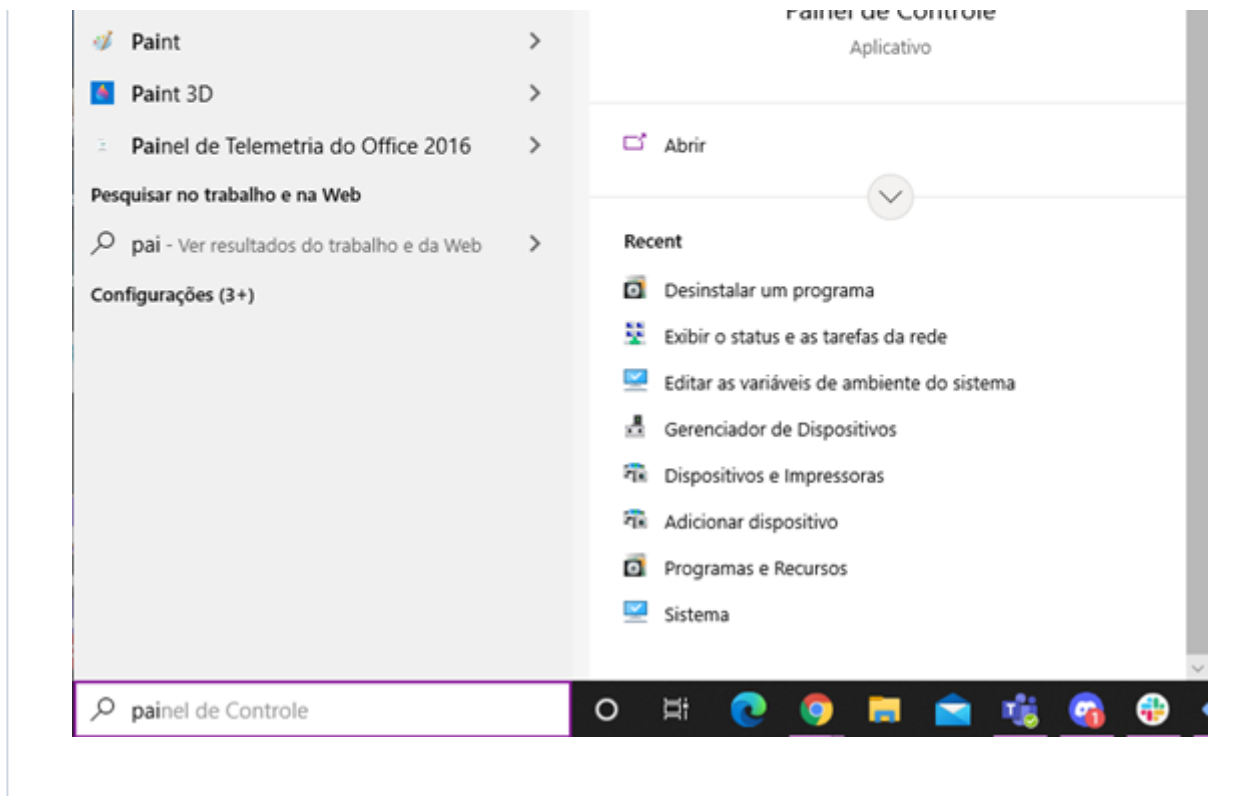


Prontinho, irá aparecer um código para que você possa copiar e colar no Gerenciador de Credenciais da sua máquina. Esse processo irá depender do seu sistema operacional.

