



# **Configuración de entorno de Formación**

**Hecho por Álvaro Monforte Marín**

**1 de Diciembre de 2024**

# Copyright



Esta obra está licenciada bajo la Licencia Creative Commons Atribución-NoComercial-SinDerivadas 3.0 España. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-nd/3.0/es/> o envíe una carta a Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Repositorio de GitHub: <https://github.com/varo6678/formacion>

# Índice general

Copyright	I
Índice general	II
Lista de Figuras	III
Lista de Tablas	IV
1. Configuración en servidor remoto	1
1.1. Crear un usuario en Ubuntu y configurar acceso remoto por SSH	1
1.1.1. Crear un usuario y añadirlo al grupo sudo	1
1.1.2. Crear la carpeta .ssh para claves SSH	1
1.1.3. Configurar las claves autorizadas	2
1.1.4. Convertir un archivo .pem a una clave pública	2
1.1.5. Crear un archivo authorized_keys_pem	2
1.2. Configurar VSCode	3
1.3. Instalacion de miniconda	4

# Índice de figuras

1.1. Comandos para crear un usuario y asignar permisos de administrador (sudo).	1
1.2. Cambio a usuario.	1
1.3. Comandos para crear un usuario y asignar permisos de administrador (sudo).	1
1.4. Detalle de los permisos para las carpetas o ficheros alojados en nuestro path.	2
1.5. Permisos para lo alojado en la carpeta .ssh. Fuente: <a href="https://jonasbn.github.io">jonasbn.github.io</a>	2
1.6. Cambio de permisos para la carpeta .ssh a 700.	2
1.7. Se crea authorized_keys y se le asignan los permisos correspondientes para la correcta configuración.	2
1.8. Se crea authorized_keys y se le asignan los permisos correspondientes para la correcta configuración.	2
1.9. Se crea authorized_keys y se le asignan los permisos correspondientes para la correcta configuración.	3
1.10. Ventana de búsqueda del entorno de VSCode	3
1.11. Ventana de búsqueda del entorno de VSCode	3
1.12. Ventana de búsqueda del entorno de VSCode	3
1.13. Ventana de búsqueda del entorno de VSCode	4
1.14. Ventana de búsqueda del entorno de VSCode	4
1.15. Ventana de búsqueda del entorno de VSCode	4
1.16. Se crea authorized_keys y se le asignan los permisos correspondientes para la correcta configuración.	4
1.17. Se crea authorized_keys y se le asignan los permisos correspondientes para la correcta configuración.	4

# Índice de cuadros



# Configuración en servidor remoto

## 1.1 Crear un usuario en Ubuntu y configurar acceso remoto por SSH

Este es el proceso para crear un usuario, darle permisos de administrador y configurar el acceso remoto por SSH.

### 1.1.1 Crear un usuario y añadirlo al grupo sudo

```
sudo adduser usuario
sudo usermod -aG sudo usuario
```

Figura 1.1: Comandos para crear un usuario y asignar permisos de administrador (sudo).

- -a: Añade al usuario a un grupo existente sin sobrescribir otros grupos a los que ya pertenece.
- -G: Especifica los grupos secundarios a los que el usuario será añadido.

### 1.1.2 Crear la carpeta .ssh para claves SSH

Ya una vez tenemos los permisos creados para usuario. Simplemente nos cambiamos a su perfil haciendo:

```
sudo su usuario
```

Figura 1.2: Cambio a usuario.

Ahora gracias a lo ya ejecutado en la figura 1.3. Como tenemos permisos de administrador, podemos obviar el comando sudo (en caso de que este fuera necesario).

```
mkdir /home/usuario/.ssh
```

Figura 1.3: Comandos para crear un usuario y asignar permisos de administrador (sudo).

Ahora nos fijamos en los permisos que tiene la carpeta. Son muchos. Demasiados para este protocolo que requiere algo mas de seguridad.

Aquí tenemos una ligera idea de los permisos requeridos. Luego efectuamos el cambio, ver figura 1.6.

