



DIO - Curso GIT e Github

Chave SSH e Token - como gerar chave no GITBASH

Usar os seguintes comandos: (não precisa estar em um diretório específico)

```
ssh-keygen -t ed25519 -C dai.koblitz@gmail.com
```

ed25519 é o tipo de criptografia da chave

(ou outro email que deve obrigatoriamente estar associado a uma conta do github)

ENTER

aqui é possível colocar uma senha (que não sera visível)

ENTER



```
otavio@perkles-desktop MINGW64 ~
$ ssh-keygen -t ed25519 -c otaviocha@gmail.com
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/Lucas/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Lucas/.ssh/id_ed25519
Your public key has been saved in /c/Users/Lucas/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:CGDr3Aa2UuoQJ9xs1kkTWzGwwJ0Aamh6xnbS9RcZUco otaviocha@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|@B%O=      OO.|
|+XO#       . +|
|O+%       . E |
|+* * O O   .  |
|. X = . S .   |
|+ +         .  |
+-----[SHA256]-----+
otavio@perkles-desktop MINGW64 ~
$
```

Para visualizar o local onde estão salvas as chaves basta entrar neste diretório:

```
cd /c/Users/daiko/.ssh
```

e listar os documentos:

```
ls
```

vai aparecer a chave privada e a publica:

```
id_ed25519 id_ed25519.pub
```

a chave que deve ser utilizada no github é sempre a chave publica.