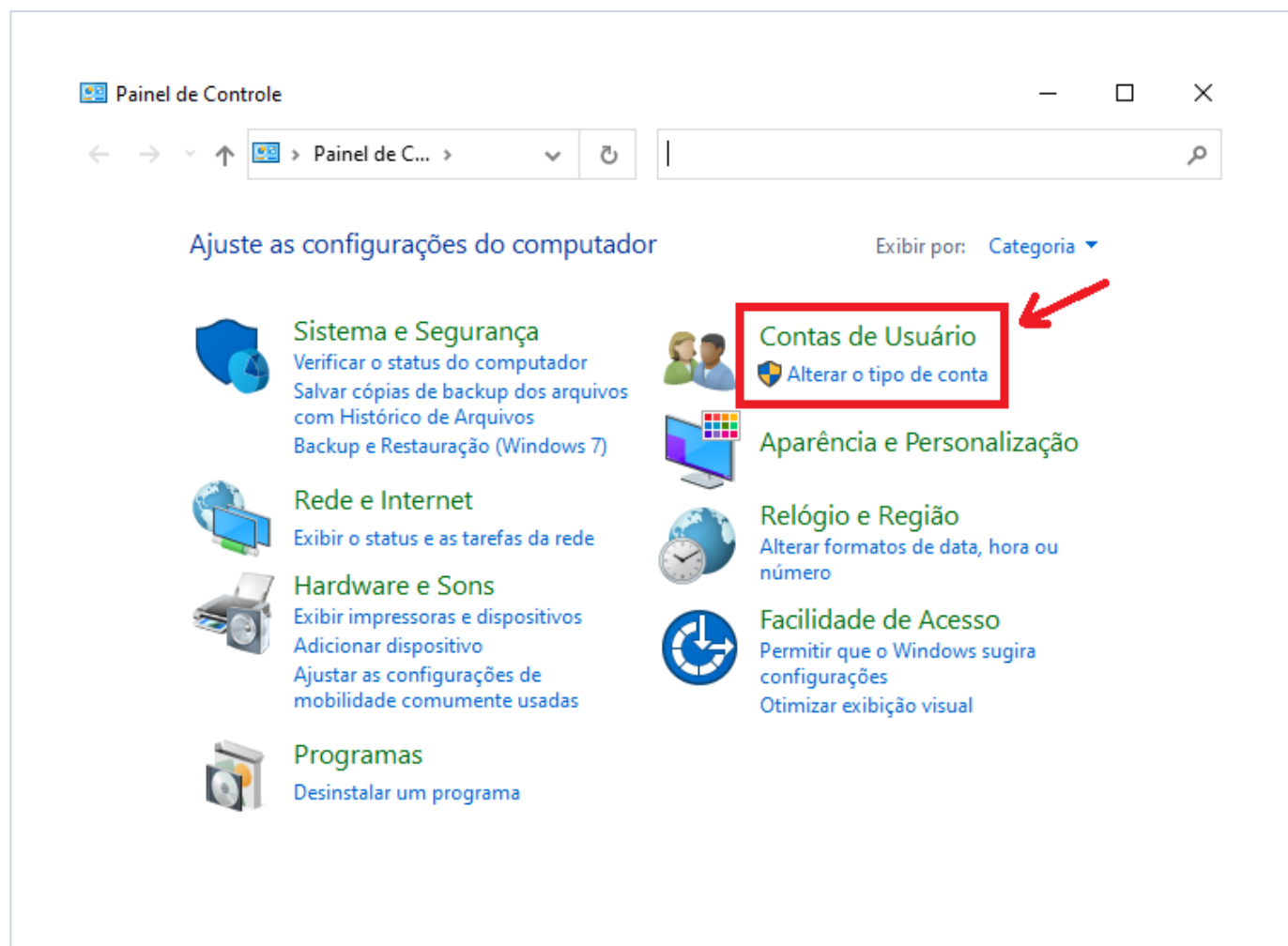
Como editar credenciais no Windows

Depois que você gerar o token, copie ele e acesse o **Painel de Controle**. Você pode encontrar digitando na barra de busca.

