

# SEGURIDAD DE LA INFORMACIÓN

## ALUMNO

López Bautista Alfredo

1319104653

1922IS

- SSH GIT a GITHUB

El primer paso a realizar es identificar la versión instalada de GIT y configurar el usuario y correo que se utilizará

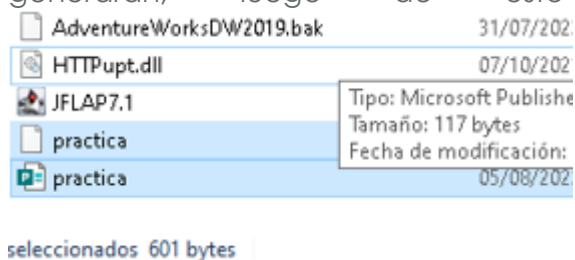
```
mrsrw@DESKTOP-JEMRQKP MINGW64 ~/Documents
$ git --version
git version 2.36.1.windows.1

mrsrw@DESKTOP-JEMRQKP MINGW64 ~/Documents
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Alfred-in
user.email=alfredo_1319104653@uptecamac.edu.mx
```

Posterior a la configuración básica se procede a generar la ssh key

```
mrsrw@DESKTOP-JEMRQKP MINGW64 ~/Documents
$ ssh-keygen -t ed25519 -C "alfredo_1319104653@uptecamac.edu.mx"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c:/Users/mrsrw/.ssh/id_ed25519): practica
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in practica
Your public key has been saved in practica.pub
The key fingerprint is:
SHA256:ECsY15UIcYg82+u7Tj0Au1UIMakTXJufJHq/nApMaoY alfredo_1319104653@uptecamac.edu.mx
The key's randomart image is:
+--[ED25519 256]--+
|O+OO+=OO..|
|o++++.OO.|
|. .B..O|
|OO.=...|
|.OO..+ S|
|=.OO|
|E=..+|
|OO..OOO|
|. O=*|
+-----[SHA256]-----+
```

En esta se le brinda un correo y un passphrase así como el nombre de los archivos que se generarán, luego de esto ya tendremos los archivos creados



Y en caso de ser necesario hay que moverlos a la carpeta .ssh en el disco local

```

mrsrw@DESKTOP-JEMRQKP MINGW64 ~/Documents
$ mkdir .ssh

mrsrw@DESKTOP-JEMRQKP MINGW64 ~/Documents
$ mv practica .ssh

mrsrw@DESKTOP-JEMRQKP MINGW64 ~/Documents
$ mv practica.pub .ssh

mrsrw@DESKTOP-JEMRQKP MINGW64 ~/Documents
$ ls .ssh
practica practica.pub

mrsrw@DESKTOP-JEMRQKP MINGW64 ~/Documents
$

```

Seguido de esto se debe proceder a copiar la llave publica generada, en el archivo practica.pub por lo que se ejecuta el siguiente comando:

```

mrsrw@DESKTOP-JEMRQKP MINGW64 ~/Documents
$ clip < ~/.ssh/practica.pub

```

Y con la llave en el portapapeles se debe configurar en nuestra cuenta de github, en la opción de keys en el apartado de settings



**Alfred-in**

Your personal account

- Public profile
- Account
- Appearance
- Accessibility

## SSH keys

New SSH key

There are no SSH keys associated with your account.

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

Se le da un nombre y se pega la llave generada

## SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



**practica**

SHA256:ECsY15UIcYg82+u7Tj0Au1UIHakTXJufJHq/nApMaoY

SSH

Added on 5 Aug 2022

Never used — Read/write

Delete

Y con esto ya esta configurada la llave que se utilizará en la comunicación de git a github. Ahora se necesita comprobar la funcionalidad y realizar la comunicación con un repositorio. Para esto se debe de ejecutar el agente ssh

```

mrsrw@DESKTOP-JEMRQKP MINGW64 ~/.ssh
$ eval "$(ssh-agent -s)"
Agent pid 1471

mrsrw@DESKTOP-JEMRQKP MINGW64 ~/.ssh
$ ssh-add ~/.ssh/prueba
Identity added: /c/Users/mrsrw/.ssh/prueba (alfredo_1319104653@uptecamac.edu.mx)

```

Y agregar el archivo de la clave privada, una vez verificado esto se puede realizar la prueba de funcionamiento con el comando

```

mrsrw@DESKTOP-JEMRQKP MINGW64 ~/.ssh
$ ssh -T git@github.com
Hi Alfred-in! You've successfully authenticated, but GitHub does not provide shell access.

```

Y como resultado vemos que se autenticó correctamente

```
mrsw@DESKTOP-JEMRQKP MINGW64 ~/.ssh
$ git clone git@github.com:Alfred-in/evaluacionU3.git
Cloning into 'evaluacionU3'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

Y finalmente se clona el repositorio donde trabajamos.