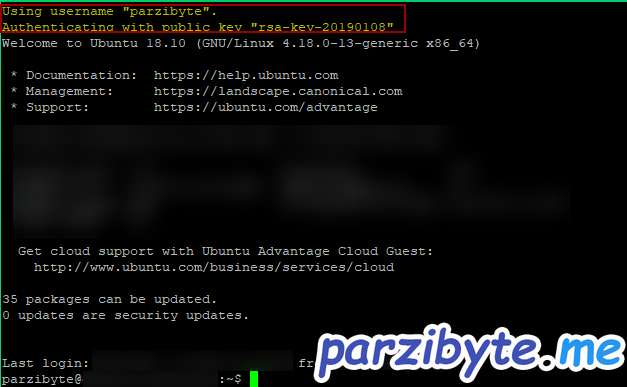
* Autenticación desde Windows a Ubuntu Server con PuTTY y claves SSH



**Autenticación desde Windows a Ubuntu Server con PuTTY y claves SSH**

[Dejar un comentario](https://parzibyte.me/blog/2019/01/11/autenticacion-windows-ubuntu-server-putty-claves-ssh/#respond) / [linux](https://parzibyte.me/blog/category/linux/), [Seguridad](https://parzibyte.me/blog/category/seguridad/), [Servidores](https://parzibyte.me/blog/category/servidores/), [windows](https://parzibyte.me/blog/category/windows/) / Por [parzibyte](https://parzibyte.me/blog/author/parzibyte/" \o "Ver todas las entradas de parzibyte)

Para loguearnos en un [servidor](https://parzibyte.me/blog/2019/01/11/autenticacion-windows-ubuntu-server-putty-claves-ssh/) con [Ubuntu Server](https://parzibyte.me/blog/2019/01/11/autenticacion-windows-ubuntu-server-putty-claves-ssh/) no se recomienda usar el acceso por usuario y contraseña; sino por **claves públicas y privadas**. De esta manera, no se necesita poner la contraseña, sino simplemente proporcionar un archivo con la clave privada.

En este tutorial **vamos a ver cómo crear un par de claves con PuTTYgen, agregarlas a nuestro servidor en el archivo authorized\_keys y finalmente loguearnos a través de PuTTY**.

Todo esto es para que al final podamos usar SSH para conectarnos a un servidor de Linux desde Windows.

**Nota:** esto conecta desde Windows a Ubuntu. Si quieres, en cambio, conectarte desde Ubuntu a otra máquina con Ubuntu [aquí está el tutorial](https://parzibyte.me/blog/2018/08/20/autenticacion-ssh-clave-publica-privada/).

**Requisitos**

Debes saber la IP o host completo de tu servidor, así como los comandos básicos de Linux.

**Nota sobre seguridad**

Nunca compartas capturas de pantalla, imágenes o cosas que muestren tus claves. Aunque se supone que la pública es pública por algo, igualmente recomiendo no compartirla.

Recuerda guardar en un lugar seguro estas claves, pues si las pierdes no podrás autenticarte (tendrás que usar usuario y contraseña) y si un atacante te la roba podrá acceder fácilmente.

En este tutorial muestro claves privadas y públicas, pero no las estoy usando en ningún lado más que para pruebas locales.

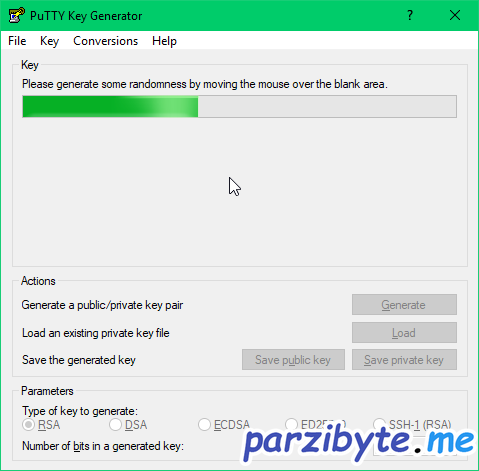
**Obtener PuTTY y PuTTYgen**

Descargamos el instalador o las herramientas por separado desde [**esta página oficial**](https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html). Debemos tener a *PuTTY* y a *PuTTYgen* como ejecutables, es decir, como un *.exe*.