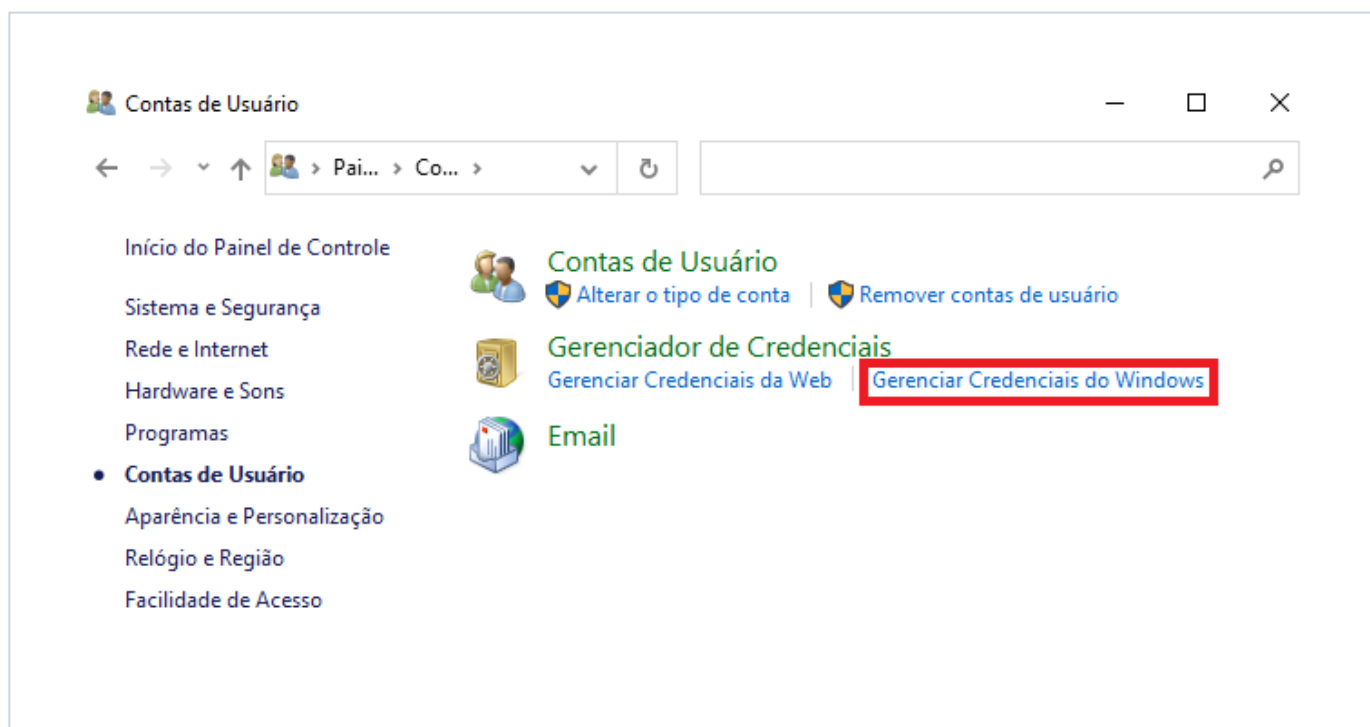




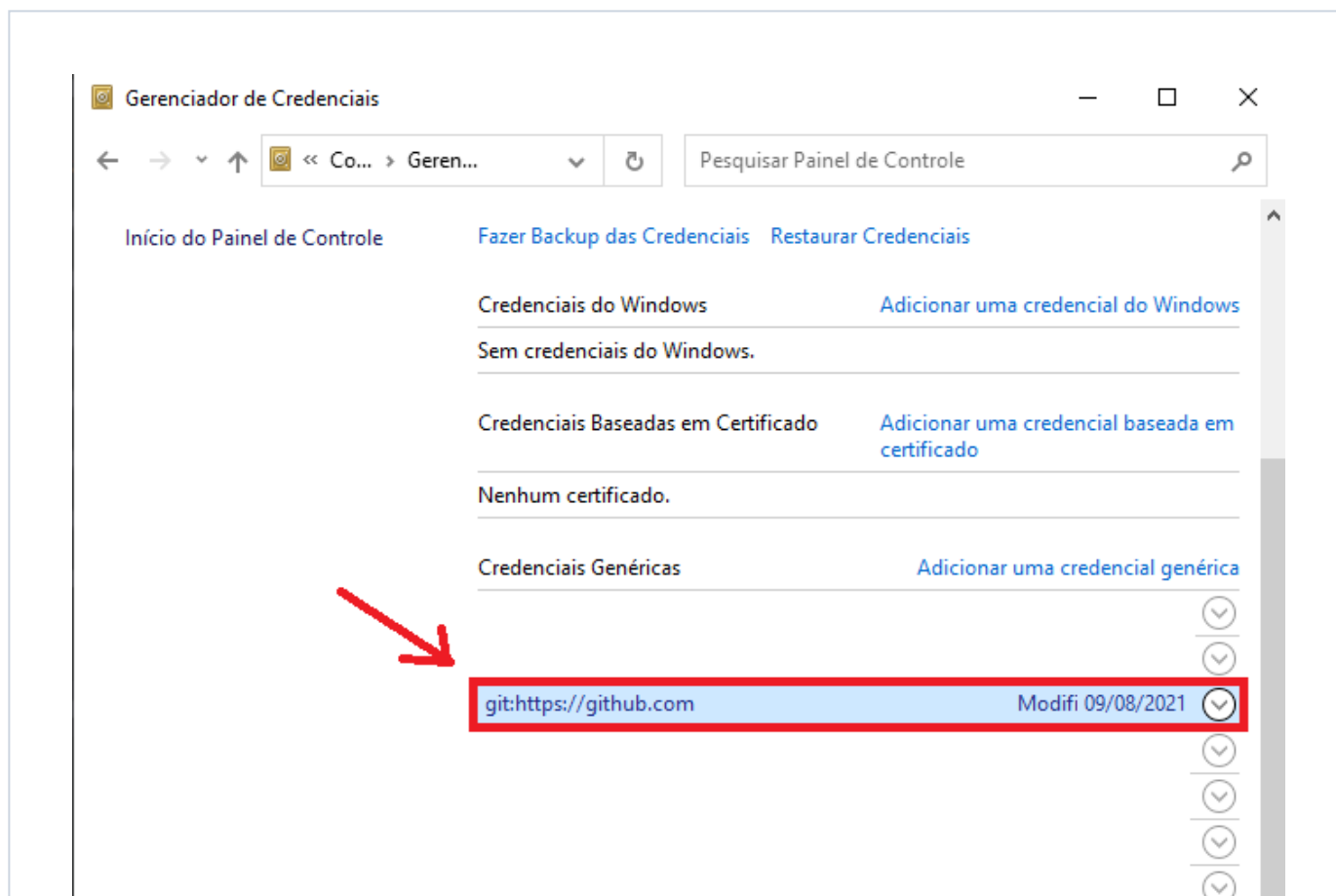
Em seguida, acesse **Contas de Usuário**:



Clique em **Gerenciar Credenciais do Windows**:



Procure na seção de **Credenciais Genéricas** o **github.com**, selecione-o e coloque para **editar**;



Consulte também
Contas de Usuário



Copie o token gerado na primeira etapa e cole na caixa de texto da **Senha**.

Editar Credencial Genérica

← → ↕ ⬅ << Ge... > Editar... ➡ ↺

Pesquisar Painel de Controle 🔍

Editar Credencial genérica

Verifique se o nome de usuário e a senha digitados podem ser usados para acessar o local.

Endereço de rede ou na Internet: git:https://github.com

Nome de usuário:


Senha:


Salvar Cancelar

Caso você não encontre o github.com, você pode selecionar a opção **Adicionar uma credencial genérica**

