

# Introdução Git e GitHub

☀ Status	Done
▼ Assign	Oswaldo Reis Perkles
☰ Principais Conhecimentos	Git   GuiHub
📅 Data	@June 21, 2022 8:00 PM

## ▼ Aula 01: Introdução ao Git

### Entendendo o que é o Git e sua importância

Git é um sistema de versionamento de código distribuído criado em 2005 por Linus Torvalds. Foi criado quando ele trabalhava em um projecto grande com muitas outras pessoas, então ele fez isso para trabalhar no Linux.



Software não é feito se não for feito de forma colaborativa.

Esse curso permite aprender e ganhar as seguintes skills:

1. Controle de versão
2. Armazenamento em Nuvem
3. Trabalho em equipe
4. Melhorar o código
5. Reconhecimento

## ▼ Aula 02: Navegação via Command Line Interface e instalação

### Comando básicos para um bom desempenho no terminal

O Git é usado totalmente no terminal, ou seja é uma aplicação que funciona com o CLI e não a GUI.

### Comandos básicos no cmd

`dir` : lista directórios

`cd` : mudar directórios

`cd ..` : voltar um nível

`cls` : limpar a tela

`TAB` : autocompleta

`mkdir` : criar directórios

`del + [nome-do-directório]` : Deleta o conteúdo do directório

`rmdir + [nome-do-directório] /s /q` : elimina o directório

## ▼ Aula 03: Entendendo como Git funciona por baixo dos panos

### Tópicos fundamentais para entender o funcionamento do Git

#### SHA1

Secure Hash Algorithm é um algoritmo de encriptação criado pela NSA e a sua saída gera um conjunto de caractere único de 40 dígitos.

#### Objectos internos do Git

- Blobs
- Trees
- Commits

### Chaves SSH e Token

A autenticação padrão tem se tornado obsoleta, por isso foram criadas novas estratégias mais seguras para se autenticar.

#### Chave SSH

Forma de estabelecer uma conexão segura entre duas máquinas aonde há sempre uma chave pública e uma chave privada.

Para estabelecer a conexão entre o Github e uma máquina é preciso seguir os seguintes passos:

#### 1. Gerar as chaves publicas e privadas:

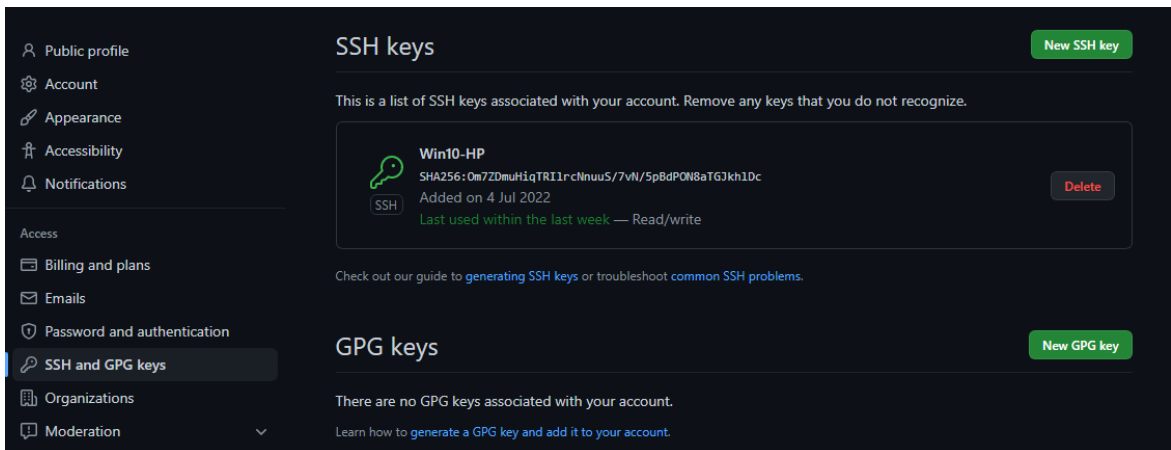
`ssh-keygen -t ed25519 -C + [email-do-github]` teremos uma saída semelhante a seguir:

```
HP@HP-HP MINGW64 ~
$ ssh-keygen -t ed25519 -C ida.infoseg@gmail.com
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/HP/.ssh/id_ed25519):
Created directory '/c/Users/HP/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/HP/.ssh/id_ed25519
Your public key has been saved in /c/Users/HP/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:0m7ZDmuHiqTRIrcNnuuS/7vN/5pBdPON8aTGJkh1Dc ida.infoseg@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|      . . .      |
|      o E        |
|      o.=        |
|      o+.        |
|      S  =+ .    |
| o . . . .+*+    |
|+ = = ++      ...o|
|o* = +*+o o ..  |
|o . *X=+o.+o    |
+-----[SHA256]-----+
```

No directório gerado, estarão as chaves públicas e privadas.