para pegar a chave publica na sua máquina, digitar

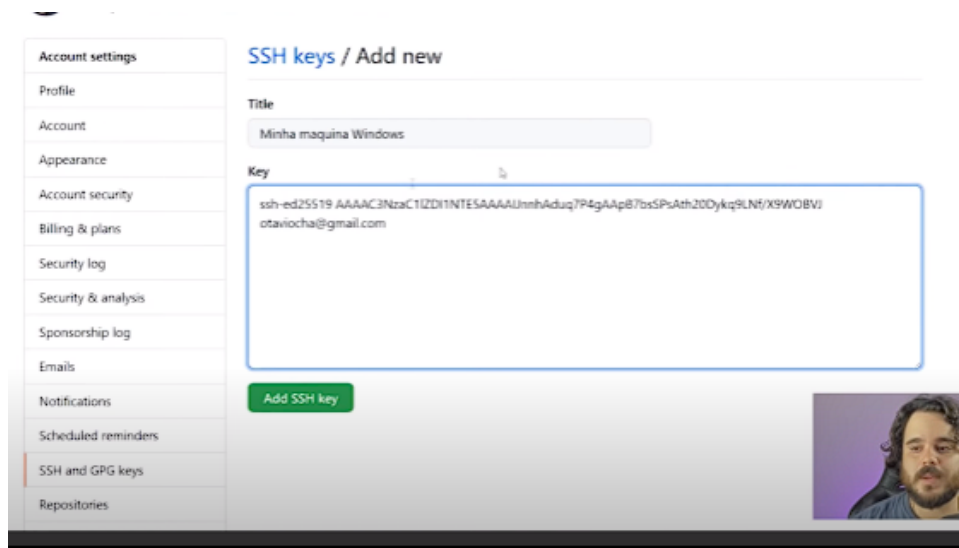
```
cat id_ed25519.pub
```

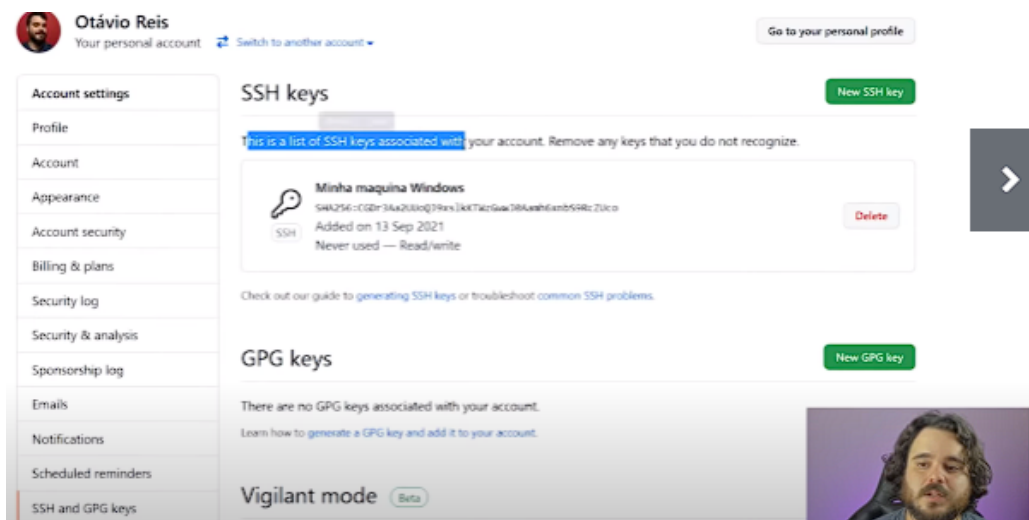
ENTER

irá retornar uma chave parecida com a da imagem:

```
otavio@perkles-desktop MINGW64 ~/.ssh
$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC11ZDI1NTE5AAAAIjnnhAduq7P4gAapB7bsSPsAth20Dykq9LNf/X9wOBVj
otaviocha@gmail.com
```

essa chave deve ser copiada para poder colar no github, na sessão SSH and GPG keys
também deve ser adicionado um nome a sua máquina para que você possa identificá-la





em seguida devemos gerar um agente e entregar a chave para ele (privada), e digitar a senha:

```
eval $(ssh-agent -s)
```

```
otavio@perkles-desktop MINGW64 ~/.ssh
$ eval $(ssh-agent -s)
Agent pid 1382

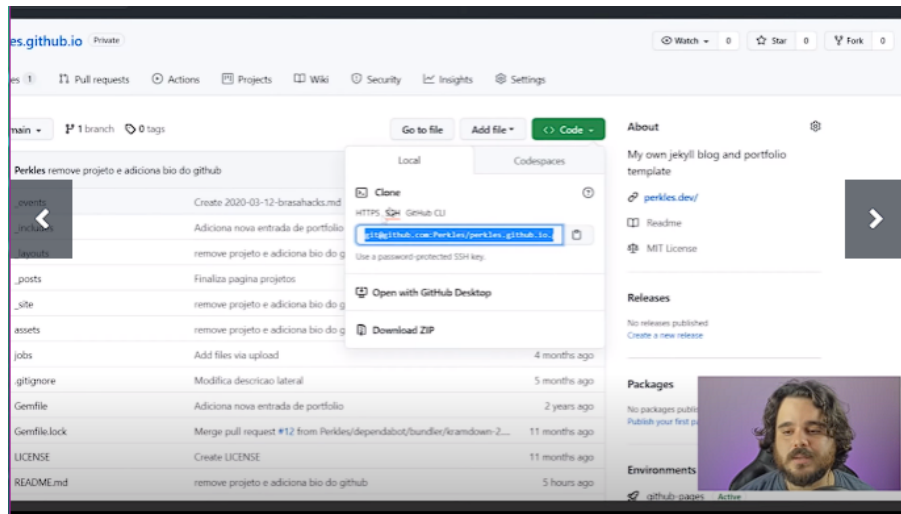
otavio@perkles-desktop MINGW64 ~/.ssh
$ ls
id_ed25519 id_ed25519.pub

otavio@perkles-desktop MINGW64 ~/.ssh
$ ssh-add id_ed25519
Enter passphrase for id_ed25519:
Identity added: id_ed25519 (otaviocha@gmail.com)

otavio@perkles-desktop MINGW64 ~/.ssh
$
```

Para clonar um repositório no Github

```
git clone git@github.com:ILadyLuckI/hello-world.git
```



e dar yes na pergunta no prompt

```
otavio@perkles-desktop MINGW64 /c/workspace/ssh-test
$ git clone git@github.com:Perkles/perkles.github.io.git
Cloning into 'perkles.github.io'...
The authenticity of host 'github.com (20.201.28.151)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXupJWG17E1IGOCspR0mTxDCARLviKw6E5SY8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (RSA) to the list of known hosts.
remote: Enumerating objects: 1278, done.
remote: Counting objects: 100% (62/62), done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 1278 (delta 24), reused 38 (delta 16), pack-reused 1216
Receiving objects: 100% (1278/1278), 21.64 MiB | 7.33 MiB/s, done.
Resolving deltas: 100% (580/580), done.
```

