

Apuntes Segunda Evaluación - SEREI

Ismael Macareno Chouikh

2025-02-19

Índice

1. FTP	3
1.1. vsftpd	3
1.1.1. Instalación	3
1.1.2. Configuración	3
1.1.3. chroot	4
1.1.4. Grupos de usuario para ftp	4
1.1.5. Crear usuarios para ftp	5
1.1.6. Usuario anónimo	6
1.2. FTP sobre SSL	7
1.2.1. Integrar seguridad SSL a FTP	7
1.2.2. Generar un certificado SSL para FTP	7
1.3. <i>Filezilla Server</i>	10
1.3.1. Instalación de <i>Filezilla Server</i>	10
1.3.2. Configuración de <i>Filezilla Server</i>	11
1.3.3. Conexión desde un cliente	12
1.3.4. Configuración del servidor	13
1.3.5. Conexión segura por SSL	15
2. SSH	17
2.1. Telnet	17
2.2. Ficheros relevantes	17
2.3. Instalación del servidor openssh	17
2.4. Ver la dirección en known_hosts en vez de el hash	18
2.5. Eliminar error intento ataque Ddos	18
2.6. Conexión por clave pública	18
2.6.1. Configuración del servidor	20
2.7. Directorio \$HOME/.ssh/	20
2.8. Algoritmo de cifrado	21
2.9. SSH-Agent	22
2.10. Windows: ssh , claves y ssh-agent	23
2.10.1. Putty	24
2.10.2. MobaXterm	28
2.11. Cluster ssh	30
2.11.1. TODO Clusters	31
2.11.2. TODO <i>Tags</i>	31
2.12. Entorno gráfico	31
2.12.1. GNU/Linux	31

2.12.2. <i>Microsoft Windows</i>	32
2.13. Tunelización	32
2.13.1. Primer ejemplo	32
2.13.2. Túnel remoto	33
2.14. Escritorio remoto	34
2.14.1. RDP	34
2.14.2. VNC	35
3. Servicio web, apache2	35
3.1. Instalación	35
3.2. Apuntes teóricos	35
3.2.1. Comandos útiles	36
3.3. Práctica	36
3.3.1. Cambiar página por defecto	36
3.3.2. <i>option indexes</i>	37
3.3.3. Crear un sitio personalizado	38
3.3.4. Contraseñas y restricción de acceso	40
3.3.5. Páginas por puertos e IPs	41
3.3.6. Sitio virtual, directorios <i>userdir</i>	44
3.3.7. Securización	45

1. FTP

1.1. vsftpd

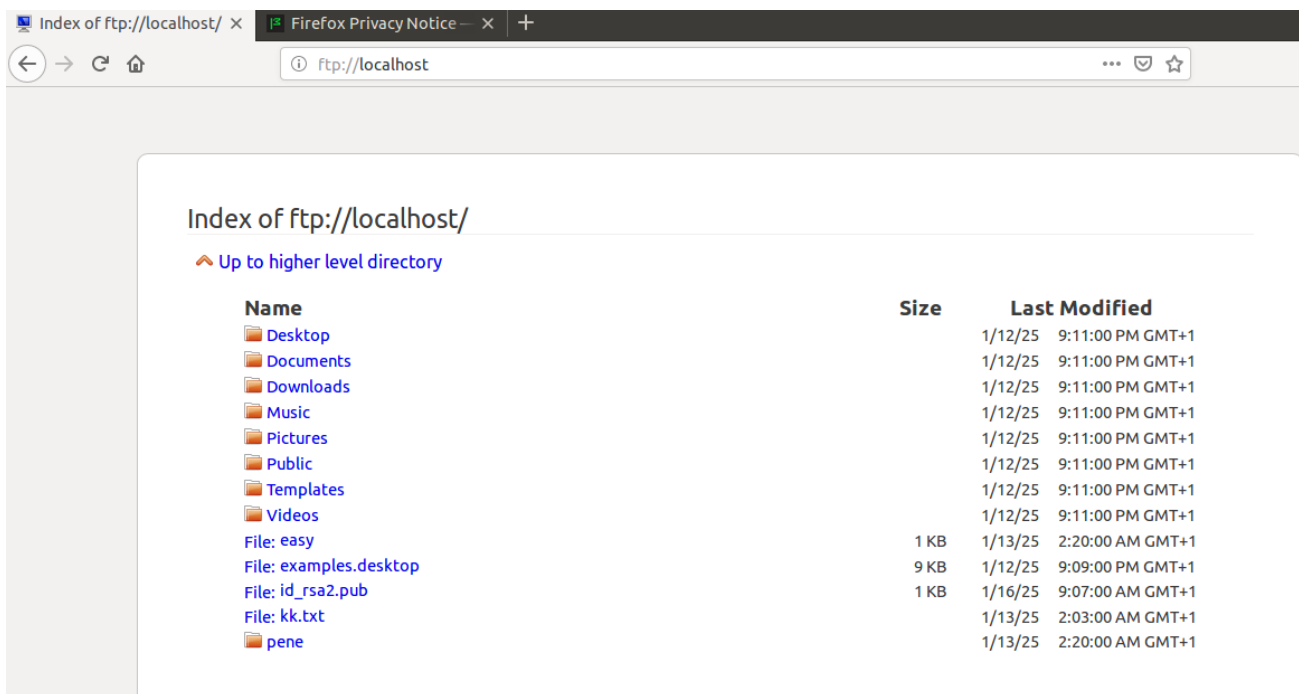
1.1.1. Instalación

Para instalar cualquier tipo de paquete en una distribución Ubuntu tendremos que usar las siguientes instrucciones:

- `sudo apt-get update`
- `sudo apt-get install vsftpd`

Podremos comprobar que el paquete está instalado mediante el uso de la instrucción `netstat -putona | grep -i vsftpd`

También se podrá comprobar accediendo desde nuestro navegador web a `ftp://localhost`



1.1.2. Configuración

Para configurar aspectos de `vsftpd` lo que tendremos que hacer será modificar el fichero `/etc/vsftpd.conf`

Como primera configuración lo que haremos será descomentar la línea 14 del fichero la cuál contiene `listen=YES` para que así cuando se inicie el sistema ftp se inicie con él.

```
# daemon started from an initscript.
listen=YES
```

Luego también descomentaremos las líneas 26, 29 y 33

```
# Uncomment this to allow local users to log in.
local_enable=YES # Conexión con los usuarios locales del servidor
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES # Los usuarios podrán escribir y descargar cosas
```

```
#  
# Default umask for local users is 077. You may wish to change this to 022,  
# if your users expect that (022 is used by most other ftpd's)  
local_umask=022 # Los permisos de los ficheros subidos serán 755
```

1.1.3. chroot

Para habilitar el **chroot** lo que tendremos que hacer será descomentar la línea 120 del fichero `/etc/vsftpd.conf`

```
chroot_local_user=YES  
chroot_list_enable=YES  
# (default follows)  
chroot_list_file=/etc/vsftpd.chroot_list
```

De esta manera conseguiremos enjaular a los usuarios, es decir, impedir el acceso a otras carpetas fuera del directorio de inicio de cada usuario.

- **chroot_list_enable=YES**: Permitir solo a ciertos usuario el poder navegar por todo el árbol de directorios del servidor
- **chroot_list_file**: Estarán listados los usuarios que pueden navegar hacia arriba por los directorios del servidor.

A los usuarios que no queramos enjaular, los tendremos que meter en el fichero `/etc/vsftpd.chroot_list`

```
root@ubuntu2:~# cat /etc/vsftpd.chroot_list  
maka
```

Para permitir la escritura tendremos que añadir la línea **allow_writeable_chroot=YES**

Para activar el modo pasivo para un máximo de 100 conexiones tendremos que añadir lo siguiente:

- **pasv_enable=YES**
- **pasv_min_port=40000**
- **pasv_max_port=40100**

Luego de haber añadido esas tres líneas al fichero `/etc/vsftpd.chroot_list` podremos reiniciar el servicio mediante la instrucción **service vsftpd restart** y seguidamente **ftp** escuchará por el puerto 21.

1.1.4. Grupos de usuario para ftp

Los usuarios que se conectarán no tendrán acceso al servidor vía **SSH** o *shell* local por lo tanto debemos darles permisos especiales.

Cuando se instala **vsftpd** se crea un grupo y usuario llamados **ftp** por defecto. Lo podremos verificar mediante la instrucción **egrep -i ftp /etc/group**

Ahora lo que haremos será habilitar una *shell* fantasma para que los usuarios **ftp** no puedan entrar en la consola del servidor.

Para permitir el acceso del usuario **nologin** a una *shell* fantasma lo que tendremos que hacer será modificar el fichero `/etc/shells` de tal manera que nos quede así:

```
# /etc/shells: valid login shells  
/bin/sh
```

```
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/tmux
/usr/sbin/nologin
```

1.1.5. Crear usuarios para ftp

Para los usuarios que pertenecerán al grupo ftp, crearemos la carpeta de los usuarios ftp en el servidor, será donde los usuarios ftp tendrán sus carpetas personales, todo en un directorio raíz para facilitar la administración.

```
root@ubuntu2:~# mkdir /home/usuariosftp
root@ubuntu2:~# chmod -R 777 /home/usuariosftp/
```

Crear usuarios con el siguiente comando. Usa *shell /usr/sbin/nologin* para evitar el acceso a la *shell* para los usuarios ftp.

- `useradd -g ftp -d /home/usuariosftp/nombre -m -s /usr/sbin/nologin -c «Nombre completo» nombre`

Los parámetros que usamos en la línea anterior son:

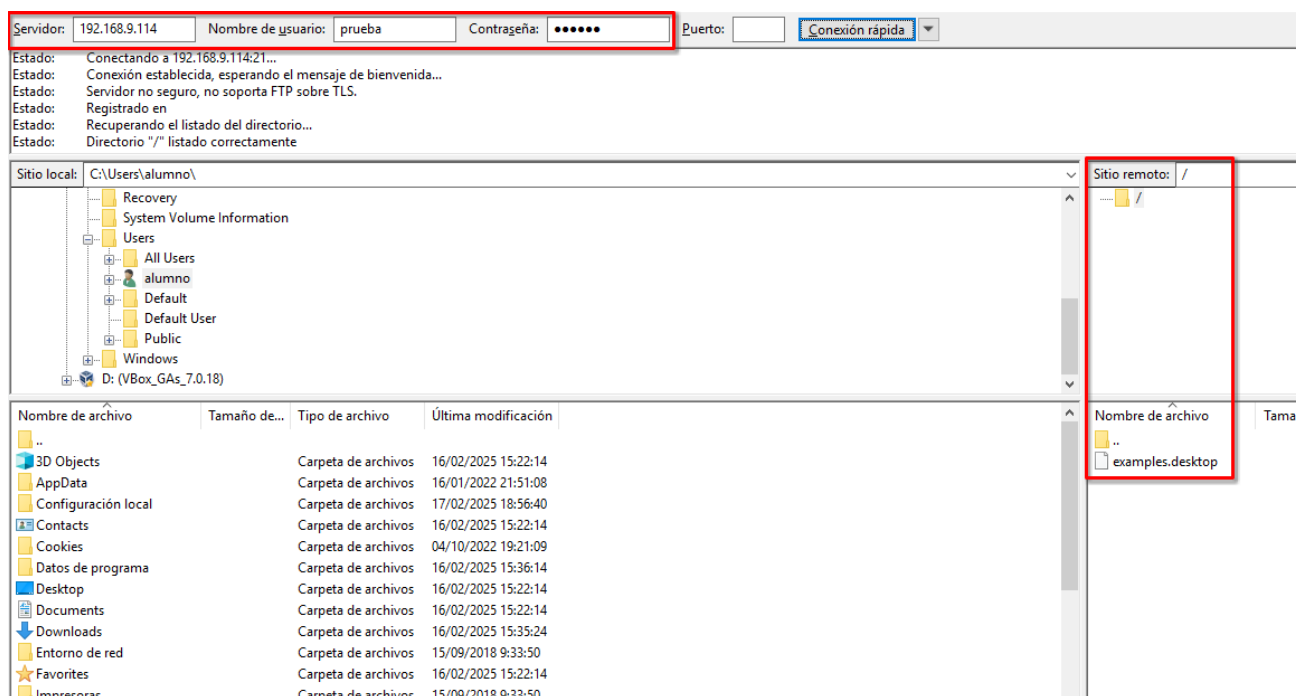
- `-g ftp`: el usuario pertenece al grupo ftp
- `-d /home/usuariosftp`: el directorio principal del usuario será `/home/usuariosftp/prueba`
- `-m` para que se cree automáticamente el usuario prueba en el directorio dentro de la raíz `/home/usuariosftp/`
- `-s /usr/sbin/nologin`: el usuario no tendrá acceso a la *shell* del sistema, así no podrá iniciar sesión en el servidor (solo acceso ftp)
- `-c "Prueba usuarios"`: el nombre completo del usuario
- `prueba`: la última palabra será el nombre del usuario

Ahora lo que tendremos que hacer será crear la contraseña para el usuario **prueba** mediante el uso de la instrucción `passwd`

```
root@ubuntu2:~# passwd prueba
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Luego de todo el procedimiento de haber creado los usuario lo que podremos hacer será comprobar que de verdad todo está bien configurado tal y como deseamos.

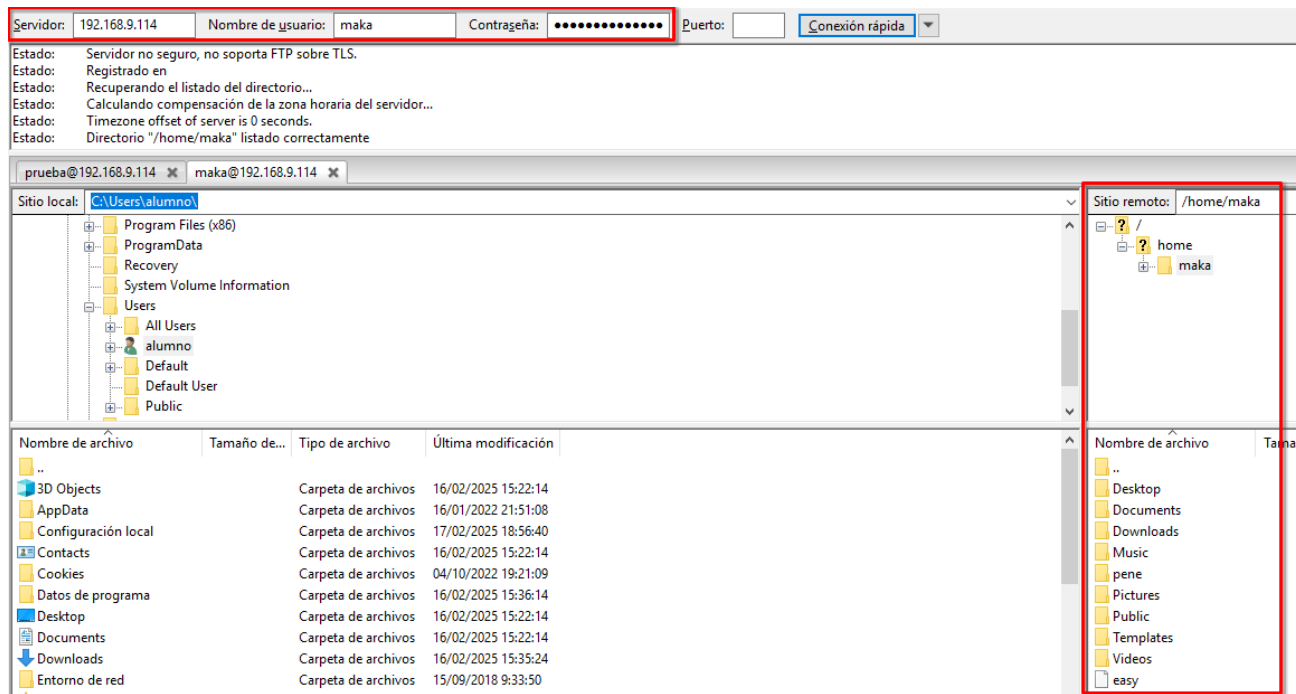
La primera comprobación la haré desde un cliente *Windows 10 LTSC* usando el programa Filezilla cliente.



Este usuario no puede iniciar sesión a la consola como dicen las directivas `nologin`

```
root@ubuntu2:~# su - prueba
This account is currently not available.
```

Ahora probaré a conectarme mediante *Filezilla client* pero a otro usuario el cuál no está enjaulado



1.1.6. Usuario anónimo

Para habilitar el usuario `anonymous` lo que tendremos que hacer será modificar el fichero `/etc/vsftpd.conf` descomentando la línea que contenga `anonymous_enable=YES` de la siguiente manera:

```
# Allow anonymous FTP? (Disabled by default)
anonymous_enable=YES
```

después de haber descomentado esa línea lo que tendremos que hacer será reiniciar el servicio mediante la instrucción `sudo service vsftpd restart` y ya podremos conectarnos usando el usuario `anonymous`

```
C:\Users\alumno>ftp 192.168.9.114
Conectado a 192.168.9.114.
220 (vsFTPd 3.0.2)
200 Always in UTF8 mode.
Usuario (192.168.9.114:(none)): anonymous
331 Please specify the password.
Contraseña:
230 Login successful.
ftp>
```

1.2. FTP sobre SSL

Si solemos usar el protocolo FTP y no queremos que nos saquen contraseñas o directamente los ficheros que no pasen sin encriptación entre un servidor y un cliente, para ello debemos habilitar la capa segura o SSL.

FTP en el puerto 21 es un gran riesgo para la seguridad porque con un analizador de paquetes TCP (Ej. *wireshark*), primero haré un esnifado para comprobar que el usuario y la contraseña viaja por la red en texto plano y los ficheros que enviemos también.

Aplique un filtro de visualización ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
7	0.003524	192.168.9.109	192.168.9.114	FTP	64	Request: AUTH TLS
8	0.003884	192.168.9.114	192.168.9.109	TCP	60	21 → 49703 [ACK] Seq=21 Ack=11 Win=29312 Len=0
9	0.004023	192.168.9.114	192.168.9.109	FTP	92	Response: 530 Please login with USER and PASS.
10	0.004195	192.168.9.109	192.168.9.114	FTP	64	Request: AUTH SSL
11	0.004496	192.168.9.114	192.168.9.109	FTP	92	Response: 530 Please login with USER and PASS.
12	0.045894	192.168.9.109	192.168.9.114	TCP	54	49703 → 21 [ACK] Seq=21 Ack=97 Win=262656 Len=0
13	0.760350	192.168.9.109	192.168.9.114	FTP	67	Request: USER prueba
14	0.760902	192.168.9.114	192.168.9.109	FTP	88	Response: 331 Please specify the password.
15	0.761264	192.168.9.109	192.168.9.114	FTP	67	Request: PASS prueba
16	0.782610	192.168.9.114	192.168.9.109	FTP	77	Response: 230 Login successful.
17	0.782803	192.168.9.109	192.168.9.114	FTP	60	Request: SYST
18	0.783136	192.168.9.114	192.168.9.109	FTP	73	Response: 215 UNIX Type: L8

1.2.1. Integrar seguridad SSL a FTP

Habrá que crear un directorio para almacenar los certificados `ssl`. Usaremos la instrucción `sudo mkdir -p /etc/ssl/private`

Luego tendremos que dar permisos a `root` para ese directorio mediante la instrucción `sudo chmod 700 /etc/ssl/private`

```
root@ubuntu2:~# mkdir -p /etc/ssl/private/
root@ubuntu2:~# chmod -R 700 /etc/ssl/private/
```

1.2.2. Generar un certificado SSL para FTP

Habrá que estar conectado con el usuario `root` ya que solo él tiene permisos a el directorio donde se almacena la clave `ssl`.

Luego habrá que ejecutar la siguiente instrucción:

- `openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem`

Y rellenar los campos de la siguiente manera:

```
root@ubuntu2:~# openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout /etc/ssl/private/vsftpd.pem
Generating a 4096 bit RSA private key
.....++
.....++
writing new private key to '/etc/ssl/private/vsftpd.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:MADRID
Locality Name (eg, city) []:ALCALA DE HENARES
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IES Alonso de Avellaneda
Organizational Unit Name (eg, section) []:ASIR
Common Name (e.g. server FQDN or YOUR name) []:asir.iesavellaneda.com
Email Address []:ismael.macareno@educa.madrid.org
```

Por último tendremos que modificar el fichero `/etc/vsftpd.conf` de tal manera que contenga las siguientes líneas:

- `ssl_enable=YES`: habilitar ssl
- `allow_anon_ssl=YES`: permite al usuario anónimo usar ssl
- `force_local_data_ssl=YES`: forzar a los usuarios ftp a usar ssl
- `force_local_login_ssl=YES`: forzar a los usuarios local a usar ssl
- `ssl_tlsv1=YES`: habilitar ssl v1
- `ssl_sslv2=NO`: no permitir ssl por seguridad
- `ssl_sslv3=NO`: no permitir ssl por seguridad
- `rsa_cert_file=/etc/ssl/private/vsftpd.pem`: definir la ruta donde esta el certificado

```
# This option specifies the location of the RSA certificate to use for SSL
# encrypted connections.
rsa_cert_file=/etc/ssl/private/vsftpd.pem
# This option specifies the location of the RSA key to use for SSL
# encrypted connections.
rsa_private_key_file=/etc/ssl/private/vsftpd.pem