Gerenciar suas credenciais

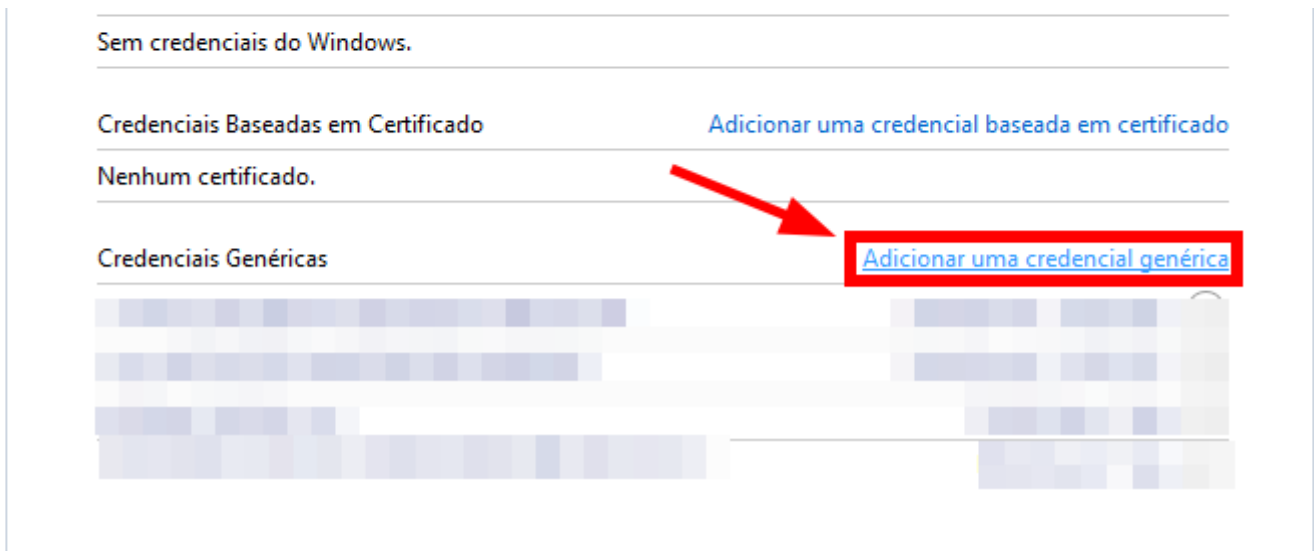
Exibir e excluir suas informações de login salvas para sites, aplicativos conectados e redes.

**Credenciais da Web**

**Credenciais do Windows**

[Fazer Backup das Credenciais](#) [Restaurar Credenciais](#)

[Credenciais do Windows](#) [Adicionar uma credencial do Windows](#)



E preencher nos campos:

Endereço de rede ou na Internet: `https://github.com/`

Nome de usuário: seu nome de usuário do GitHub

Senha: seu token pessoal gerado anteriormente

Digite o endereço do site e as informações de credencial

Verifique se o nome de usuário e a senha digitados podem ser usados para acessar o local.

Endereço de rede ou na Internet:

Nome de usuário:

Senha:

Prontinho, depois você pode realizar operações Git novamente sem requisições de login até a data de expiração do token gerado.

Como editar credencial Git no Linux

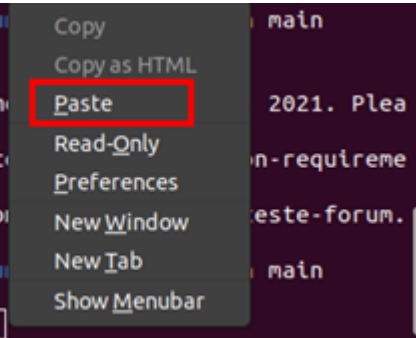
No Linux, este processo pode ser feito somente no terminal. Ao abrir um repositório Git nele e fazer comandos como `git pull`, `git push` ou `git clone`, o Git irá fazer uma requisição

de usuário e senha, caso você colocar a sua senha do GitHub, como era feito anteriormente, irá receber a mensagem citada no início:

```
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git pull origin main
Username for 'https://github.com': camilafernanda
Password for 'https://camilafernanda@github.com':
remote: Support for password authentication was removed on August 13, 2021. Please use a personal access token instead.
remote: Please see https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/ for more information.
fatal: Authentication failed for 'https://github.com/camilafernanda/teste-forum.git/'
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$
```

Dessa forma, basta você realizar o comando novamente e no lugar da senha, colar o seu token gerado:

```
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git pull origin main
Username for 'https://github.com': camilafernanda
Password for 'https://camilafernanda@github.com':
remote: Support for password authentication was removed on August 13, 2021. Please use a personal access token instead.
remote: Please see https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/ for more information.
fatal: Authentication failed for 'https://github.com/camilafernanda/teste-forum.git/'
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$
```