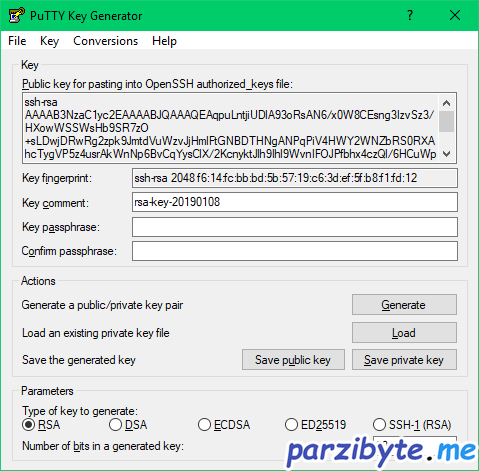
Para ello puedes buscar en Windows (en la barra de búsqueda) y deben salir los resultados, o si descargaste los archivos por separado entonces ejecuta cada exe según sea requerido.

**Paso 1: generar par de claves**

Abrimos **PuTTYgen** (dejamos las opciones intactas, como se ven en la captura) y hacemos click en **Generate**. Nos pedirá mover el puntero sobre el rectángulo, todo esto para generar algo aleatorio.

[](https://parzibyte.me/blog/wp-content/uploads/2019/01/Generando-clave-con-PuTTYgen.png)Generando clave con PuTTYgen

Cuando termine, mostrará el par de claves generadas en una ventana como esta:

[](https://parzibyte.me/blog/wp-content/uploads/2019/01/Par-de-claves-generadas-con-PuTTYgen.png)Par de claves generadas con PuTTYgen

Eso generó dos claves, una pública y una privada. Hacemos click en **Save public key** para guardar la pública, y en **Save private key** para guardar la privada.

Si pregunta si deseamos guardarla sin una **passphrase**, indicamos que sí. O igualmente puedes poner a una de las mismas para agregar una protección extra. En mi caso no lo haré.

La extensión de la clave pública no debería importar, y la de la privada es normalmente una con extensión *.ppk*

Arriba está también una clave **que comienza con ssh-rsa**, haz click derecho sobre ella para **Seleccionar todo el texto**, luego, cópialo y/o **guárdalo en un lugar seguro**.

En mi caso he guardado la clave privada y pública en mi disco, y el texto de hasta arriba que debemos copiar es el siguiente:

ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEAqpuLntjiUDlA93oRsAN6/x0W8CEsng3IzvSz3/HXowWSSWsHb9SR7zO+sLDwjDRwRg2zpk9JmtdVuWzvJjHmlFtGNBDTHNgANPqPiV4HWY2WNZbRS0RXAhcTygVP5z4usrAkWnNp6BvCqYysClX/2KcnyktJlh9Ihl9WvnIFOJPfbhx4czQl/6HCuWpwGhZRqKBojH/hbwfpu7UapmIU2hl0q2nAvCHTamMmDFdayYE8NwTPyO8gBLNKPTi6slKyCsxirHy6Izdd0geo6G8XWo0S/y0zAxqlE98NpUkIRO6xiHxE14J4W+assYL9o7UF/jhGuRKGVlQdsSE5YB4bJQ== rsa-key-20190108

Copy

Recuerda que en tu caso cambiará.

**Paso 2: agregar clave al servidor de Linux**

Ya tenemos nuestras claves, ahora le diremos a Linux que acepte conexiones con nuestra clave pública. Para ello, **debemos añadir el texto que copiamos** a un archivo llamado *authorized\_keys* que reside en *~/.*[*ssh*](https://parzibyte.me/blog/2019/01/11/autenticacion-windows-ubuntu-server-putty-claves-ssh/) dependiendo de nuestro usuario.