# Habilitar escribir a los usuarios enjaulados
allow_writeable_chroot=YES

#Certificado SSL
```



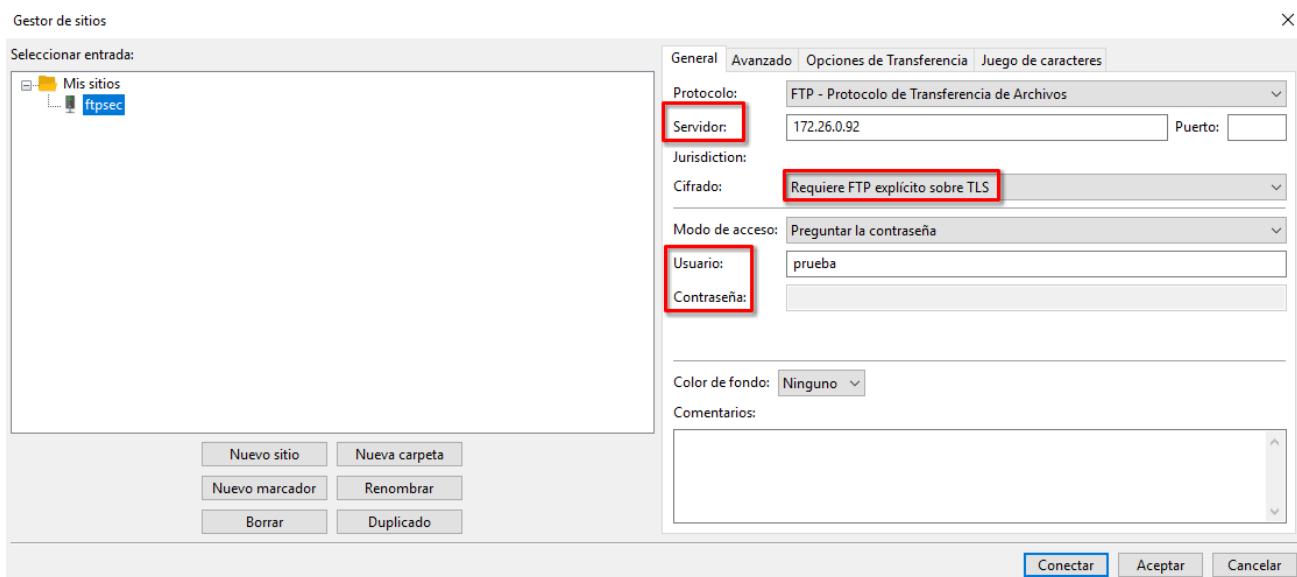
```
ssl_enable=YES
allow_anon_ssl=YES
force_local_data_ssl=YES
force_local_logins_ssl=YES
ssl_tlsv1=YES
ssl_sslv2=NO
## Compatibilidad
ssl_sslv3=NO
require_ssl_reuse=NO
ssl_ciphers=HIGH
```

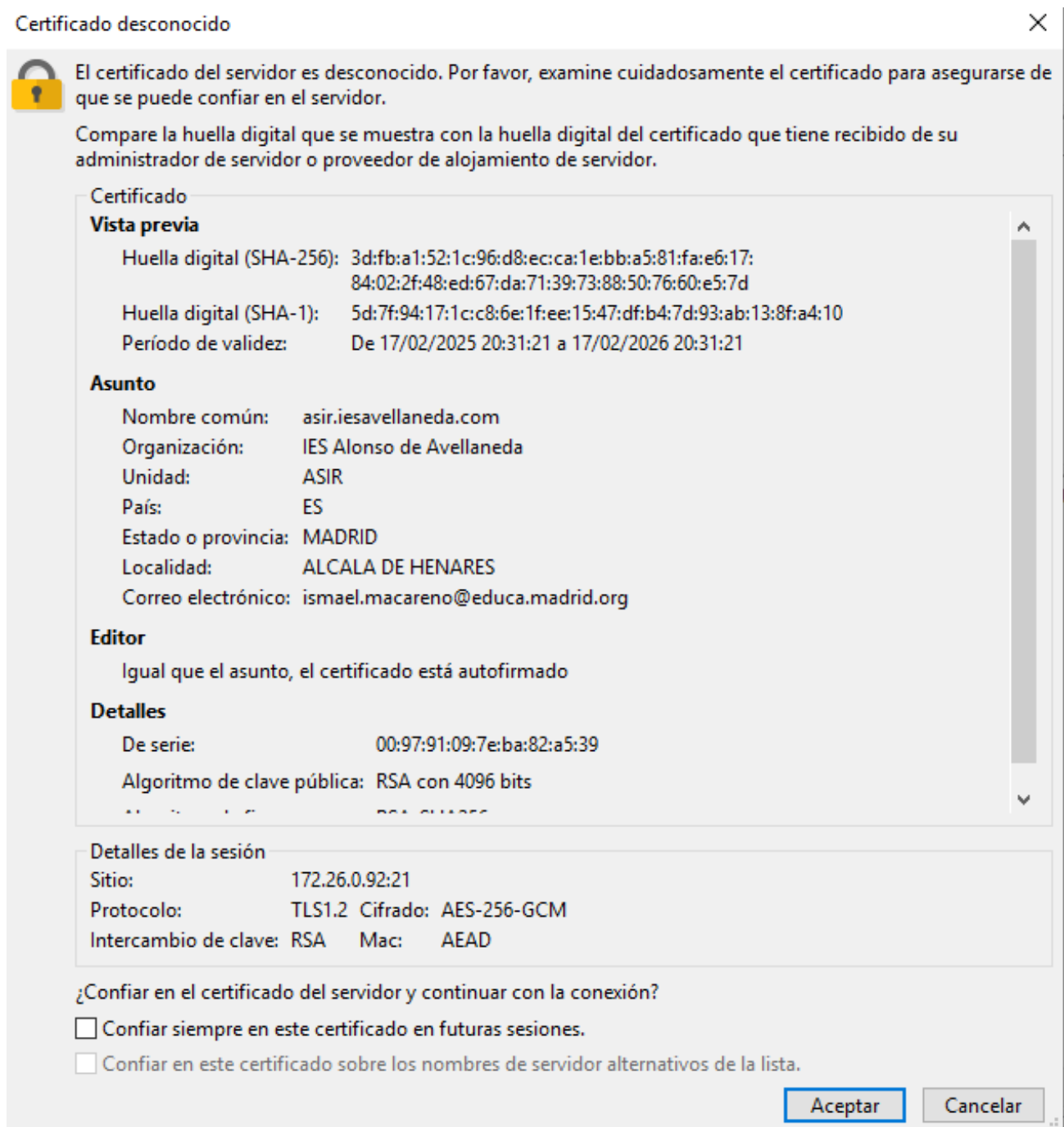
Guardamos el fichero y reiniciamos el servicio mediante la instrucción `sudo service vsftpd restart`

Ahora vamos con las comprobaciones, si intentamos conectarnos desde nuestro Filezilla cliente se rechaza la conexión por los usuarios `ftp` pero permite los anónimos.

Para conectarnos desde un usuario autenticado tendremos que habilitar **SSL** en el *software* de Filezilla cliente. Para ello en Filezilla iremos a:

1. Sitio nuevo
2. Habilitar TLS
3. Rellenaremos los datos para conectarnos





1.3. Filezilla Server

1.3.1. Instalación de Filezilla Server

Necesitaremos una máquina virtual *Windows Server 2016 Datacenter*.

Cuando tengamos la máquina virtual *Windows* lo que haremos será descargar un **.exe** de la versión *Filezilla server 0.9.51* y ejecutarlo para instalar.

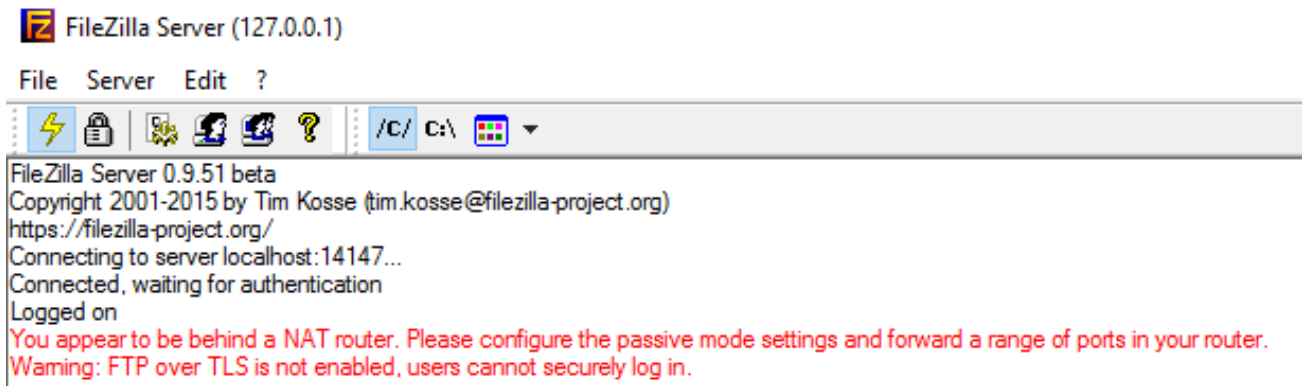
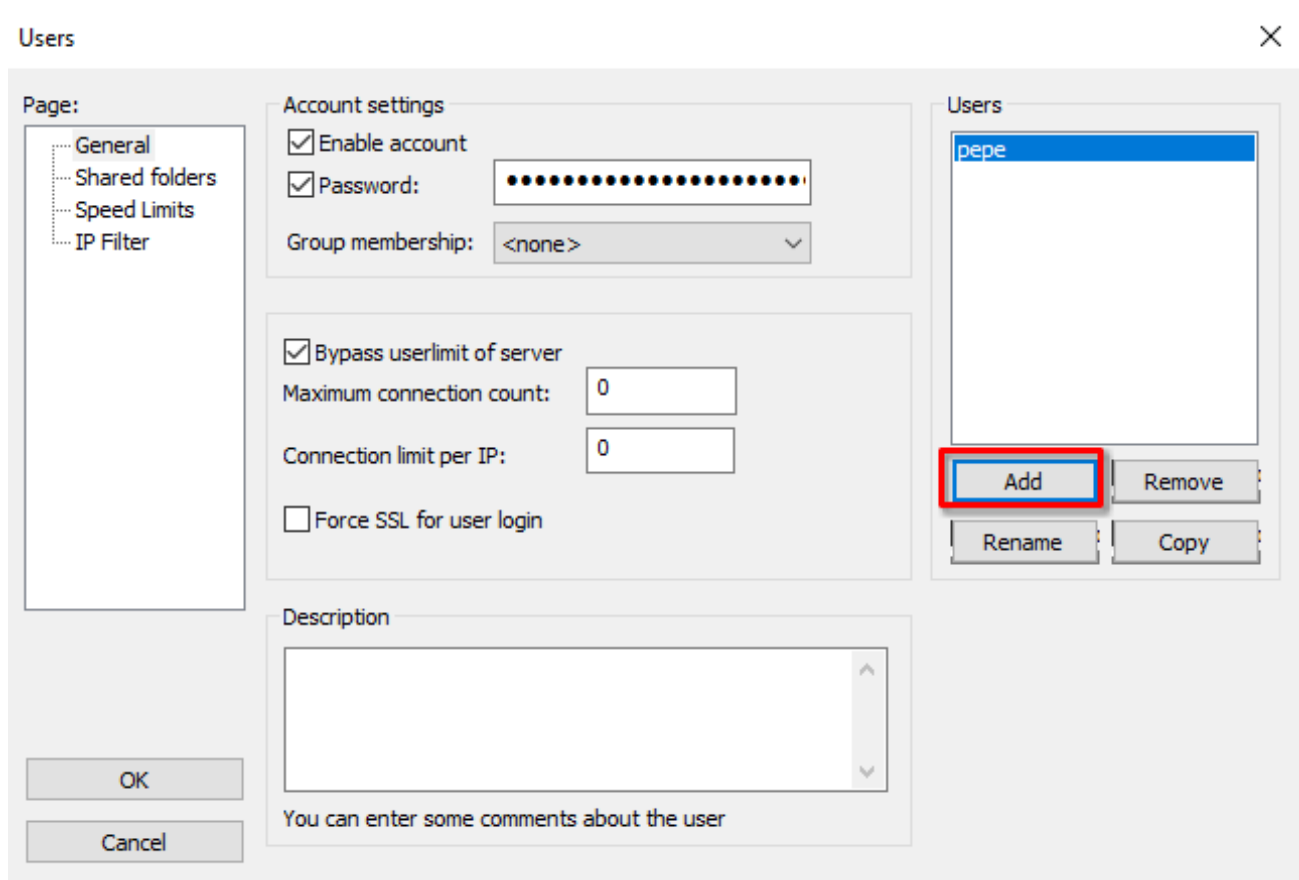


Figura 1: Macareno, Ismael (2025). Resultado instalación *Filezilla server* [PNG]. Propia

1.3.2. Configuración de *Filezilla Server*

Comenzaremos con la creación de un nuevo usuario. Para ello tendremos que presionar sobre el icono que muestra una persona.

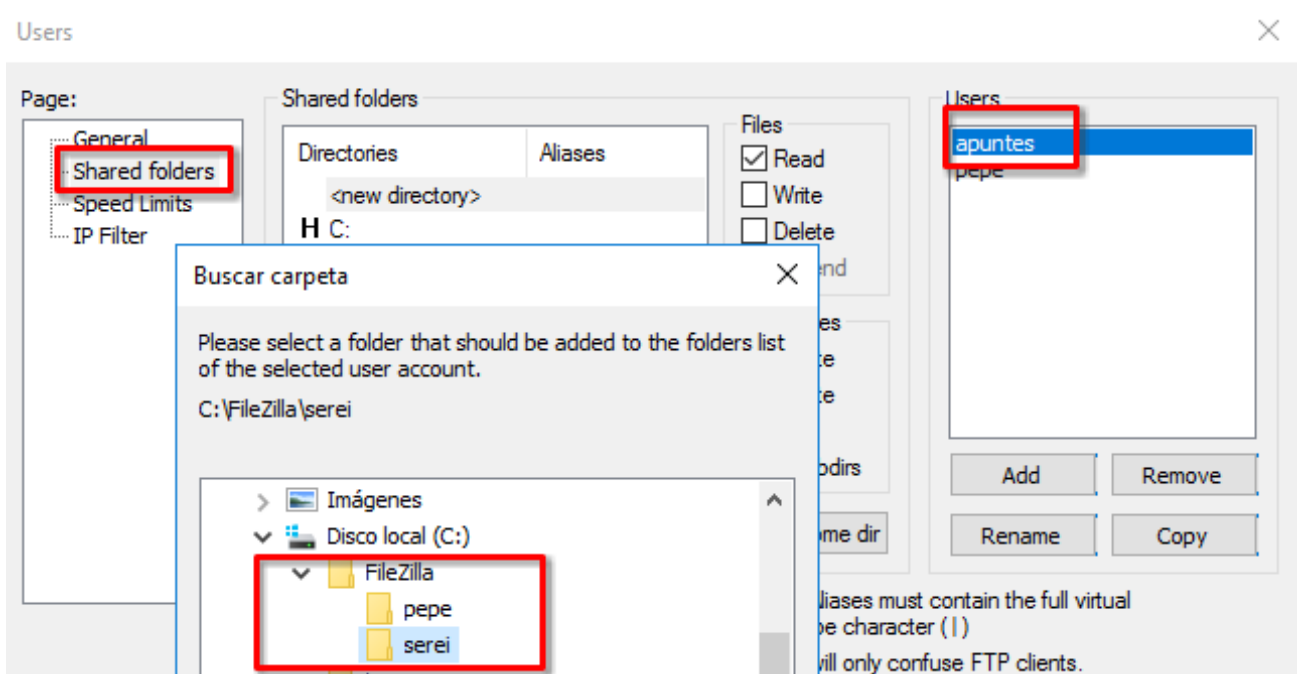
Cuando nos aparezca la ventana lo que tendremos que hacer será presionar sobre el botón que pone **Add**



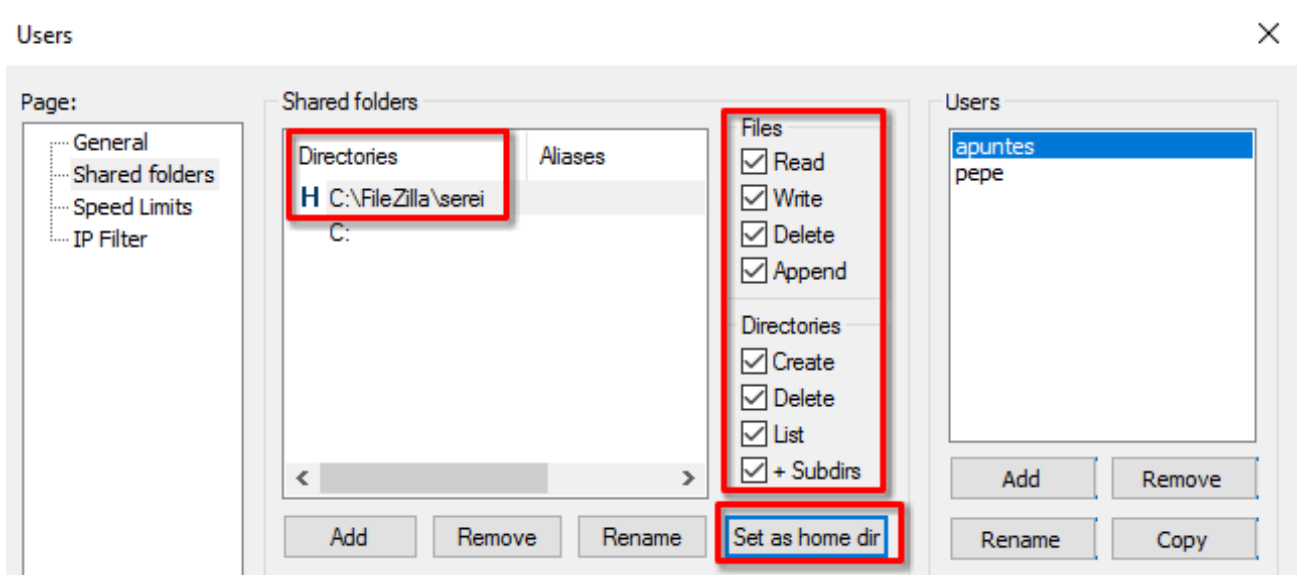
Al presionar sobre *add* lo que ocurrirá es que nos aparecerá una ventana solicitando el nombre del nuevo usuario que queremos crear.

Después en la página de *shared folders*, tendremos que presionar en *add* en nuevo para añadir una nueva carpeta. Lo que vamos a hacer es, en disco local (C:\) vamos a crear una nueva carpeta que se llame **Filezilla**, y dentro de

la misma, crear una carpeta **serei** para nuestro nuevo usuario.



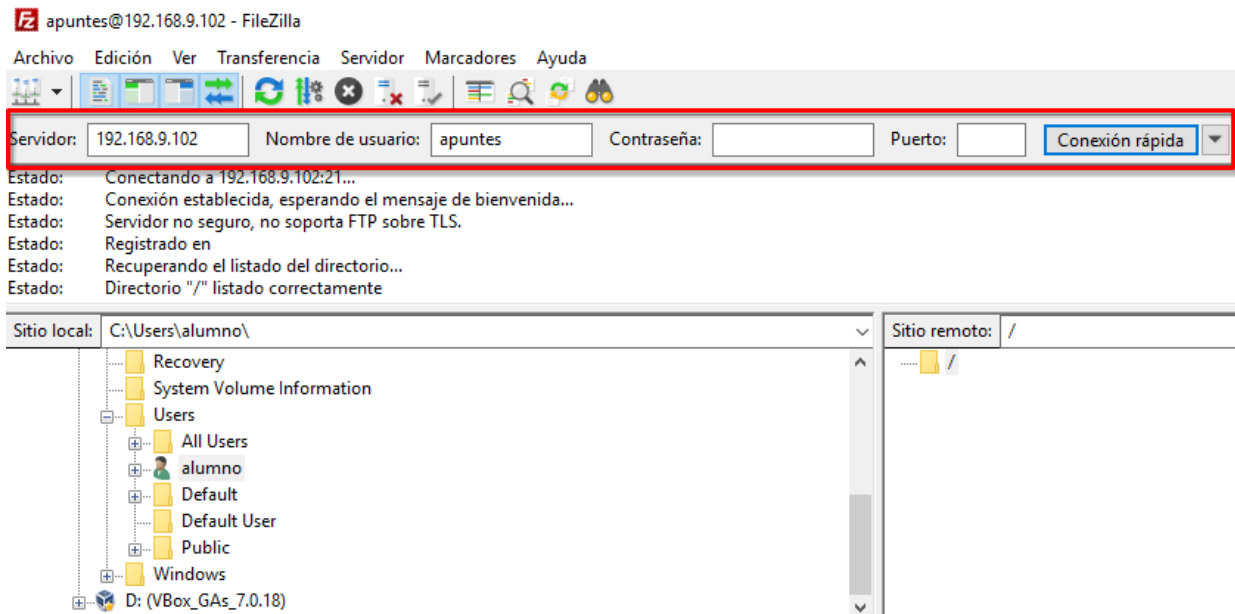
Ahora le tendremos que asignar los permisos que deseemos sobre este directorio a este usuario. En mi caso le voy a dar todos, ya que es su propio directorio. Podríamos dejar menos para un directorio compartido o anónimo. Dados los permisos, tenemos que presionar sobre la opción de **Set as home dir** para que sea considerado su \$HOME y sea donde conecte el usuario por defecto



1.3.3. Conexión desde un cliente

Ahora vamos a comprobar si funciona el servidor conectándonos desde un cliente tanto *Windows 10 LTSC* como GNU/Linux.