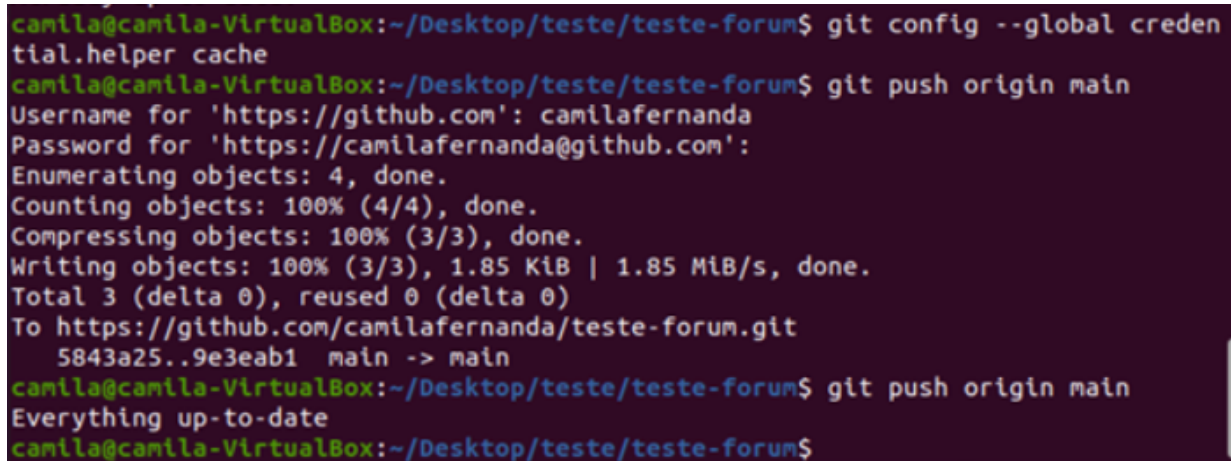
Assim, o comando irá ser realizado com sucesso:

```
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git pull origin main
Username for 'https://github.com': camilafernanda
Password for 'https://camilafernanda@github.com':
From https://github.com/camilafernanda/teste-forum
 * branch      main      -> FETCH_HEAD
Already up to date.
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$
```

Porém, apesar de estar funcionando, não é viável ter que colar o token toda vez que for fazer esses comandos, certo? Para isso, basta escrever o seguinte comando para que a última senha utilizada fique salva:

```
git config --global credential.helper cache
```

Assim, no próximo comando feito, o Git irá solicitar autenticação novamente, porém nos seguintes comandos, o Git não irá pedir seu usuário e senha novamente:

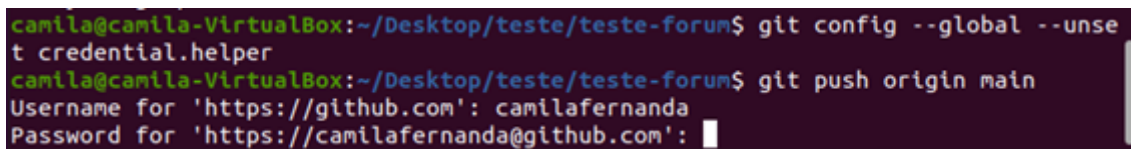


```
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git config --global creden
tial.helper cache
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git push origin main
Username for 'https://github.com': camilafernanda
Password for 'https://camilafernanda@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.85 KiB | 1.85 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/camilafernanda/teste-forum.git
 5843a25..9e3eab1  main -> main
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git push origin main
Everything up-to-date
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$
```

E por fim, caso você queira tirar o token das configurações, basta fazer:

```
git config --global --unset credential.helper
```