para conferir se o repositório foi clonado basta dar ls no diretório

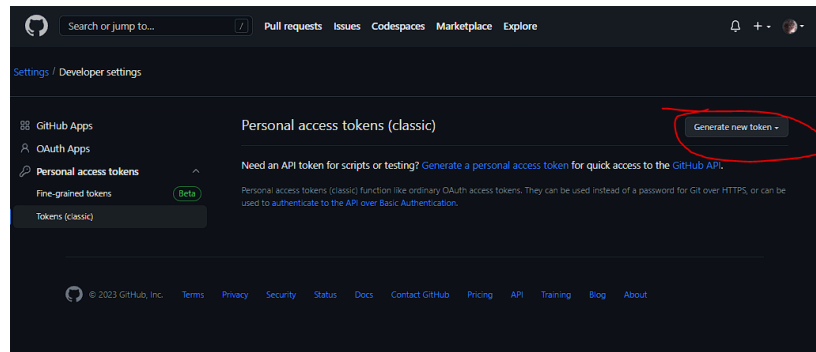
```
ls
```

```
daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/workspace/ssh-test
$ ls
hello-world/

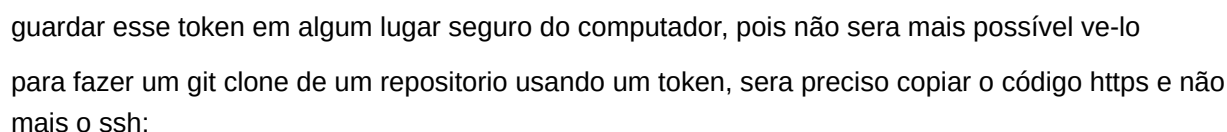
daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/workspace/ssh-test
$
```

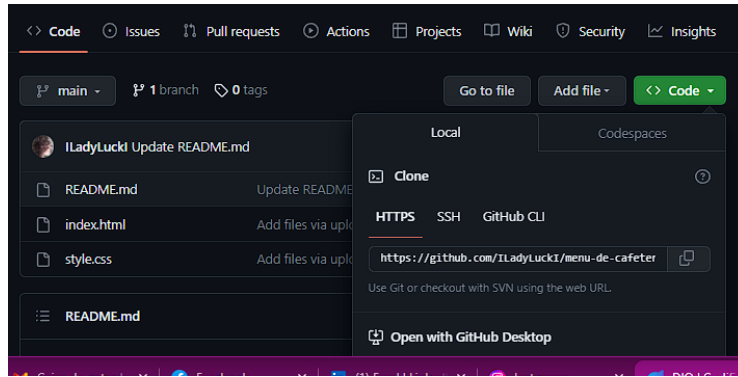
Token de acesso pessoal - 2ª forma de autenticação segura

Para configurar o token, deve ir no Github em “Developer settings”, token(classic) e “generate new token”



Marcar ou não uma data de expiração do token, dar um nome para o token e marcar a opção “Repo”





Iniciando o Git

para iniciar o Git e deve-se usar o comando

```
git init
```

em seguida configurar o nome de usuário e email que sera usado

Usando o Git pela primeira vez

O Git precisará de algumas configurações:

*email:

```
git config --global user.email "dai.koblitz@gmail.com"
```

*autor:

```
git config --global user.name Daiana
```

```
daiko@DESKTOP-56NDR40 MINGW64 /d/programacao/dio-cursos/workspace/livro-receitas (master)
$ git config --global user.email "dai.koblitz@gmail.com"

daiko@DESKTOP-56NDR40 MINGW64 /d/programacao/dio-cursos/workspace/livro-receitas (master)
$ git config --global user.name Daiana

daiko@DESKTOP-56NDR40 MINGW64 /d/programacao/dio-cursos/workspace/livro-receitas (master)
$ |
```

Adicionando um arquivo

No diretório onde se encontra o arquivo a ser adicionado, digite:

```
git add *
```

e em seguida:

```
git commit -m "commit inicial" (ou qualquer outra string)
```

```
daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/workspace
$ ls
livro-receitas/ ssh-test/ token-test/

daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/workspace
$ cd livro-receitas

daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/workspace/livro-receitas (master)
$ git add *

daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/workspace/livro-receitas (master)
$ git commit -m "commit inicial"
[master (root-commit) 8f0be27] commit inicial
1 file changed, 51 insertions(+)
create mode 100644 strogonoff.md

daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/workspace/livro-receitas (master)
$ |
```

Modificando o username e o email

usar o seguinte comando:

```
git config --global --unset user.email
```

```
git config --global --unset user.name
```

Adicionando um arquivo ao repositório remoto (GitHub)

primeiro deve se mostrar ao git o caminho do repositório remoto para onde deve ser empurrado o arquivo, usando o seguinte comando:

```
git remote add origin https://github.com/fulanodetal/pasta/arquivo.git
```