Por ejemplo, si nuestro usuario es *maria\_jose*, nuestro directorio de trabajo es */home/maria\_jose*, y por lo tanto, *~* apunta a ese directorio.

En caso de ser root, nuestro directorio *~* es */root*. El punto es que debemos crear la carpeta *.ssh* en nuestro directorio de trabajo, y dentro de esa carpeta, el archivo *authorized\_keys*.

Para crearlo, hacemos esto:

mkdir ~/.ssh

chmod 0700 ~/.ssh

touch ~/.ssh/authorized\_keys

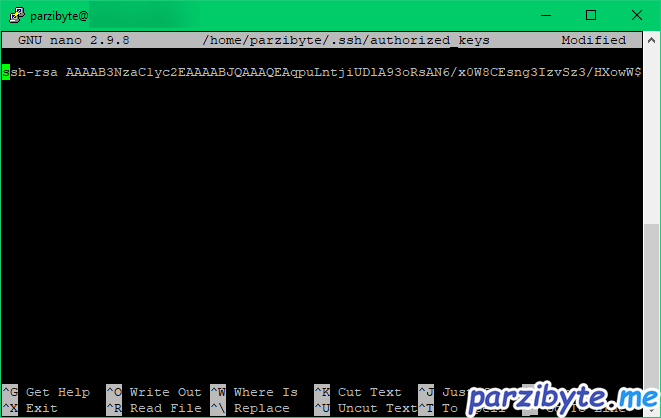
chmod 0644 ~/.ssh/authorized\_keys

Copy

Eso se encargará de darle [**permisos a los archivos**](https://parzibyte.me/blog/2018/08/22/ejercicios-chown-chgrp-chmod-en-linux/), así como de crear a los mismos si no existen. Una vez que ya esté creado, lo vamos a editar con nano así:

nano ~/.ssh/authorized\_keys

Dentro de él vamos a **pegar la clave** que copiamos anteriormente, **no las que guardamos como archivos**, sino la clave que copiamos. El archivo debe verse así:

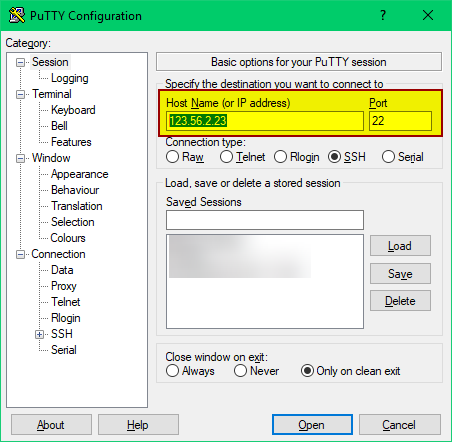
[](https://parzibyte.me/blog/wp-content/uploads/2019/01/Modificar-archivo-authorized_keys.png)Modificar archivo authorized\_keys

No se ve toda la clave, pero no importa mientras esté pegada completamente. Después guardamos cambios con CTRL + O, presionamos Enter y luego CTRL + X para salir.

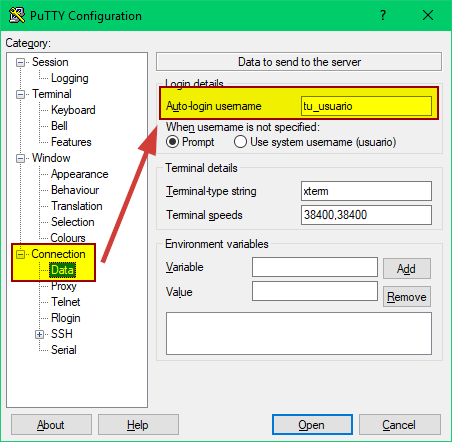
Con eso hemos terminado de [configurar](https://parzibyte.me/blog/2019/01/11/autenticacion-windows-ubuntu-server-putty-claves-ssh/) el servidor. Ahora vamos a autenticarnos sin contraseña.