1. Conexión desde un cliente *Windows* Para conectarnos desde un *Windows 10* cliente lo primero que tendremos que hacer será tener instalado el programa *Filezilla* cliente en nuestra máquina cliente.



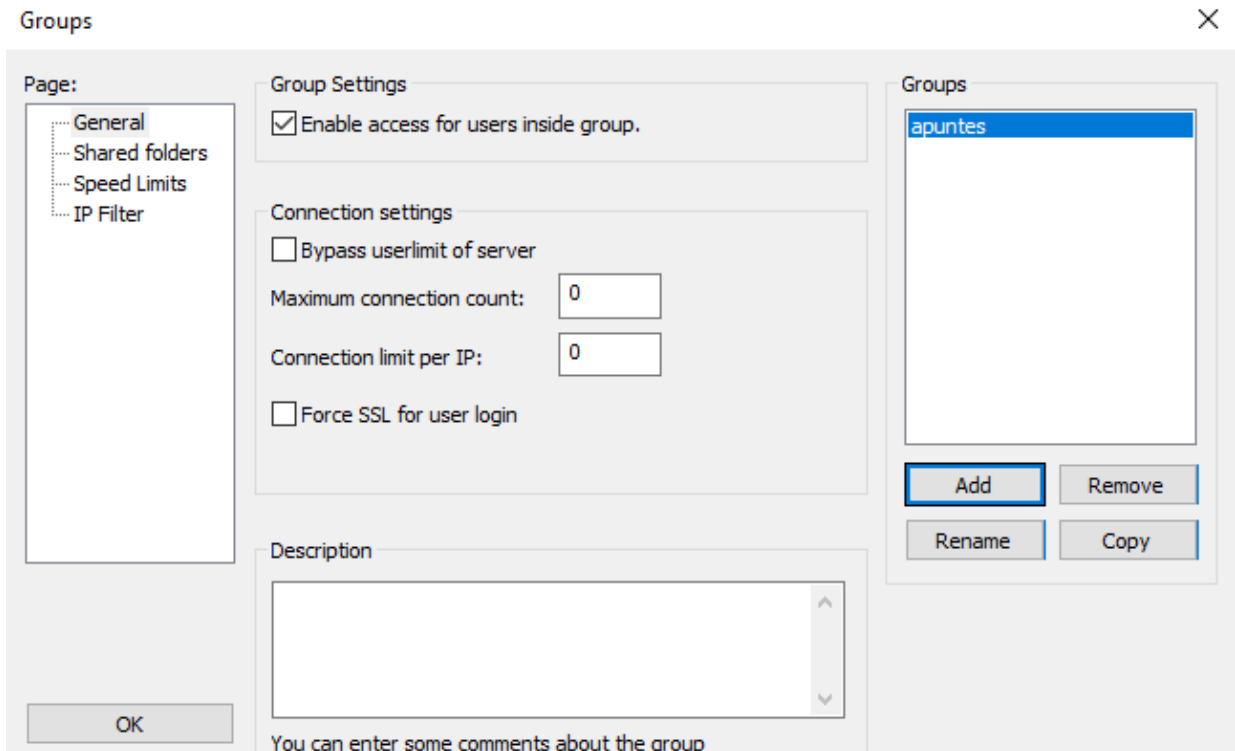
2. Conexión desde un cliente GNU/Linux Para conectarnos desde un cliente GNU/Linux (en mi caso una máquina Fedora 39), lo que tendremos que hacer será usar la instrucción **ftp** desde una terminal de la siguiente manera:

- **ftp <DIRECCION IP SERVIDOR>**

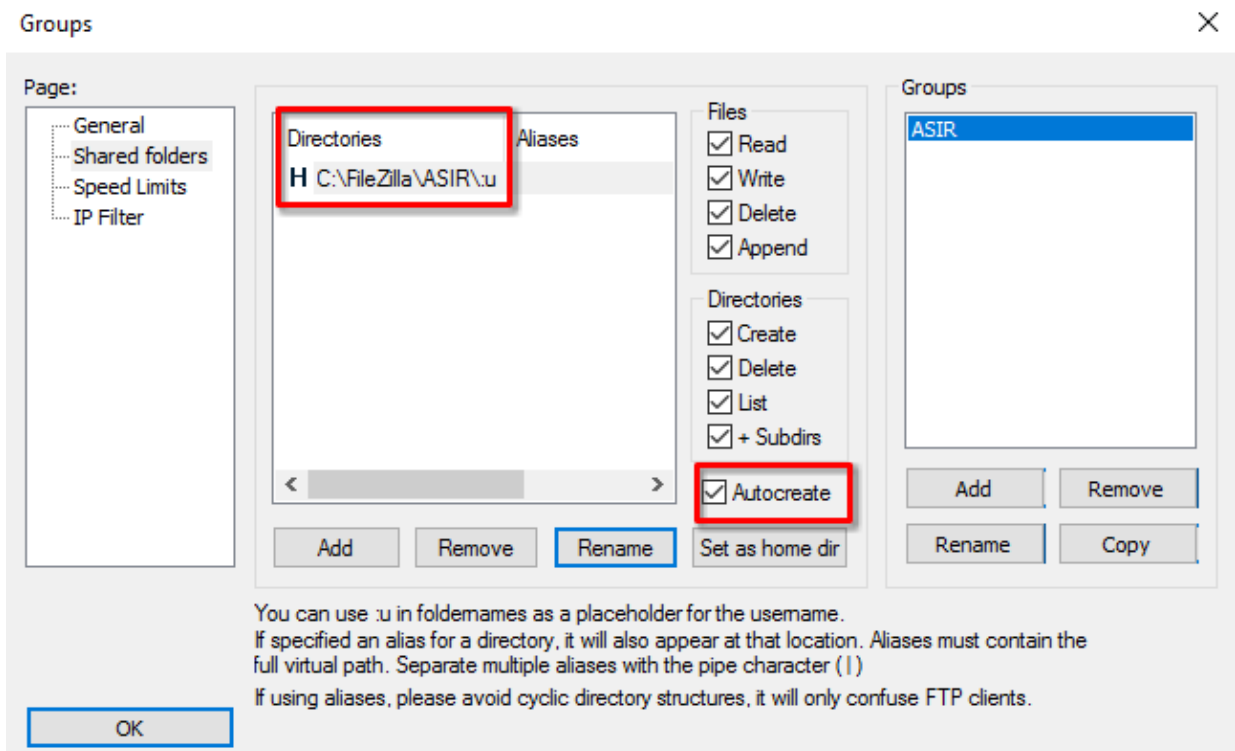
```
maka:~/ $ ftp 192.168.9.102
Connected to 192.168.9.102 (192.168.9.102).
220-FileZilla Server 0.9.51 beta
220-written by Tim Kosse (tim.kosse@filezilla-project.org)
220 Please visit https://filezilla-project.org/
Name (192.168.9.102:maka): apuntes
331 Password required for apuntes
Password:
230 Logged on
Remote system type is UNIX.
ftp>
```

1.3.4. Configuración del servidor

1. Creación de un grupo Ahora vamos a probar a crear un grupo para poder gestionar los usuarios de forma colectiva. El procedimiento es prácticamente igual que crear un usuario, pero presionando sobre el icono que muestra **dos personas**.



También crearemos una carpeta compartida para el grupo, en este caso con una peculiaridad. Después de asignar la carpeta, en este caso la he llamado ASIR, como el grupo, vamos a seleccionar la opción de **Rename**, y le añadiremos un `\u`:. Además pondremos la casilla de **autocreate**. De esta forma le asignamos a cada usuario del grupo que se conecte por FTP una carpeta propia, y además, no tendremos que crearla nosotros, sino que se creará automáticamente en la primera conexión de cada usuario.



2. Limitación de acceso En la pestaña de **edit > Settings** podemos ver en **General settings** dos opciones:

- **IP BINDINGS** → Donde elegimos las direcciones IP por la que tenemos disponible el servidor FTP. Si ponemos un * significará que ofrecemos el servicio por todas las direcciones IP disponibles.
- **IP FILTER** → Para filtrar las IPs que pueden acceder al servidor. Se puede filtrar por IP concreta (sin máscara de subred) o por rangos IP (subredes con máscaras de subred)

1.3.5. Conexión segura por SSL

Para hacer segura la conexión del servidor, de nuevo en *edit > settings*, ahora en el apartado de **FTP over TLS settings**.

En este caso presionaremos sobre **Generate new certificate** y rellenaremos los campos. En el *common name*, según el cliente FTP que usemos, será necesario poner el nombre que usaremos en el momento de establecer la conexión.

This dialog will help you to create a new private key and a self-signed certificate, needed by FileZilla Server to accept TLS connections.

Please fill out the required information. Wrong or missing information may confuse clients.