#### 2. Adicionar chave publica gerada no GitHub

- Acessar a Definições do GitHub e de seguida a opção SSH and GPG Keys



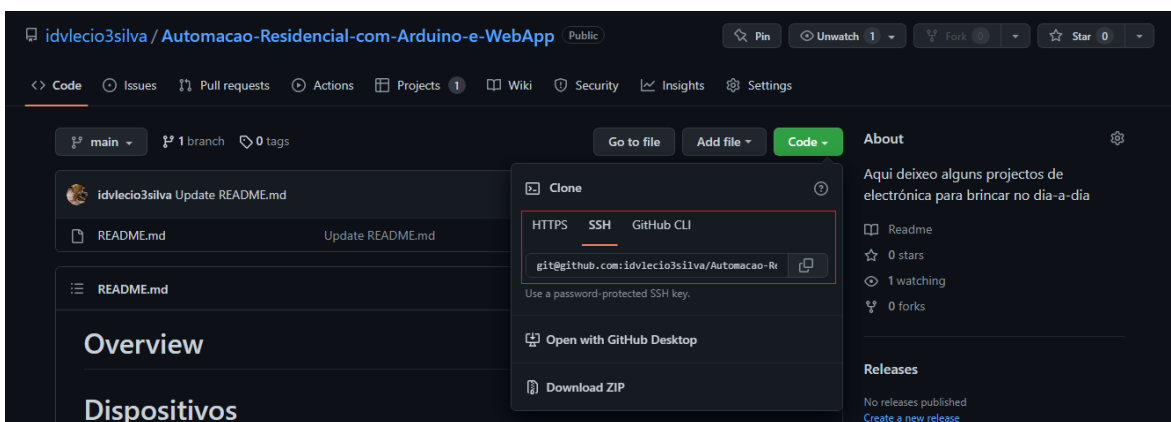
### 3. Passar a chave para o agent-ssh

`eval $(ssh-agent -)` : Esse comando inicia o ssh-agent. Para passar a chave criada usa-se o comando: `ssh-add id-ed25519`. Isso terá uma saída parecida:

```
HP@HP-HP MINGW64 ~/.ssh
$ ssh-add id_ed25519
Enter passphrase for id_ed25519:
Identity added: id_ed25519 (ida.infoseg@gmail.com)
```

### 4. Teste: clonar um repositório

- Copiar o link SSH de um repositório no GitHub



- Clonar o repositório

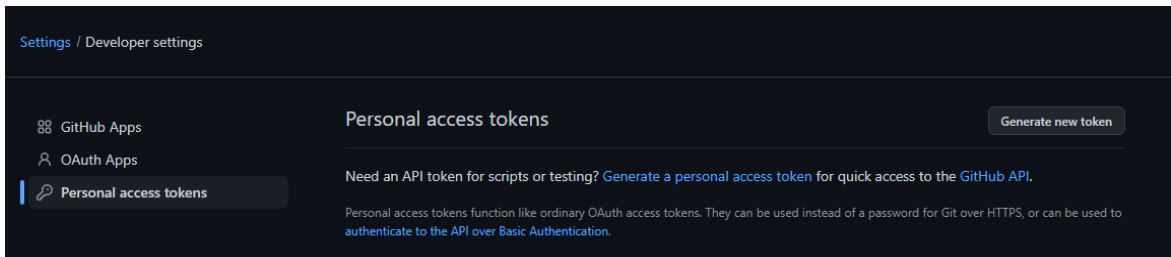
```
HP@HP-HP MINGW64 ~/Workspace/ssh-test
$ git clone git@github.com:idvlecio3silva/Automacao-Residencial-com-Arduino-e-WebApp.git
Cloning into 'Automacao-Residencial-com-Arduino-e-WebApp'...
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 31 (delta 6), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (31/31), 9.10 KiB | 847.00 KiB/s, done.
Resolving deltas: 100% (6/6), done.
```