**Paso 3: configurar PuTTY con host, usuario y claves**

Es momento de abrir PuTTY y configurar algunas cosas. Al abrirlo, ponemos en la siguiente ventana el host y puerto, que normalmente es el 22:

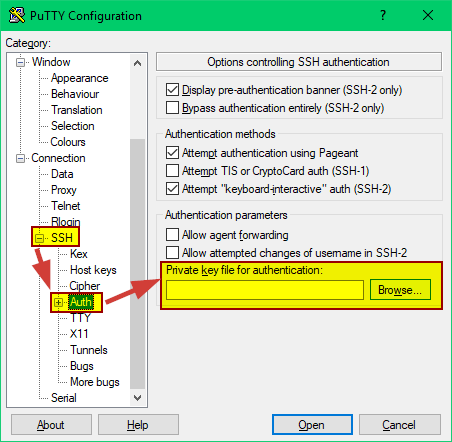
[](https://parzibyte.me/blog/wp-content/uploads/2019/01/PuTTY-paso-1.png)PuTTY paso 1

Recuerda que el host es el host de tu servidor, o la IP. Ahora vamos a configurar el usuario con el que nos vamos a conectar por defecto, así no tenemos que escribirlo cada vez. Para ello, navegamos a **Connection** > **Data** y en **Auto-login username** escribimos el **nombre de usuario** de Linux.

[](https://parzibyte.me/blog/wp-content/uploads/2019/01/PuTTY-paso-2-proporcionar-usuario-que-se-loguea.png)PuTTY paso 2, proporcionar usuario que se loguea

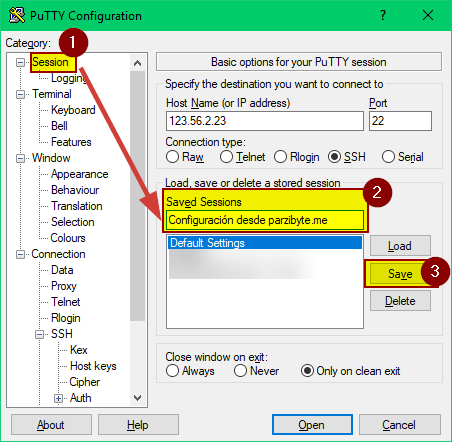
**Nota:** por ejemplo, si pegaste el contenido de la clave pública en el directorio de trabajo de *maria\_jose* (y entonces estabas logueado como *maria\_jose*) entonces pon, de nuevo, *maria\_jose* para loguearte con ese usuario. **Si fuera otro usuario**, primero deberías añadirlo a su directorio de trabajo con el proceso de arriba.

Ahora viene el **paso más importante**, expandimos la sección de **SSH**, hacemos click en **Auth** y en donde dice que proporcionemos la clave privada, **seleccionamos el archivo de la clave privada** haciendo click en **Browse** (es la que guardamos anteriormente, no es la que copiamos, es la que tiene extensión **ppk**):

[](https://parzibyte.me/blog/wp-content/uploads/2019/01/PuTTY-paso-3-agregar-clave-privada.png)PuTTY paso 3, agregar clave privada

**Nota:** aunque en mi caso el espacio a la izquierda de donde dice **Browse** está vacío, en el tuyo debería tener la ruta de la clave.