Key size: ☐ 1280 bit ☒ 2048 bit ☐ 4096 bit

2-Digit country code:

Full state or province:

Locality (City):

Organization:

Organization unit:


Contact E-Mail:

Common name (Server address):

Save key and certificate to this file:

Generating the certificate may take some time depending on the key size.

Al usar esta creación de certificado se crea solo un fichero *certificate.crt* que contiene tanto el certificado como la clave privada.



 certificate: Bloc de notas

Archivo Edición Formato Ver Ayuda

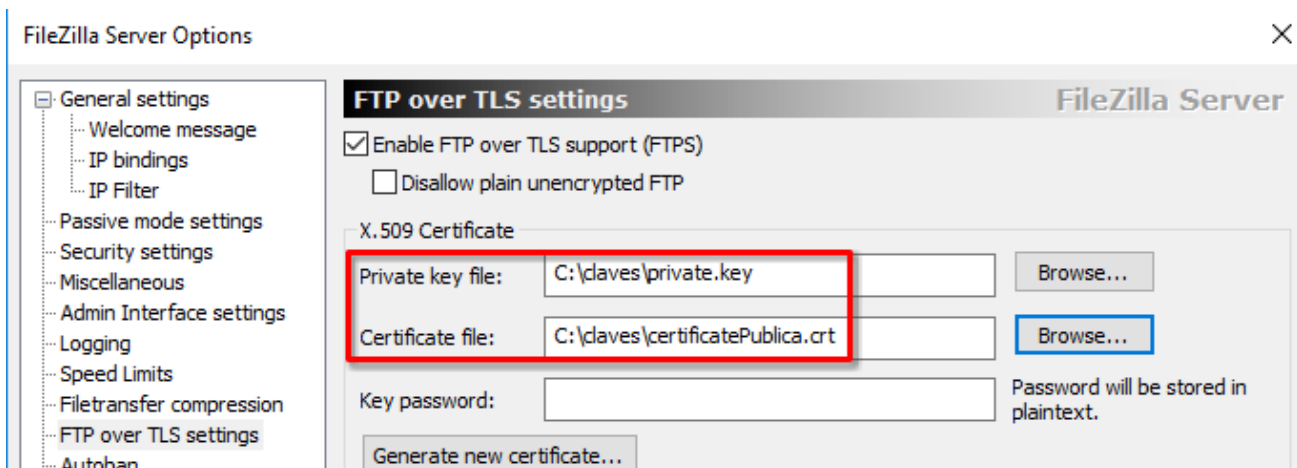
```
-----BEGIN RSA PRIVATE KEY-----
MIIeowIBAAKCAQEAqvBf0+rkvCRaAzo0CcIb89yjpDPqGy8y/aYhLZSPXx1KbXZa
CCvI6SkGT0Jb7GgCgJlXIDvBpgGef9uQGtValt2C7pnGnUGH35bcp6FDqvGy7tR8
VY2D46QW4NKmRnu+FS/iH7AycbUW32i/LVMB9/i+uLrJeZEdZwLKETyQmn4WZB1b
0sMHVGZvzR+5Rg9BAKIOePq42uxqURB3whVTOQHqoB4PXU/QRmTv2aoNbE4Ft83z
XA7kP61kfU4ngs1QQbS377H4XoJFhTgYpmzmsP6gszkNrtY0FqaQsn+PWG67vX28
i80jEkCumXP20KSZYaGEdTGE3kqHQ1BdOm76hwIDAQABAoIBAAZi1CwYyhKMkxAr
zGXQ61phk3s+tS/uw2jWG8coFXLoFS+hjBtiB08uSR4FoI1kGTPUhnvdvs4EoXcF
LIzUC1pUy989vz2AhSLE79kTHQP1o0lohnxrNNGT/4bGV4+a5a7E6x1j53531TLj
JsD6qMgqcnVUBNKVqRN/9yxNptkebPOGAYsZhG0UCR1Lz05r/LqxFasJxiU1q6aG
NdKM9hItJsN3+/AV3ngnyOyx0zFUeVLfNAzYeivvr7bDRr91Sqsj43JuDSi07cpy
WtNB5W9x1cuOenRKdwh21j7RiTAgf86tVZ1t/nPEtb+HMB14qu7MHWf6yxWiZCI
b61bugECgYEA2gH/y1gYzr4d0TOXq+FSThB0WJxc1eL5h/q2+fbwiSx1naP697Wm
84pMoERh3WjLkqnW3BN/NWYHITIEm4uJFHqrjh0FwR1XCk8gmN6MvFBCNMAPDF4P
PcVbepk+2pFRsjrt/65Gh340fJoDi/s26MbKgGf9zjYj52HgKDFne/cCgYEAyLp8
Z5zoLdus9Z+2AJxYKw4XFMDJRCCLD+11lfmzyHdLnryvtBeTQwG3JvVEBD4bRRwh
3s7H1zmqvm+vYHPPxRS7pmgi+Wt9qzfu23dk7KXEDJMg4km0t+8/8Fut1N3Z91C
SvcajWf0iSXGZwyhPPwca/dpLCwnrjm3TD8/MfECgYAi0DQdh/uE4CSYKpgCu8ST
EAz0y1rnm9QN0xZjkBZbgcYK9nVbUoEdMzUp5vHxnhRuNgikZ0inyhcOZ96DusBO
```

Aquí podemos verlo. Será necesario sacar las dos partes a dos ficheros separados, uno *private.key* con la clave privada y otro *certificatePublica.crt* con el certificado.

Este equipo > Disco local (C:) > claves

Nombre	Fecha de modifica...	Tipo	Tamaño
 certificatePublica.crt	16/02/2025 16:02	Certificado de seg...	2 KB
 private.key	16/02/2025 16:02	Archivo KEY	2 KB

Y ahora tendremos que indicar en la configuración la ruta de los ficheros.



Cuando nos conectemos desde un cliente mediante FTPS, como el certificado no está firmado por una entidad verificadora, sino que lo hemos hecho a nivel local de una forma casera, será necesario cancelar la verificación del certificado. Para ello, una vez hayamos establecido la conexión habrá que escribir lo siguiente:

- `set ssl:verify-content no`

En caso de querer hacer esta configuración de manera permanente lo que tendremos que hacer será modificar el fichero `/etc/lftp.conf` y poner en el final las siguientes líneas

