CLAVES SSH

INDICE DOCUMENTO

GENERAR CLAVES SSH

GENERAR UNA NUEVA CLAVE SSH

ssh-keygen

```
Generating public/private rsa key pair.
Enter file in which to save the key (/{usuario}/.ssh/id_rsa):
```

Presionamos ENTER y nos guardara las claves en el directorio .ssh/ dentro de la carpeta home de nuestro usuario.
Si previamente ya teniamos generadas unas claves veremos lo siguiente:

```
/home/{usuario}/.ssh/id_rsa already exists.
Overwrite (y/n)?
```

Si sobreescribimos las claves no podremos usar las que ya teniamos generadas.
Despues deberiamos ver el siguiente mensaje:

```
Enter passphrase (enter for no passphrase):
Si tenemos una passphrase añadiremos una capa mas de seguridad para evitar accesos
no autorizados.
```

Despues de añadir la passphrase veremos algo similar a esto:

```
Your public key has been saved in /home/{usuario}/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:qWAy5xagI+gZ5OzFNCWnptGr+ETSmcoxfafEslyJVwk {user}@{user}
The key's randomart image is:
+---[RSA 3072]----+
|    . E          |
|   . = . .       |
|  o.*   o        |
| =oB+= o  .      |
| *BBB=B .S       |
| *=O*Boo.        |
| o*.oo..         |
|  o .            |
|   .             |
```

```
+----[SHA256]-----+
```

Ahora si vamos al directorio

`/home/{usuario}/.ssh/`

veremos lo siguiente:

`id_rsa.pub`	→	Clave pública
`id_rsa`	→	Clave privada

COPIAR PUBLIC-KEY A OTRO EQUIPO *Y CONECTARSE DESDE ÉL*

Web Medium .Com

Si queremos conectarnos a otro equipo usando el par de claves que hemos generado debemos agregar la clave pública al archivo authorized_keysdel equipo al que nos queremos conectar.

Para ello, primero, necesitamos enviar la clave pública al equipo remoto, esto lo podemos hacer por medio de correo electrónico, FTP, SSH o podemos usar ssh-copy-id.

COPIAR CLAVE PÚBLICA EN EQUIPO REMOTO

1. **Asegurarnos que existe el Directorio .ssh/ en la carpeta Home del Usuario**

```
mkdir -p ~/.ssh
```

2. **Copiar la clave pública al archivo `authorized_keys`**

```
echo "$(cat ~/.ssh/id_rsa.pub)" >> ~/.ssh/authorized_keys
```

cambiar ~/.ssh/id_rsa.pub por el directorio donde este tu clave pública

CONECTARSE AL EQUIPO REMOTAMENTE:

Volver a nuestro equipo donde hemos generado las claves y conectarnos al equipo donde hemos copiado la clave pública.

```
ssh usuario@ip_equipo
```

La primera ve que nos conectemos al equipo veremos lo siguiente

```
Enter passprashe (empty for no passprashe):
```

Escribimos la contraseña que habiamos introducido al crear las claves. Si no habia introducido ninguna contraseña simplemente presionamos ENTER

ya tenemos una conexión SSH al equipo remoto usando claves SSH.

GIT-HUB / GIT-LAB

GIT-HUB

GENERAR CLAVE SSH PARA GITHUB

1. **Generamos las claves con el mail de la cuenta de GitHub:**

```
ssh-keygen -t ed25519 -c "dev.g.......@gmail.com"
```

```
Enter passphrase (empty for no passphrase): [Type a passphrase]
```
↵ *(enter para no crear una frase de seguridad)*

```
Enter same passphrase again: [Type passphrase again]
```
↵ *(enter para no crear una frase de seguridad)*

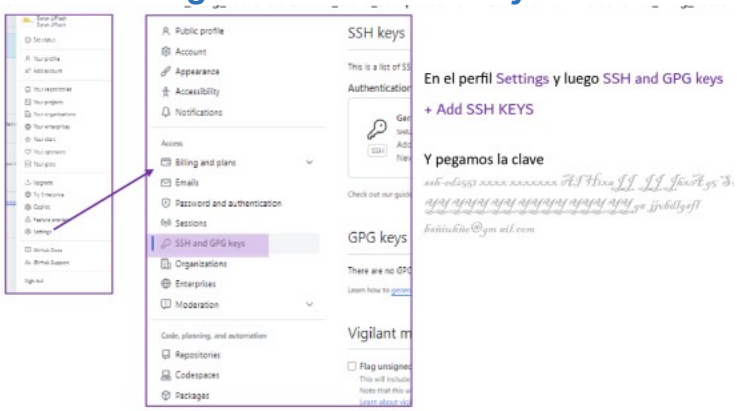
2. **Copiamos la clave pública en la cuenta de GitHub,**

```
cat ~/.ssh/id_ed25519.pub
```

```
ssh-ed25519 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXX
XXXXXXXXX XXXXXXX dev.g....@gmail.com
```

3. **Y luego en la Cta de GitHub:**

Perfil/settings/SSH and GPG Keys



DIFERENTES SSH-KEYS PARA DIFERENTES CUENTAS

GENERAR SSH-KEYS (1 POR CTA)

AJUSTES DE CONFIGURACIÓN SSH Y DE GIT:

- *(SSH)* *Generar una key SSH diferente para cada cuenta de GitHub*
- *(SSH)* *Crear un archivo de configuración SSH global*
- *(Git)* *Configurar el remote origin de cada repositorio*
- *(Git)* *Settear el name y el email para cada repositorio*

NAVEGAR AL REPOSITORIO

```
cd ~/Documents/work/git-tricks-and-tutorials
```

MOSTRAR EL REMOTE

```
git remote -vdiferentes ssh-keys para diferentes cuentas

  # origin    https://github.com/ChemaCLi/git-tricks-and-tutorials.git (fetch)
  # origin    https://github.com/ChemaCLi/git-tricks-and-tutorials.git (push)
```

GENERAR Y PREPARAR LAS NUEVAS KEYS SSH

Lo primero que debemos hacer es navegar a la carpeta .ssh de nuestra computadora.

