Tópicos Fundamentais para Entender o Funcionamento do Git

O Git faz o controle de versões dos códiggos, salvando uma alteração a cada mudança de arquivo original. O SHA1 é uma função de has criptográfica, serve para criptografar e identificar **(etiquetar)** arquivos/mudanças/operações. Inicialmente foi criado pela Agência Nacional de Segurança dos Estados Unidos, sendo protocolo de comunicação padrão. Confira no exemplo abaixo:

```
Gabriel_Victorino@DESKTOP-A192CLP MINGW64 ~//eekkop
 $ openssl sha1 texto. txt
SHA1(texto.txt)= 0a4d55a8d778e5022fab701977c5d840bbc486d0
Gabriel_Victorino@DESKTOP-A192CLP MINGW64 ~/Desktop
 $ openssl sha1 texto. txt
 SHA1 (texto. txt)= 0a4d55a8d778e5022fab701977c5d840bbc486d0
```

Objetos Internos

```
Gabriel_Victorino@DESKTOP-A192CLP MINGW64 ~/Desktop
$ echo -e 'conteudo' | openssl sha1
(stdin)= 65b0d0dda479cc03cce59528e28961e498155f5c

Gabriel_Victorino@DESKTOP-A192CLP MINGW64 ~/Desktop
$ echo 'conteudo' | git hash-object --stdin
fc31e91b26cf85a55e072476de7f263c89260eb1

Gabriel_Victorino@DESKTOP-A192CLP MINGW64 ~/Desktop
$ echo -e 'blob 9\0conteudo' | openssl sha1
(stdin)= fc31e91b26cf85a55e072476de7f263c89260eb1
```

Chaves SSH e Tokens

Uma chave ssh permite uma conexão que não requere senha para carregar arquivos e conecta o computador ao GitHub. Desse modo, são criadas duas chaves (pública e privada), tudo encriptado.

> É necessário possuir uma Conta do Github e o Gitbash.

Digite o código abaixo para criar uma senha e ativar a chave ssh.

```
Gabriel_Victorino@DESKTOP-A192CLP MINGW64 ~
$ ssh-keygen -t ed25519 -C gabriel.victorino2004@gmail.com
```

Ex.: Senha -> gabriel2004

O output esperado é o seguinte:

```
Gabriel_Victorino@DESKTOP-A192CLP MINGW64 ~
$ ssh-keygen -t ed25519 -C gabriel.victorino2004@gmail.com
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/Gabriel_Victorino/.ssh/id_ed25519):
gabriel2004
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in gabriel2004
Your public key has been saved in gabriel2004.pub
The key fingerprint is:
SHA256:9xO88dKRx4Qq1nLeMRH8DJfK0f9VA9FoPb/jSS82SIc gabriel.victorino2004@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|             +O .|
|             +oXo|
|            o ==B|
|           o + =*|
|        S = B.= *|
|         o *EB.O.|
|           .*o* +|
|            .o+o.|
|             . o |
+----[SHA256]-----+
```

Desse modo, as chaves foram salvas nos arquivos : gabriel2004 **(chave privada)** e gabriel2004.pub **(chave publica.)**

Abaixo é possível exibir o sha chave.

```
Gabriel_Victorino@DESKTOP-A192CLP MINGW64 ~
$ cat gabriel2004.pub
```

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGsMDPId9GQa6WC/tUpSe0CWoku1WlFzltaAVn5wKh0Q
gabriel.victorino2004@gmail.com
```

No github, vamos em Settings -> SSH e GPG Keys -> Adicionar SSH Keys. Assim, adicionamos as **chaves públicas.** Por fim, vamos ativar o processo ssh-agent para gerenciar as chaves.

```
Gabriel_Victorino@DESKTOP-A192CLP MINGW64 ~
$ eval $(ssh-agent -s)
Agent pid 1665 //Numero do Processo
```

Por fim, passamos a chave privada para conferir com a chave pública e autorizar a conexão.

```
Gabriel_Victorino@DESKTOP-A192CLP MINGW64 ~
$ ssh-add gabriel2004

//Adiciona a chave.
Gabriel_Victorino@DESKTOP-A192CLP MINGW64 ~
$ ssh-add gabriel2004
Enter passphrase for gabriel2004:
Identity added: gabriel2004 (gabriel.victorino2004@gmail.com)
```

Por fim, quando quisermos clonar um repositório privado, usaremos a chave ssh.