```
set dns:order "inet6 inet"
set ssl:verify-certificate no
```

1. Comprobación de conexión mediante FTPD Tendremos que realizar una conexión mediante `lftp`

```
maka:~/ $ lftp -u apuntes ftps://192.168.9.102
Password:
lftp apuntes@192.168.9.102:~>
```

2. Comprobación de firma del certificado en una máquina GNU/Linux

- `openssl x509 -in certificatePublica.crt -noout -fingerprint -[sha256|sha1]`

2. SSH

2.1. Telnet

- El paquete de `telnet` es `telnetd`

2.2. Ficheros relevantes

- `/etc/ssh/ssh_config`: fichero de configuración de parámetros del cliente
- `/etc/ssh/sshd_config`: fichero de configuración de parámetros del servidor
- `$HOME/.ssh/authorized_keys`: fichero que contiene las claves públicas de los hosts autorizados a conectarse por autenticación de clave pública
- `$HOME/.ssh/known_hosts`: guarda las conexiones que se han realizado a servidores `ssh` una vez aceptadas las *fingerprints*
- `$HOME/.ssh/config`: nos permite configurar ciertos parámetros de los que se definen en el fichero `/etc/ssh/ssh_conf` pero de manera individual para el usuarios que lo usa y para la conexión concreta especificada

2.3. Instalación del servidor openssh

Para instalar un servidor `ssh` lo que tendremos que hacer será hacer uso de la instrucción:

- `sudo apt-get install openssh-server`

En caso de querer instalar un cliente `ssh` tendremos que hacer uso de la instrucción:

- `sudo apt-get install openssh-client`

Luego podremos establecer una conexión al servidor desde un cliente ejecutando la instrucción

- `ssh nombreUsuario@DireccionIPMaquin`

```
maka@ubuntu3:~$ ssh maka@172.20.10.5
The authenticity of host '172.20.10.5 (172.20.10.5)' can't be established.
ECDSA key fingerprint is e6:cf:d5:13:d4:cf:7e:b7:9a:01:6c:86:c6:23:6f:75.
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added '172.20.10.5' (ECDSA) to the list of known hosts.
maka@172.20.10.5's password:
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 4.4.0-142-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Your Hardware Enablement Stack (HWE) is supported until April 2019.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

maka@ubuntu2:~$
```

2.4. Ver la dirección en known_hosts en vez de el hash

Ahora lo que haremos será hacer configuraciones para que en vez de ver un **hash** cuando miremos el fichero `$HOME/.ssh/known_hosts` veámos la dirección IP que se ha conectado a nuestro servidor.

Primero antes de hacer nada lo que haremos será mirar en la máquina servidor el fichero `$HOME/.ssh/known_hosts` para ver si tenemos algún registro

```
maka@ubuntu2:~/.ssh$ cat known_hosts
|1|VJm5fgA1jJBxCHzMjP+651QexPY=|SsSzyK9hTHUzxSpDMeWhlGgznuK= ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbGljbGZCB2ErfSGex08oq0JQtLUXfLYjruHe8h8XBy7UKVJg=
```

Lo que se puede apreciar arriba es la entrada que tenemos en el fichero `$HOME/.ssh/known_hosts` de nuestra máquina servidor después de haber realizado un **ssh** desde nuestra máquina cliente.

Para ver la dirección IP que se ha conectado a nosotros en vez de ese **hash** lo que tendremos que hacer será, en nuestra máquina servidor, modificar el fichero `/etc/ssh/ssh_config` en la línea que contenga **HashKnownHosts** de tal manera que su valor sea **no**.

```
# RekeyLimit 1G 1h
SendEnv LANG LC_*
HashKnownHosts no
```

NO FUNCIONA

2.5. Eliminar error intento ataque Ddos

Muchas veces vamos a tener que eliminar entradas de nuestro fichero `known_hosts` para que al cliente le vuelva a aparecer el mensaje del *fingerprint*, para hacer esto tendremos que usar la instrucción:

- `ssh-keygen -f $HOME/.ssh/known_hosts -R DIRECCIONIPCLIENTE`

2.6. Conexión por clave pública

Lo primero que tendremos que hacer será generar un par de claves pública-privada en el cliente.

Para generar estas claves tendremos que hacer uso de la instrucción `ssh-keygen` de la siguiente manera:

- `ssh-keygen -t rsa`

```
maka@ubuntu201:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/maka/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/maka/.ssh/id_rsa.
Your public key has been saved in /home/maka/.ssh/id_rsa.pub.
The key fingerprint is:
30:c2:99:66:79:3b:1a:93:d0:aa:13:67:65:7d:47:20 maka@ubuntu201
The key's randomart image is:
+--[ RSA 2048]-----+
|      E ...      |
|    o = . .      |
|   . % = . .      |
|  B + = .        |
|. + + o S        |
| =   + .         |
|o   .            |
| .               |
|                 |
+-----+
```

Las claves se almacenarán en `$HOME/.ssh/`

```
maka@ubuntu201:~$ ls -la .ssh/
total 20
drwx----- 2 maka maka 4096 feb 18 18:35 .
drwxr-xr-x 19 maka maka 4096 feb 18 18:32 ..
-rw----- 1 maka maka 1679 feb 18 18:35 id_rsa
-rw-r--r-- 1 maka maka 396 feb 18 18:35 id_rsa.pub
-rw-r--r-- 1 maka maka 444 dic 9 09:39 known_hosts
```

Luego de haber creado las claves lo que tendremos que hacer será enviar la **clave pública** al servidor, en mi caso lo haré mediante la instrucción `scp`

- `maka@ubuntu201:~$ scp .ssh/id_rsa.pub ana@172.20.10.5:/home/ana/`

Una vez tengamos la clave pública del cliente en el servidor lo que tendremos que hacer será añadir la clave al fichero `$HOME/.ssh/authorized_keys`

AVISO

Puede que el fichero `authorized keys` no exista. En ese caso lo que haremos será renombrar la clave pública con el nombre de `authorized key`

```
root@ubuntu2:/etc/ssh# cd /home/ana/.ssh/
root@ubuntu2:/home/ana/.ssh# ls -la
total 12
drwx----- 2 ana ana 4096 feb 18 18:55 .
```

```
drwxr-xr-x 4 ana ana 4096 feb 18 18:41 ..
-rw-r--r-- 1 ana ana 396 feb 18 18:38 authorized_keys
-rw-r--r-- 1 ana ana 0 feb 18 18:26 known_hosts
```

Ahora tendríamos que poder acceder desde la máquina cliente a la máquina servidor sin tener que escribir ningún tipo de *password*

```
maka@ubuntu201:~$ ssh ana@172.20.10.5
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 4.4.0-142-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2019.
Last login: Tue Feb 18 18:42:54 2025 from 172.20.10.2
ana@ubuntu2:~$
```

Con la opción `-v` de la instrucción `ssh` podríamos ver el proceso en el cuál se leen las claves. La opción que nos daría la mejor información de lo que está pasando sería

- `debug1: Offering RSA public key: /home/maka/.ssh/id_rsa`

En caso de tener más de una clave en nuestro cliente tendríamos que usar la opción `-i` de la instrucción `ssh` de la siguiente manera:

- `ssh -i «NOMBRE-CLAVE-PRIVADA» usuario@destino`

2.6.1. Configuración del servidor

Podemos habilitar o deshabilitar la conexión mediante claves público-privada con la opción `PubKeyAuthentication` en el fichero `/etc/ssh/sshd_config`

2.7. Directorio `$HOME/.ssh/`

Es un directorio el cuál se crea solo en la `$HOME` del usuario con el que estés conectando. Para añadir ciertos parámetros de configuración, debemos crear un fichero llamado `config` y ahí podemos poner los parámetros que queramos.