Dentro del directorio .ssh encontrarás los archivos id_rsa y id_rsa.pub.

En mi caso estas son las keys que uso para mis proyectos personales, así que crearé una carpeta personal para colocarlas allí, y crearé una carpeta work para la cuenta del trabajo

CREA UNA CARPETA PERSONAL Y OTRA WORK (P.EJ) DENTRO DE /.ssh

```
mkdir -p ~/.ssh/personal
mkdir -p ~/.ssh/work
```

Meter las claves que tenemos con mail profesional en work

```
mv id_ed25519  id_ed25519.pub ~/.ssh/work/
```

GENERA LA NUEVA KEY DENTRO DE LA CARPETA ~/.ssh/personal

```
cd ~/.ssh/personal
```

```
ssh-keygen -t rsa -b 4096 -C "mi_cuenta@personal"
```

Al presionar Enter nos preguntará qué nombre le queremos dar al archivo.Usaré "personal" en mi caso.

```
Enter file in which to save the key (/Users/chema/.ssh/id_rsa):
```

```
personal
```

Presiona Enter un par de veces para continuar sin poner una contraseña, o especifica una si lo prefieres

Ahora tendrás tus keys ssh organizadas de esta manera:

```
.ssh/work/id_ed25519 + id_ed_25519.pub
.ssh/personal/personal + personal.pub
```

AGREGAR LA NUEVA SSH KEY EN TU CUENTA DE GITHUB DEL TRABAJO.

ARCHIVO CONFIG SSH

La clave y la magia de todo lo que estamos haciendo recae en gran parte sobre este paso, así que presta atención.

Crea un archivo de texto plano con el nombre config dentro de tu carpeta .ssh. Asegúrate de que no tenga la extensión .txt o cualquier otra. El nombre del archivo debe ser config únicamente.

Escribe este contenido, poniendo el nombre que hayas elegido

```
# Configuracion de GitHub para mi cuenta Personal
Host github.com-personal
HostName github.com
User git
IdentityFile ~/.ssh/personal/personal
# Configuracion de GitHub para mi trabajo
Host github.com-work
HostName github.com
User git
IdentityFile ~/.ssh/work/id_ed25519
```

Presta mucha atención a este detalle. Estamos usando un Host github.com-personal y un Host

github.com-work. El sufijo -personal y -work le servirá a Git para escoger qué keys usar más adelante.

Además, el IdentityFile de cada configuración debe coincidir con los nombres de las carpetas y keys que creamos.

CONFIGURAR EL REMOTE ORIGIN DEL REPOSITORIO

LA 1 VEZ AL HACER GIT CLONE

hay que modificar la Url para que el host quede como

github.com-work si es un repo de tu cuenta del trabajo,

github.com-personal si es tu repo de la cuenta personal.

```
git clone git@github.com-work:mi_repo.git
```

SI YA TIENES EL REPO CLONADO/DESCARGADO

Escribe en la terminal los siguientes comandos para navegar a tu repositorio, verificar la URL del remote origin, y modificarla para que utilice la configuración work.

Verifica el remote origin

```
cd ~/Documents/work/mi_repo/
git remote -v
```

```
# origin    git@github.com:mi_repo.git (fetch)
# origin    git@github.com:mi_repo.git (push)
```

Modificar la URL para que ahora use el host github.com-work

```
git remote set-url origin git@github.com-work:mi_repo.git
```

Ahora verifica la nueva configuración del remote origin de Git.

```
git remote -v
```

```
# origin    git@github.com-work: mi_repo.git (fetch)
# origin    git@github.com-work: mi_repo.git (push)
```

Y finalmente valida que puedas comunicarte con GitHub haciendo git pull:

```
git pull -ff
```

```
# Already up to date.
```

SETTEAR FIRMA DE GIT PARA EL REPOSITORIO ACTUAL

Último paso para detalles finos.

Vamos a hacer que el nombre y el email que aparecen en cada commit sea diferente para nuestro repositorio (en este caso el repo del trabajo)

settear tu firma para el repositorio del trabajo:

```
git config user.email chema@devu.community
git config user.name ChemaCL
```

Verifica la configuración con el comando

```
git config -list
```

```
# Esta configuración aparecerá hasta abajo
remote.origin.url=git@github.com-work:ChemaCLi/git-tricks-and-
tutorials.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
user.email=chema@devu.community
user.name=ChemaCLi
```

Configurar el remote origin los repositorios de la cuenta personal para que usen la ssh de la carpeta personal siguiendo el paso 3 de esta guía.

También recuerda que debes modificar el remote origin de todos los repositorios que ya habías clonado.