## Token de Acesso Pessoal

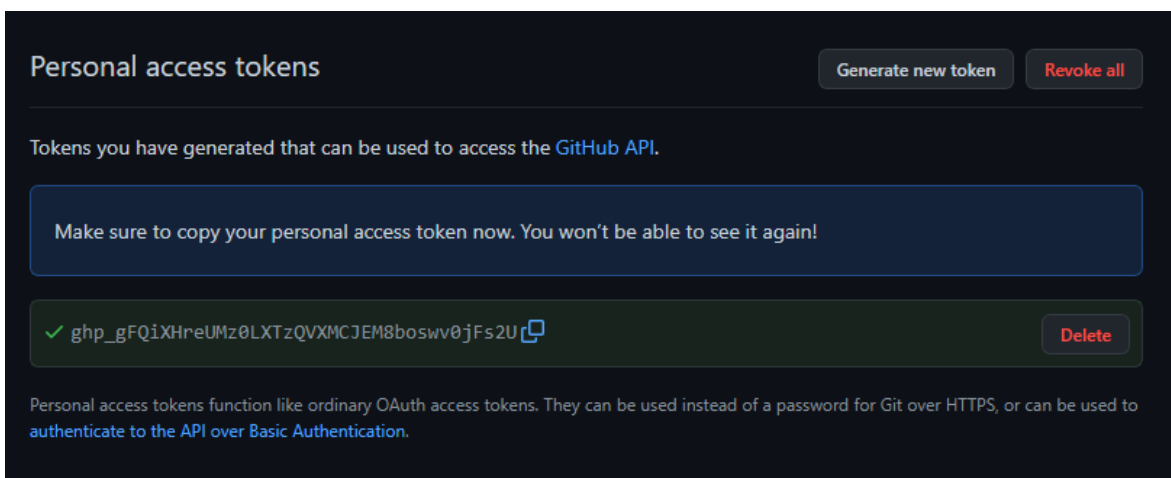
Gera um token para sempre que quiser fazer um commit utilizar o token ao invés da password.

### 1. Gerar o Token

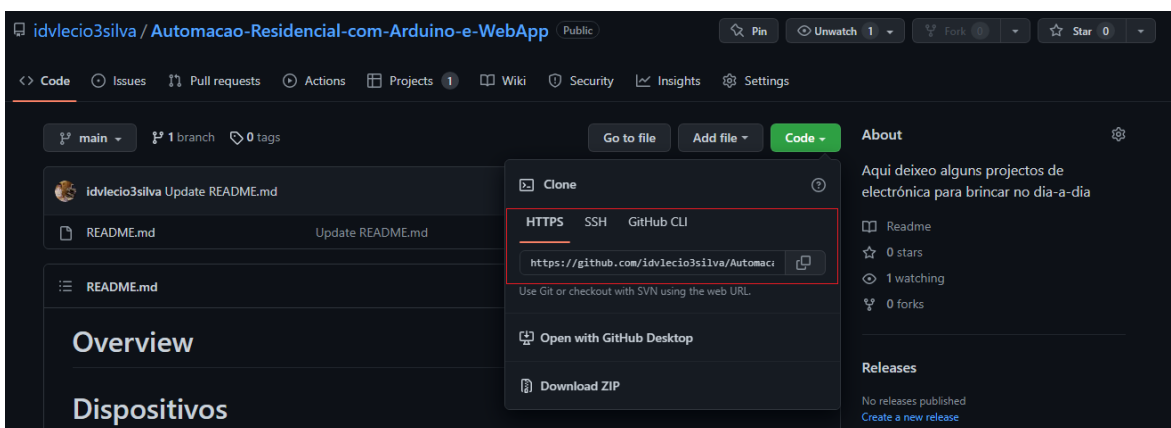
- Developer Setting no GitHub
- Personal Acesso Tokens
  - Generate New Token