```
Host 172.20.10.5-rsa
    Hostname 172.20.10.5
    HostKeyAlgorithms ssh-rsa
    ForwardX11 yes
    CheckHostIp no
    HostKeyAlias 172.20.10.5-rsa
    Port 2222
    User ana
```

- `Hostname 172.20.10.5`: Dirección IP a la que se conecta
- `HostKeyAlgorithms ssh-rsa`: El cliente solo usará el algoritmo `ssh-rsa` para autenticarse con el servidor

- **ForwardX11 yes:** Habilita el reenvío de las X11, permitiendo ejecutar aplicaciones gráficas en el servidor viéndolas en el cliente
- **CheckHostIp no:** Evita que `ssh` verifique si la IP del servidor ha cambiado
- **HostKeyAlias 172.20.10.5-rsa:** usa 172.20.10.5-rsa como alias para la clave del host en `~/.ssh/known_hosts`
- **Port 2222:** En lugar del puerto estándar (22), usará el puerto 2222
- **User ana:** Especifica el usuario con el que hará *login* en el servidor

2.8. Algoritmo de cifrado

El algoritmo por defecto es `ecdsa-sha2-nistp256`, pero lo podemos modificar estableciendo la conexión de la siguiente manera:

- `ssh -o HostKeyAlgorithms=ssh-rsa ana@172.20.10.5`

Si no eliminamos el contenido del fichero `$HOME/.ssh/known_hosts` nos aparecerá el siguiente error

```

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
ec:ed:1e:48:f1:8d:22:4e:e2:dc:58:e1:c6:b0:f7:d0.
Please contact your system administrator.
Add correct host key in /home/maka/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /home/maka/.ssh/known_hosts:3
  remove with: ssh-keygen -f "/home/maka/.ssh/known_hosts" -R 172.20.10.5
RSA host key for 172.20.10.5 has changed and you have requested strict checking.
Host key verification failed.

```

Lo único que tendremos que hacer será ejecutar la instrucción `echo "" > known_hosts` para eliminar el contenido del fichero y ya nos podremos conectar y encima sin contraseña porque tenemos la clave pública en el servidor

```

maka@ubuntu201:~/.ssh$ echo "" > known_hosts
maka@ubuntu201:~/.ssh$ ssh -o HostKeyAlgorithms=ssh-rsa ana@172.20.10.5
The authenticity of host '172.20.10.5 (172.20.10.5)' can't be established.
RSA key fingerprint is ec:ed:1e:48:f1:8d:22:4e:e2:dc:58:e1:c6:b0:f7:d0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.20.10.5' (RSA) to the list of known hosts.
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 4.4.0-142-generic x86_64)

* Documentation:  https://help.ubuntu.com/

New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2019.
Last login: Tue Feb 18 18:57:13 2025 from 172.20.10.2

```

2.9. SSH-Agent

Es un programa que gestiona las claves privadas `ssh` en segundo plano, evitando que tengas que ingresar la contraseña de tu clave privada cada vez que te conectas a un servidor, incluso evita tener que ingresar la clave de paso.

Vamos a generar una clave que tenga frase de paso para comprobar que esto es cierto.

Para generar la clave de paso tendremos que hacer uso de la instrucción `ssh-keygen -t rsa`

```
maka@ubuntu201:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/maka/.ssh/id_rsa): id_rsa_frase
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_rsa_frase.
Your public key has been saved in id_rsa_frase.pub.
The key fingerprint is:
6c:33:c5:1d:58:28:32:17:3a:c1:30:8c:48:62:dc:07 maka@ubuntu201
The key's randomart image is:
+--[ RSA 2048 ]-----+
|+.E+o. .. +o      |
|+...o+oo.o. .    |
|   . o+ .o .     |
|       o .       |
|       S        |
|       . o       |
|                 |
|                 |
|                 |
+-----+
```

Ahora tendremos que ejecutar la instrucción `ssh-agent` y añadiremos la nueva frase de paso

```
maka@ubuntu201:~$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-Fh3MJbVg9pL/agent.4470; export SSH_AUTH_SOCK;
SSH_AGENT_PID=4471; export SSH_AGENT_PID;
echo Agent pid 4471;
maka@ubuntu201:~$ ssh-add id_rsa_frase
Enter passphrase for id_rsa_frase:
Identity added: id_rsa_frase (id_rsa_frase)
```

Con la instrucción `ssh-add -l` listamos las frases de paso que tengamos

```
maka@ubuntu201:~$ ssh-add -l
2048 6c:33:c5:1d:58:28:32:17:3a:c1:30:8c:48:62:dc:07 id_rsa_frase (RSA)
2048 30:c2:99:66:79:3b:1a:93:d0:aa:13:67:65:7d:47:20 maka@ubuntu201 (RSA)
```

Ahora lo que tendremos que hacer será enviar la clave pública al servidor, en mi caso seguiré usando la instrucción `scp`

- `maka@ubuntu201:~$ scp id_rsa_frase.pub ana@172.20.10.5:/home/ana`

Una vez tengamos la clave pública nueva en la máquina servidor lo que haremos será ejecutar una instrucción la cuál lee el contenido del fichero de la clave pública y lo añade al fichero de `$HOME/.ssh/authorized_keys`

```
■ cat id_rsa_frase.pub » .ssh/authorized_keys
```

Y ya nos podríamos conectar usando la opción `i` con esa frase de paso

```
maka@ubuntu201:~$ ssh -i id_rsa_frase ana@172.20.10.5
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 4.4.0-142-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2019.
Last login: Tue Feb 18 19:19:15 2025 from 172.20.10.2
ana@ubuntu2:~$
```

2.10. Windows: ssh, claves y ssh-agent

Para instalar el servidor OpenSSH en una máquina Windows 10 la ruta a seguir es:

1. Ajustes
2. Aplicaciones
3. Aplicaciones y características
4. Administrar funciones opcionales
5. Agregar una característica

El `ssh-agent` se habilita en **Servicios** > *OpenSSH Authentication Agent*. Lo pondremos en automático

Propiedades: OpenSSH Authentication Agent (Equipo local)



General Iniciar sesión Recuperación Dependencias

Nombre de servicio: ssh-agent

Nombre para mostrar: OpenSSH Authentication Agent

Descripción: Agent to hold private keys used for public key authentication.

Ruta de acceso al ejecutable:
C:\Windows\System32\OpenSSH\ssh-agent.exe

Tipo de inicio: Automático

Para generar un par de claves público-privada en Windows se hará con la misma instrucción usada en GNU/Linux desde el CMD

```
C:\Users\alumno>ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\alumno/.ssh/id_rsa): is_rsa_win
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in is_rsa_win.
Your public key has been saved in is_rsa_win.pub.
The key fingerprint is:
SHA256:sQbRnhD0pIhVRq/8GkuAWyvbQ+6zG3OJw38mm/mKC+M alumno@win10-ltsc
The key's randomart image is:
+---[RSA 2048]-----+
|  .O+ +O          |
| O O *...         |
| . . . =O..       |
| . . . .OO        |
| . O O   S        |
| +. + O .         |
| =OB = .          |
| .*+O.*O          |
| .EB*OB.          |
+----[SHA256]-----+
```

Luego le daremos la clave pública al servidor mediante la instrucción `scp`

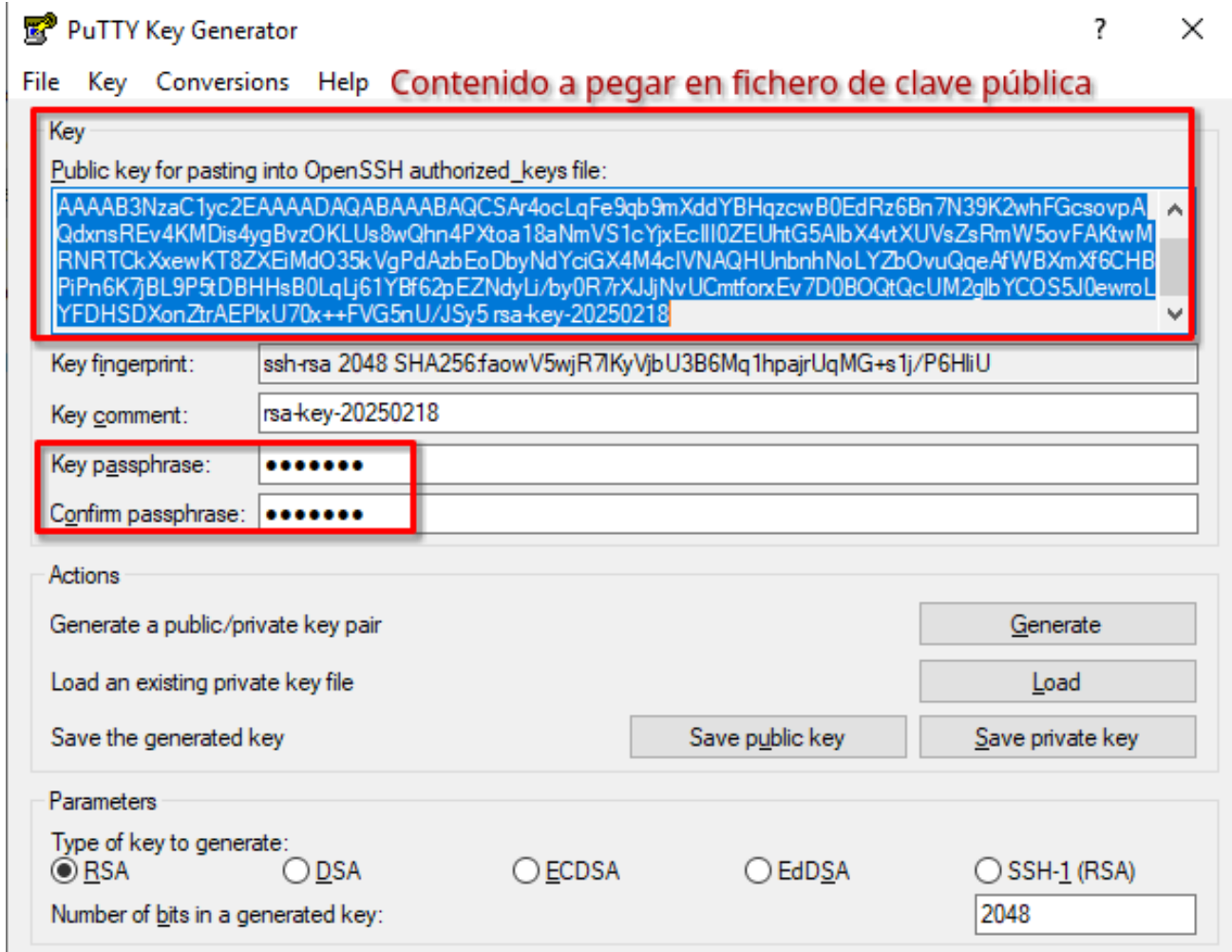
Luego en el servidor lo que haremos será añadir el contenido de la clave pública al fichero `/.ssh/authorized_key`

Ya nos podremos conectar sin ningún tipo de problema.

2.10.1. Putty

Lo primero que haremos será instalar el *software* putty

Una vez instalado el *software* lo que haremos será abrir el programa **PuttyGen** para crear un par de claves



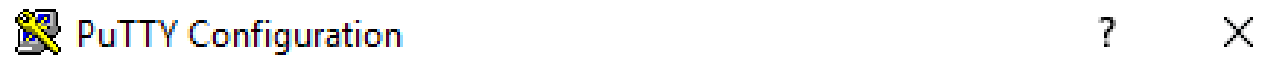
Luego de tener algo parecido a lo que se puede ver en la imagen de arriba lo que haremos será:

1. Añadir la clave de paso
2. Copiar el texto que aparece en rojo porque será el contenido de la clave pública
3. Presionar los botones de *save public key* & *save private key*

Después de esto ya podremos realizar el procedimiento ya conocido de:

1. Enviar la clave pública al servidor mediante `scp`
2. Añadir en el servidor la clave pública al fichero `.ssh/authorized_keys`

En este caso probaré a conectarme a la máquina servidor desde el *software* Putty y no desde el `cmd`.



Category:

- Colours
- Connection**
- Data
- Proxy
- SSH**
- Kex
- Host keys
- Cipher
- Auth**
- Credenti
- GSSAPI
- TTY

Credentials to authenticate with

Public-key authentication

Private key file for authentication:
C:\Users\alumno\Desktop\private-ubunt **Browse...**

Certificate to use with the private key (optional):
 Browse...

Plugin to provide authentication responses

Plugin command to run



Category:

- Colours
- Connection**
- Data**
- Proxy
- SSH**
- Kex
- Host keys
- Cipher
- Auth**
- Credenti
- GSSAPI
- TTY

Data to send to the server

Login details

Auto-login username

When username is not specified:
☒ Prompt ☐ Use system username (alumno)

Terminal details

Terminal-type string

Terminal speeds