- La carpeta .ssh es donde se almacenan las claves SSH.
- El permiso 700 asegura que solo el propietario pueda leer, escribir y ejecutar.

```
> ll
total 24
drwxr-x--- 3 prueba prueba 4096 Dec  1 18:59 ./
drwxr-xr-x 6 root root 4096 Dec  1 18:58 ../
-rw-r--r-- 1 prueba prueba 220 Dec  1 18:58 .bash_logout
-rw-r--r-- 1 prueba prueba 3771 Dec  1 18:58 .bashrc
-rw-r--r-- 1 prueba prueba 807 Dec  1 18:58 .profile
drwxrwxr-x 2 prueba prueba 4096 Dec  1 18:59 .ssh/
```

Figura 1.4: Detalle de los permisos para las carpetas o ficheros alojados en nuestro path.

```
Permissions on folder and files should read:
• .ssh/ directory: 700 ( drwx----- )
• public key ( .pub file): 644 ( -rw-r--r-- )
• private key ( id_rsa ): 600 ( -rw----- )
• Your home directory should not be writeable by the group or others: 755 ( drwxr-xr-x ).
• Configuration file: .ssh/config: 600 ( -rw----- )
```

Figura 1.5: Permisos para lo alojado en la carpeta .ssh. Fuente: [jonasbn.github.io](https://jonasbn.github.io)

```
sudo chmod 700 /home/usuario/.ssh
```

Figura 1.6: Cambio de permisos para la carpeta .ssh a 700.

### 1.1.3 Configurar las claves autorizadas

A priori tenemos un fichero con extension .pem. Este al ser la clave privada tendra un permiso 600. Que debemos asegurarnos de que sea asi. Ademas debera pertenecer al propio usuario. Lo unico que este debe estar alojado en nuestro ordenador personal (en local). En el servidor remoto tiene que estar con la denominacion authorized\_keys la clave publica. Pero podemos dejar a modo de placeholder hasta ponerla su fichero de destino vacio. Ademas con sus permisos correctos

```
touch /home/usuario/.ssh/authorized_keys
sudo chmod 600 /home/usuario/.ssh/authorized_keys
sudo chown -R usuario:usuario /home/usuario/.ssh
```

Figura 1.7: Se crea authorized\_keys y se le asignan los permisos correspondientes para la correcta configuración.

- Con el comando touch creamos el nuevo fichero authorized\_keys.
- El archivo authorized\_keys contiene las claves públicas permitidas para autenticarse.
- chmod 600: El archivo debe ser accesible solo para el propietario.
- chown: Cambia la propiedad de la carpeta .ssh y su contenido al usuario correspondiente.

### 1.1.4 Convertir un archivo .pem a una clave pública

Si tienes una clave privada en formato .pem, puedes generar la clave pública de esta manera:

```
ssh-keygen -y -f authorized_keys_pem > authorized_keys
```

Figura 1.8: Se crea authorized\_keys y se le asignan los permisos correspondientes para la correcta configuración.

### 1.1.5 Crear un archivo authorized\_keys\_pem

Le ponemos una denominacion diferente para no confundirla con la publica de destino.



```
touch authorized_keys_pem
```

Figura 1.9: Se crea `authorized_keys` y se le asignan los permisos correspondientes para la correcta configuración.

Esto crea un archivo vacío llamado `authorized_keys_pem`, necesario para almacenar la clave privada. Finalmente quedara copiar el contenido de esta dentro del `authorized_keys` publico que habiamos hecho anteriormente en la figura 1.17.

Luego abrimos el editor de texto nano (ya preinstalado) copiamos y pegamos el contenido de `authorized_keys_pem` en `authorized_keys`. Tambien podriamos simplemente haber copiado `authorized_keys_pem` dentro de la misma carpeta con el nombre `authorized_keys` y despues asignarle los permisos correspondientes como hicimos en la figura 1.17.

La configuracion ya quedaria lista para sentencia.

## 1.2 Configurar VSCode

Aqui tenemos que acceder a la siguiente ventana, apretando `ctrl+shift+p`. En caso de que no aparezca, poner en la ventana `ssh` y buscar dicho resultado.

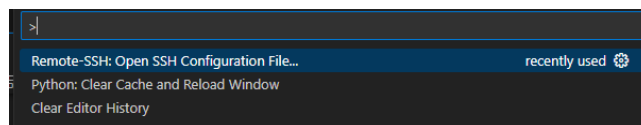


Figura 1.10: Ventana de busqueda del entorno de VSCode

Ahora hacemos *click* y vamos a la carpeta alojada en nuestro usuario del S0 y despues incorporamos las variables que necesita VSCode para poder conectarse via OpenSSH.