Token Gerado: ghp\_gFQixHreUMz0LXTzQVXMCJEM8boswv0jFs2U



## 2. Clonar um repositório



**OBS:** quando se usa o Token para a autenticação, deve-se usar o HTTPS ou o link da página.

```
HP@HP-HP MINGW64 ~/Workspace/token-test
$ git clone https://github.com/idvlecio3silva/Automacao-Residencial-com-Arduino-e-WebApp.git
Cloning into 'Automacao-Residencial-com-Arduino-e-WebApp'...
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 31 (delta 6), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (31/31), 9.10 KiB | 2.28 MiB/s, done.
Resolving deltas: 100% (6/6), done.
```

## ▼ Aula 04: Primeiros comandos com Git

### Iniciando Git e criando um Commit

Iniciar o repositório do git: `git init`

Mover arquivo e dar início: `git add`

Criar commit: `git commit`

### Configurando o directório de trabalho

Quando usamos o git pela primeira vez, precisamos fazer algumas configurações.

```
git config --global user.email + "email"
```

```
git config --global user.name + [Nome-de-usuário]
```

### Adicionar um arquivo no directório

```
git add *
```

```
git commit -m "commit inicial"
```

```
HP@HP-HP MINGW64 ~/Workspace/livro-receitas (master)
$ git commit -m "commit inicial"
[master (root-commit) 6d1a278] commit inicial
1 file changed, 20 insertions(+)
create mode 100644 Strogonoff de Frango.md.txt
```

## ▼ Aula 05: Ciclo de vida dos arquivos no Git

### Passo-a-passo no ciclo de vida

git init: inicia o repositório

O ciclo de vida dos de um arquivo no Git tem duas formas: Tracked e Untracked

Untracked: o git não sabe da existência do arquivo

### Tracked

Tracked está subdividido em 3 partes:

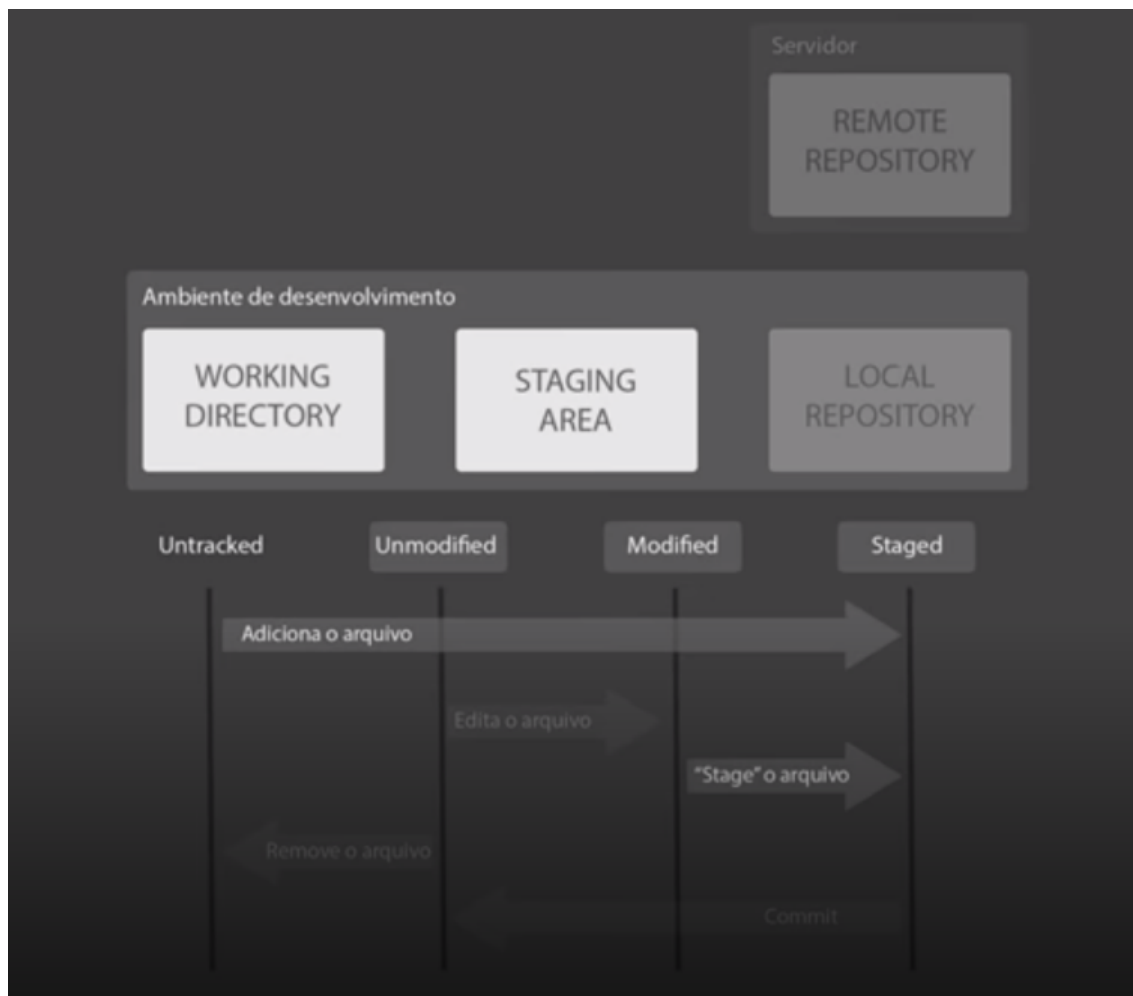
- Unmodified: arquivo não modificado. Depois do fazer alguma modificação, para o estado modified. Nesse processo, o Git compara o SHA1.
- Modified: arquivo modificado
- Staged: Local aonde ficam armazenados arquivos que seram usados noutros tipos de agrupamentos.