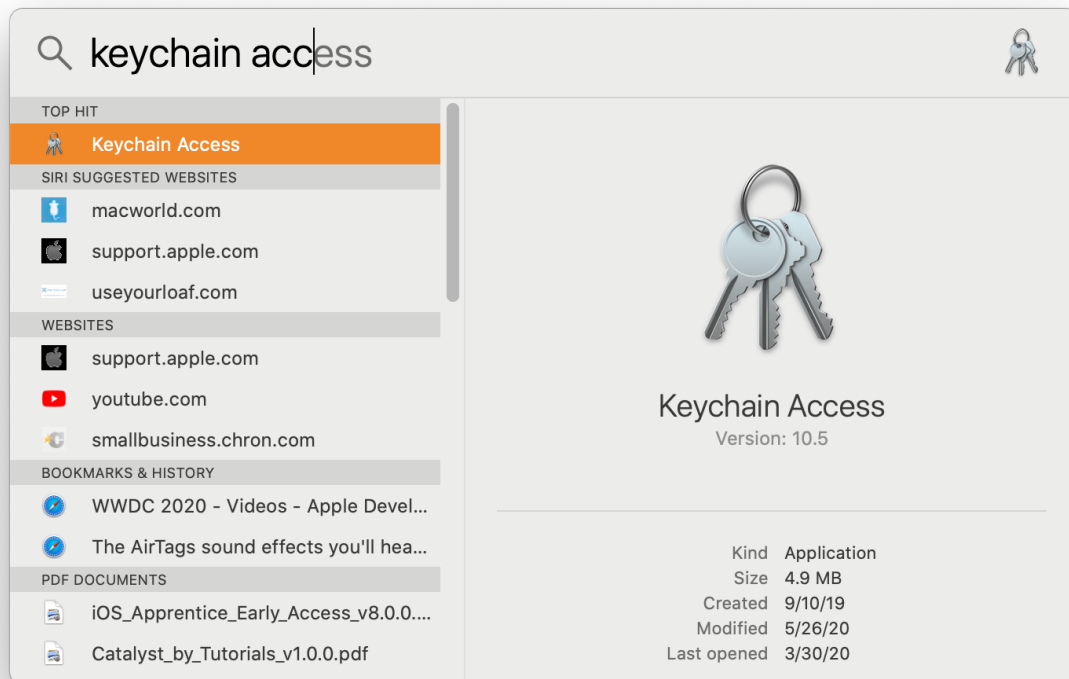
Assim, você terá os pedidos de requisições novamente:



```
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git config --global --unse
t credential.helper
camila@camila-VirtualBox:~/Desktop/teste/teste-forum$ git push origin main
Username for 'https://github.com': camilafernanda
Password for 'https://camilafernanda@github.com':
```

Como editar credenciais no MacOS

Para setar o token no Mac, é semelhante ao processo do Windows. Acesse **Keychain access**, para encontrar você pode clicar no ícone da lupa no canto direito da barra de menu.



Fonte: Imagem retirada da [Documentação do GitHub](#)

Em seguida, procure por **github.com**, clique em **internet password**, edite e cole o token pessoal gerado.

Caso queira retirar ele, no terminal digite:

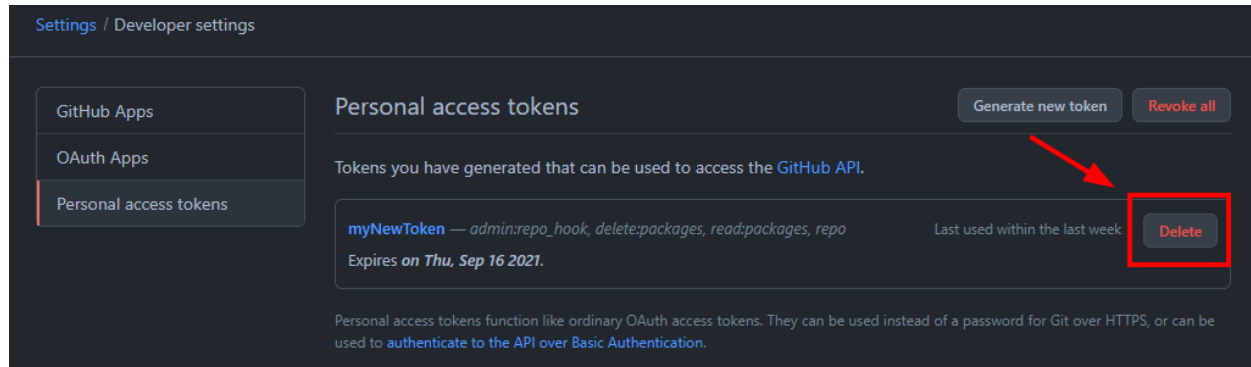
```
git credential-osxkeychain erase
```

Conclusão

Em suma, essa mudança de utilização de tokens para autenticação é bastante interessante, pois ele pode lhe oferecer um controle maior e mais segurança. Você pode fazer todo esse processo citado acima quantas vezes achar necessário, bem como configurar as permissões e o tempo de expiração de cada token gerado da maneira de achar melhor.