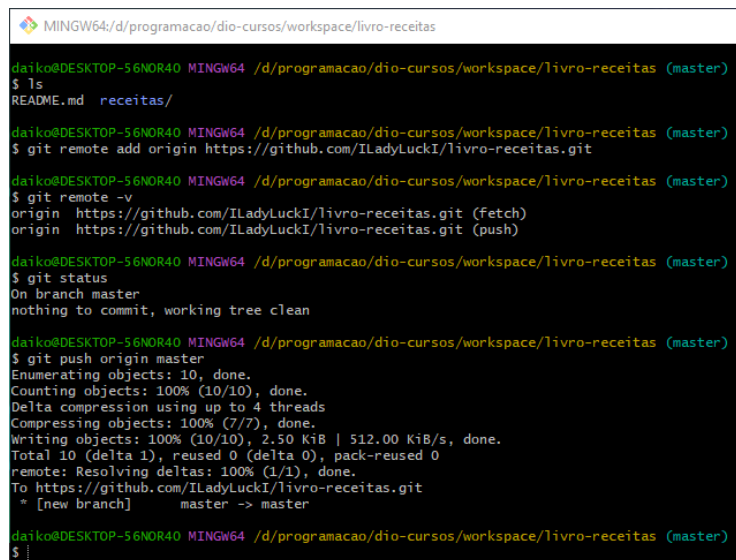

em seguida:

```
git remote -v
```

dar um git status para confirmar se esta tudo ok, em seguida empurrar o arquivo com o comando:

```
git push origin master
```

ficando dessa forma



```
MINGW64: d:/programacao/dio-cursos/workspace/livro-receitas
daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/workspace/livro-receitas (master)
$ ls
README.md  receitas/
daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/workspace/livro-receitas (master)
$ git remote add origin https://github.com/ILadyLuckI/livro-receitas.git
daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/workspace/livro-receitas (master)
$ git remote -v
origin https://github.com/ILadyLuckI/livro-receitas.git (fetch)
origin https://github.com/ILadyLuckI/livro-receitas.git (push)
daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/workspace/livro-receitas (master)
$ git status
On branch master
nothing to commit, working tree clean
daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/workspace/livro-receitas (master)
$ git push origin master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 2.50 KiB | 512.00 KiB/s, done.
Total 10 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/ILadyLuckI/livro-receitas.git
 * [new branch]      master -> master
daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/workspace/livro-receitas (master)
$ |
```

o comando “origin” é apenas um apelido para indicar de onde esta partindo o arquivo, poderia ser qualquer outro nome.

Commitando alterações no código

para commitar alterações no código e empurrar a nova versão para o GitHub, primeiro fazer as alterações necessárias no arquivo, salvar, em seguida usar o seguinte comando:

```
git status
```

o git status mostra que a edição ainda não esta “staged”, ou seja não esta preparada para o commit ainda, sendo necessario adicioná-la primeiro para que depois o commit seja adicionado

```
git add *
```

seguido de git status para confirmar se a edição esta “staged”

então fazer o commit:

```
git commit -m "mensagem a ser descrita"
```

feito esse processo, é so dar seguimento normalmente ao projeto, com o comando

```
git push origin master
```

 (em alguns casos o repositório remoto pode exigir uma autenticação)

```
daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/bootcamp (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/bootcamp (master)
$ git add *

daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/bootcamp (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/bootcamp (master)
$ git commit -m "Adiciona observações ao arquivo README"
[master 82a2321] Adiciona observações ao arquivo README
 1 file changed, 1 insertion(+), 1 deletion(-)

daiko@DESKTOP-56NOR40 MINGW64 /d/programacao/dio-cursos/bootcamp (master)
$ git push origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 721 bytes | 721.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/ILadyLuckI/desafio-dio-gitegithub.git
 1e71d4b..82a2321 master -> master
```

Resolvendo conflitos

caso o comando de git push seja rejeitado, o motivo pode ser que no repositório remoto esteja uma nova versão do código, sendo necessário fazer a edição do código para então empurra-lo de volta ao repositório remoto, neste caso é preciso puxar o arquivo do repositório remoto para o local, usando o comando

```
git pull origin master
```

Comandos

ls - exibe uma lista de arquivos dentro de um diretório

cd - muda de diretório, deve estar acompanhado do nome do diretório desejado, para voltar um nível, usar cd ..

pwd - exibe o caminho até o seu diretório atual

clear ou ctrl+I- limpa o prompt

cd .. - voltar um nível na hierarquia de pastas

mkdir - cria um novo diretório

-a - mostra arquivos ocultos (usar ls -a)

-m - passa uma mensagem