**OBS:** Depois de um arquivo ser eliminado ele volta para o estado untracked

No processo de desenvolvimento existe a separação de dois ambientes - ambiente de desenvolvimento que a máquina onde trabalha e a versão no repositório remoto que é o servidor.

Os arquivos ficam sempre se alterando entre o Working Directory e o Staging Area. Quando é feito um commit, o arquivo é integrado ao repositório local que pode ser empurrado para o Repositório Remoto ou no Servidor.



`git status`: permite ver o estado do arquivo

```
HP@HP-HP MINGW64 ~/Workspace/livro-receitas (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:   Strogonoff de Frango.md.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Strogonoff.md

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

Para criar um commit usa-se o comando: `git add *` Esse comando envia os arquivos para o estado Staged.

```

HP@HP-HP MINGW64 ~/Workspace/livro-receitas (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    Strogonoff de Frango.md.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README
        receitas/

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

HP@HP-HP MINGW64 ~/Workspace/livro-receitas (master)
$ git add *
warning: in the working copy of 'README', LF will be replaced by CRLF the next time Git touches it

HP@HP-HP MINGW64 ~/Workspace/livro-receitas (master)
$ ls
README  receitas/

HP@HP-HP MINGW64 ~/Workspace/livro-receitas (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   README
        new file:   receitas/Strogonoff.md

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    Strogonoff de Frango.md.txt

HP@HP-HP MINGW64 ~/Workspace/livro-receitas (master)
$ git commit -m "adicionar index"
[master 4b1d861] adicionar index
 2 files changed, 26 insertions(+)
 create mode 100644 README
 create mode 100644 receitas/Strogonoff.md

```

## ▼ Aula 06: Introdução ao GitHub

### Trabalhando com GitHub

Colocar um arquivo no Repositório Remoto (GitHub): `git remote add origin + [link-https-do-repositório no GitHub]`

```

HP@HP-HP MINGW64 ~/Workspace/livro-receitas (master)
$ git remote add origin https://github.com/idvlecio3silva/livro-receitas.git

```

Listar os repositórios no Repositório Remoto: `git remote -v`

```

HP@HP-HP MINGW64 ~/Workspace/livro-receitas (master)
$ git remote -v
origin https://github.com/idvlecio3silva/livro-receitas.git (fetch)
origin https://github.com/idvlecio3silva/livro-receitas.git (push)

```

Empurrar o arquivo do Repositório Local para o Repositório Remoto: `git push origin master`

```

HP@HP-HP MINGW64 ~/Workspace/livro-receitas (master)
$ git push origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 947 bytes | 47.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/idvlecio3silva/livro-receitas.git
 * [new branch]      master -> master

```

## ▼ Aula 07: Resolvendo conflitos

Como os conflitos acontecem no GitHub e como resolvê-los