Em casos de falhas em segurança, por exemplo, a pessoa estará limitada às permissões

Em casos de falhas em segurança, por exemplo, a pessoa estará limitada às permissões dadas ao token e sua conta GitHub continua salva. Além disso, caso isso ocorra, apague seu token indo no GitHub, em seguida em Settings, Developer settings e Personal access tokens, mesmo passo a passo para gerar o token, porém clique em **Delete**:



Assim, você estará seguro novamente.

Caso você tenha curiosidade e queira entender um pouquinho mais sobre o token e outros tipos de autenticação, temos esse artigo que você possa se interessar:

[Tipos de Autenticação: Senha, Token, JWT, Dois Fatores e Mais](#)

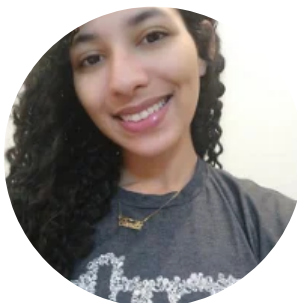
E para aprender mais sobre autenticação e segurança, confira:

[Podcast sobre Segurança](#)

[O que é Json Web Token \(JWT\)?](#)

[Curso Node.js e JWT: autenticação com tokens na Alura](#)

[Curso Autenticação mais segura com 2 fatores na Alura](#)



Camila Fernanda Alves

Sou quase engenheira pela Universidade Federal de Uberlândia e instrutora aqui na Alura. Atuei como estagiária do Scuba Team e já realizei algumas monitorias de Física, Circuitos Elétricos e até mesmo Processamento de Sinais. Atualmente, estou iniciando na produção de conteúdos relacionados a DevOps e também estou aqui para mostrar as maravilhas da tecnologia com foco em

Nova exigência do Git de autenticação por token, o que é e o que devo fazer? | Alura
relacionados a DevOps e também estou aqui para mergulhar em tecnologia com vocês e,
sempre que possível, tentar ajudá-los a encontrar as respostas que precisam nos fóruns. No
tempo livre, gosto de jogos da Nintendo e Valorant.

Veja outros artigos sobre
[DevOps](#)

Quer mergulhar em tecnologia e aprendizagem?

Receba a newsletter que o nosso CEO escreve pessoalmente, com insights do
mercado de trabalho, ciência e desenvolvimento de software

Escreva seu email

ME INSCREVA

Nossas redes e apps



Sobre nós

Como Funciona

Trabalhe conosco

Todos os cursos

Para Empresas

Depoimentos

Para Escolas

Instrutores(as)

Política de Privacidade

Dev em <T>

Termos de Uso

Status

Conteúdos

Fale Conosco

Alura Cases

Email e telefone

Imersões

Perguntas frequentes

Artigos

Falar pelo WhatsApp

Podcasts

Novidades e Lançamentos

Seu e-mail

RECEBER

CURSOS

Cursos de Programação

Lógica | Python | PHP | Java | .NET | Node JS | C | Computação | Jogos | IoT

Cursos de Front-end

HTML, CSS | React | Angular | JavaScript | jQuery

Cursos de Data Science

Ciência de dados | BI | SQL e Banco de Dados | Excel | Machine Learning | NoSQL | Estatística

Cursos de DevOps

AWS | Azure | Docker | Segurança | IaC | Linux

Cursos de UX & Design

Photoshop e Illustrator | Usabilidade e UX | Vídeo e Motion | 3D

Cursos de Mobile

React Native | Flutter | iOS e Swift | Android, Kotlin | Jogos

Cursos de Inovação & Gestão

Métodos Ágeis | Softskills | Liderança e Gestão | Startups | Vendas