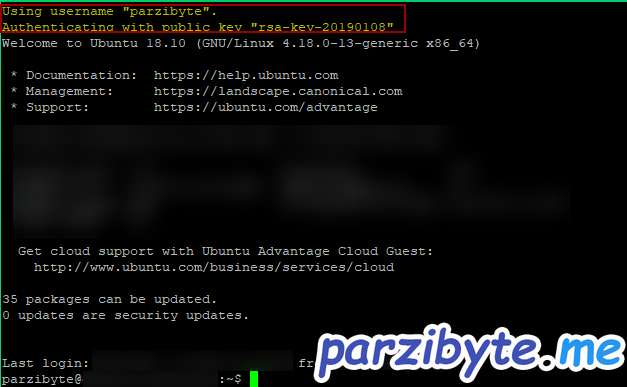
Ahora vamos a guardar todo lo que configuramos. Para ello, volvemos a la sección **Session** y le ponemos un nombre a la misma, luego hacemos click en **Save**:

[](https://parzibyte.me/blog/wp-content/uploads/2019/01/PuTTY-paso-4-guardar-sesi%C3%B3n-y-configuraciones.png)PuTTY paso 4, guardar sesión y configuraciones

De esta manera, la próxima vez que intentemos loguearnos bastarán un par de clicks.

**Paso 4: conectarnos con PuTTY desde Windows a Linux, con SSH**

Ahora sí bastará dar doble click en la sesión que guardamos anteriormente (también se puede seleccionar y hacer click en **Open**):

[](https://parzibyte.me/blog/wp-content/uploads/2019/01/Acceso-desde-SSH-a-servidor-Linux-con-clave-privada.png)Acceso desde SSH a servidor Linux con clave privada

Lo importante es que en la parte de arriba aparece que nos estamos [autenticando](https://parzibyte.me/blog/2019/01/11/autenticacion-windows-ubuntu-server-putty-claves-ssh/) con el usuario que configuramos en PuTTY, usando la clave privada elegida anteriormente.

De esta manera no pide ninguna contraseña.

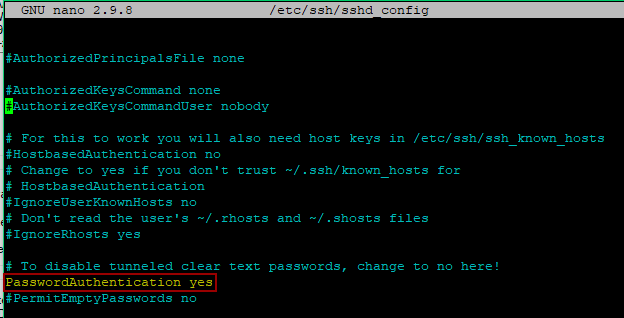
**Siguientes pasos**

Opcionalmente puedes desactivar el login por usuario y contraseña a través de SSH. Es decir, que ya nadie pueda entrar con usuario y contraseña, sino únicamente por claves privadas. La desventaja de esto es que si pierdes las claves, nunca podrás entrar.

En caso de que lo desees así, edita el archivo /etc/ssh/sshd\_config con:

sudo nano /etc/ssh/sshd\_config

En él, busca la línea que dice **PasswordAuthentication**:

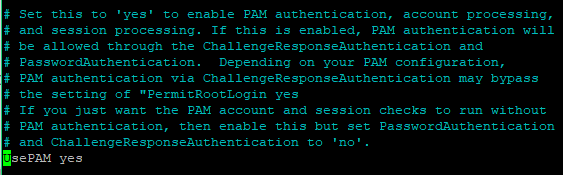
[](https://parzibyte.me/blog/wp-content/uploads/2019/01/Cambiar-password-authentication.png)Cambiar password authentication

Para que quede así:

PasswordAuthentication no

Igualmente con una que dice **usePAM**, debe quedar así:

usePAM no

[](https://parzibyte.me/blog/wp-content/uploads/2019/01/Usepam-ssh.png)Usepam ssh

Guarda cambios con CTRL + O, presiona Enter y luego CTRL + X.

Para aplicar los cambios, ejecuta finalmente:

sudo systemcl restart ssh

O en versiones anteriores:

sudo reload ssh

También funciona con:

sudo service ssh reload

Y con eso hemos terminado.