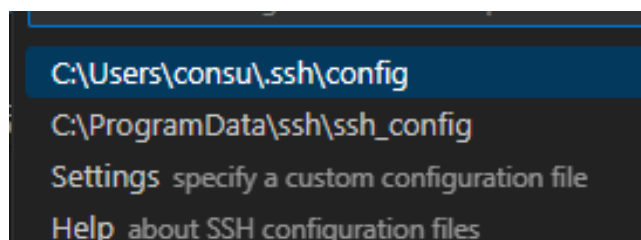


Figura 1.11: Ventana de busqueda del entorno de VSCode

Esta seria la configuracion:

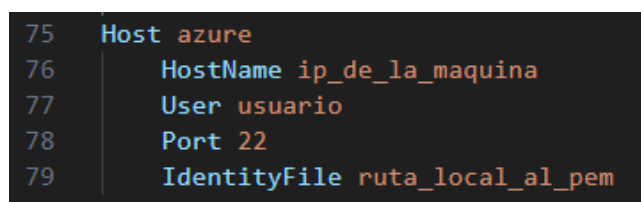


Figura 1.12: Ventana de busqueda del entorno de VSCode

Saldra una ventana en la que tenemos que poner el SO de destino, en nuestro caso a priori deberia ser Linux (independientemente de la distribucion).

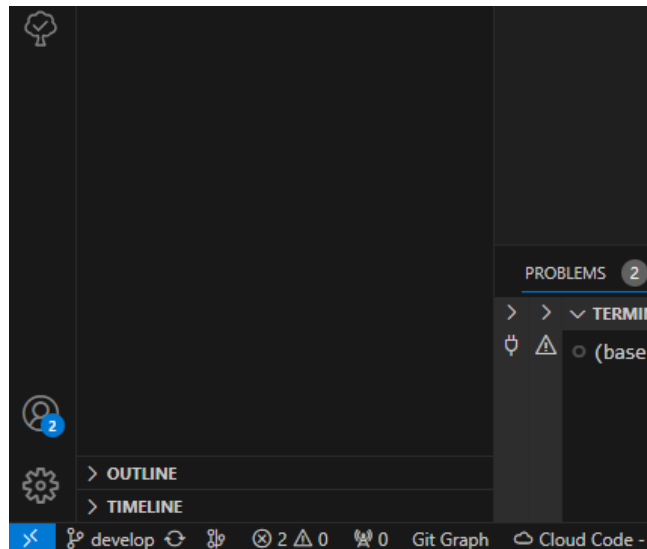


Figura 1.13: Ventana de búsqueda del entorno de VSCode

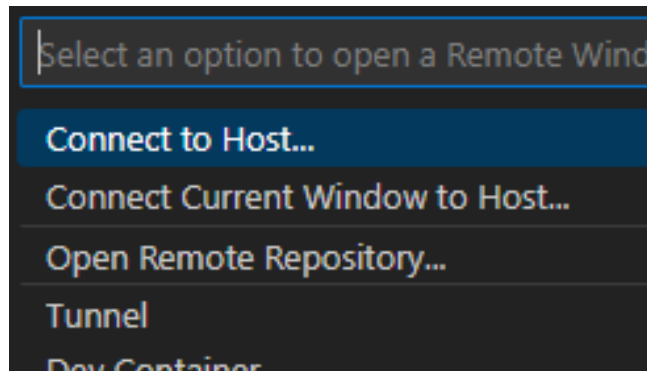


Figura 1.14: Ventana de búsqueda del entorno de VSCode



Figura 1.15: Ventana de búsqueda del entorno de VSCode

### 1.3 Instalacion de miniconda

```
mkdir $HOME
cd $HOME
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O miniconda.sh
chmod +x miniconda.sh
./miniconda.sh
export PATH=~/miniconda3/bin:$PATH
source ~/.bashrc
```

Figura 1.16: Se crea authorized\_keys y se le asignan los permisos correspondientes para la correcta configuración.

```
conda --version
conda create -n nombre_del_entorno python=3.12
```

Figura 1.17: Se crea authorized\_keys y se le asignan los permisos correspondientes para la correcta configuración.