```
ana@ubuntu2: ~  
Using username "ana".  
Authenticating with public key "rsa-key-20250218"  
Passphrase for key "rsa-key-20250218":  
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 4.4.0-142-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com/  
  
New release '16.04.7 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Your Hardware Enablement Stack (HWE) is supported until April 2019.  
Last login: Tue Feb 18 20:57:45 2025 from 172.20.10.6  
ana@ubuntu2:~$
```

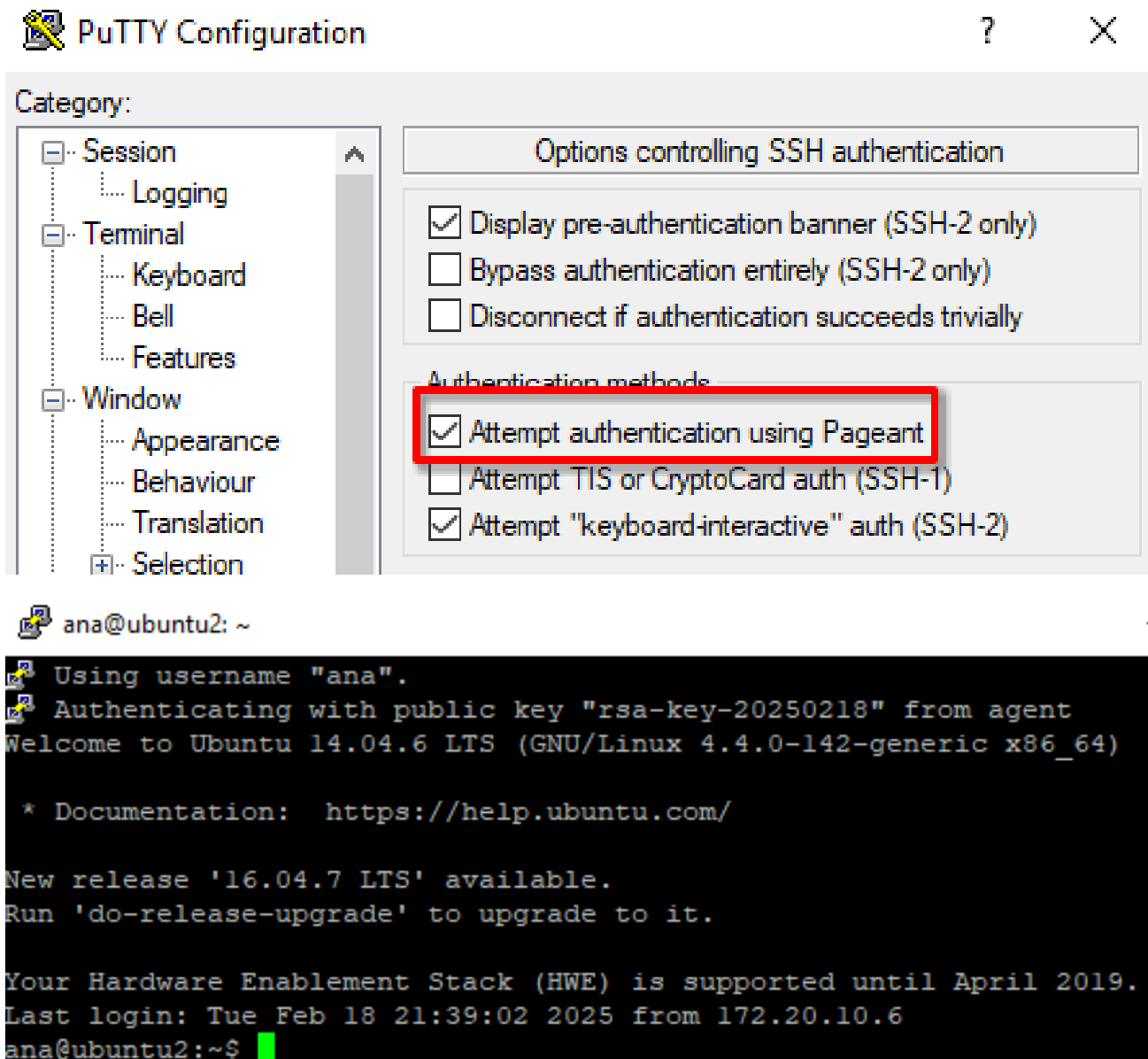
Se puede apreciar en la última imagen de arriba que aún nos está pidiendo la frase de paso. Para evitar esto lo que tendremos que hacer será abrir el programa **Pageant** y añadir el agente.

Cuando presionemos sobre **Pageant** en el buscador de *windows* no ocurrirá nada especial, pero .ªparecerá"lo siguiente



Lo que haremos será hacer clic derecho sobre ese icono >añadir clave y añadiremos nuestra clave privada.

Luego en la configuración de Putty lo que haremos será en conexión >ssh >auth, marcar la casilla de autenticarse usando pageant



Como se puede apreciar ya no se nos pide la frase de paso para acceder al usuario ana.

2.10.2. MobaXterm

Para generar un par de claves público-privadas en mobaxterm lo que hay que hacer es ir a *tools > MobaKeyGen*

La interfaz es casi igual a la de putty por lo que habrá que hacer clic en **generar claves** y mover el ratón.

En este caso no vamos a crear ningún par de claver pero lo que si que vamos a hacer es realizar una conexión usando este programa.

Session settings

✕



Basic SSH settings

 Remote host *
☒ Specify username
 Port

Advanced SSH settings

Terminal settings

Network settings

Bookmark settings

☒ X11-Forwarding
 ☒ Compression
 Remote environment:
Execute command: ☐ Do not exit after command endsSSH-browser type: ☐ Follow SSH path (experimental)
☒ Use private key

Expert SSH settings

Execute macro at session start:

OK

Cancel

```

3. 192.168.1.113 (ana)
• MobaXterm Personal Edition v25.0 •
  (SSH client, X server and network tools)

► SSH session to ana@192.168.1.113
  • Direct SSH      : ✓
  • SSH compression : ✓
  • SSH-browser     : ✓
  • X11-forwarding  : ✓ (remote display is forwarded through SSH)

► For more info, ctrl+click on help or visit our website.

Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 4.4.0-142-generic x86_64)

* Documentation: https://help.ubuntu.com/

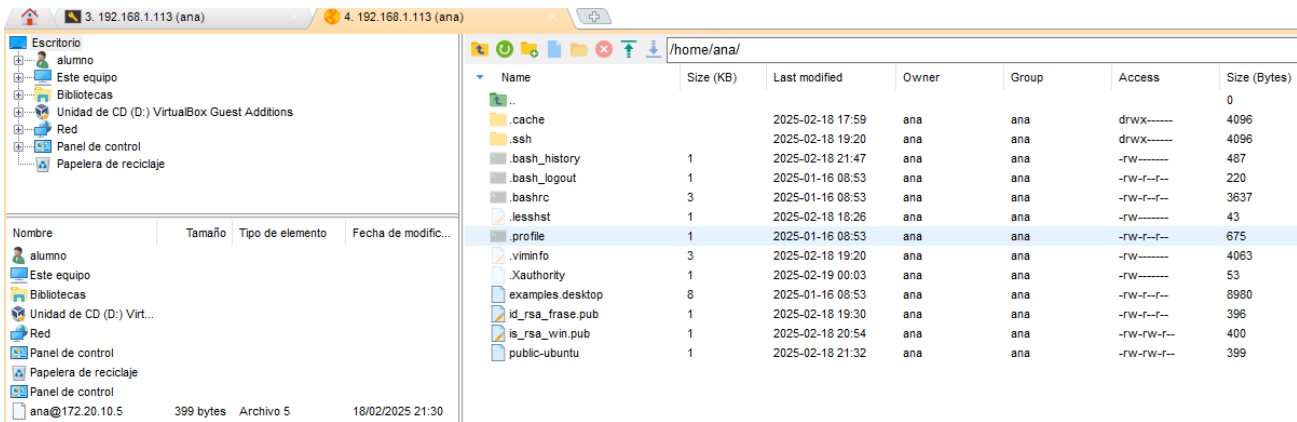
New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2019.
Last login: Tue Feb 18 21:43:59 2025 from 172.20.10.6
/usr/bin/xauth: file /home/ana/.Xauthority does not exist
ana@ubuntu2:~$
  
```

AVISO

Hay que tener iniciado el Pageant para que funcione

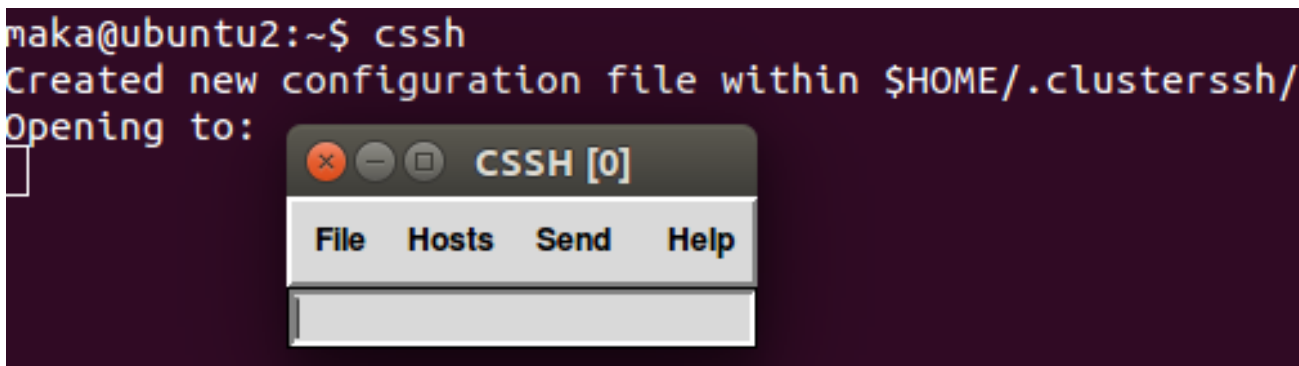
Con MobaXterm también podemos conectarnos de una manera gráfica vamos al apartado de *Session* > **SFTP**

**2.11. Cluster ssh**

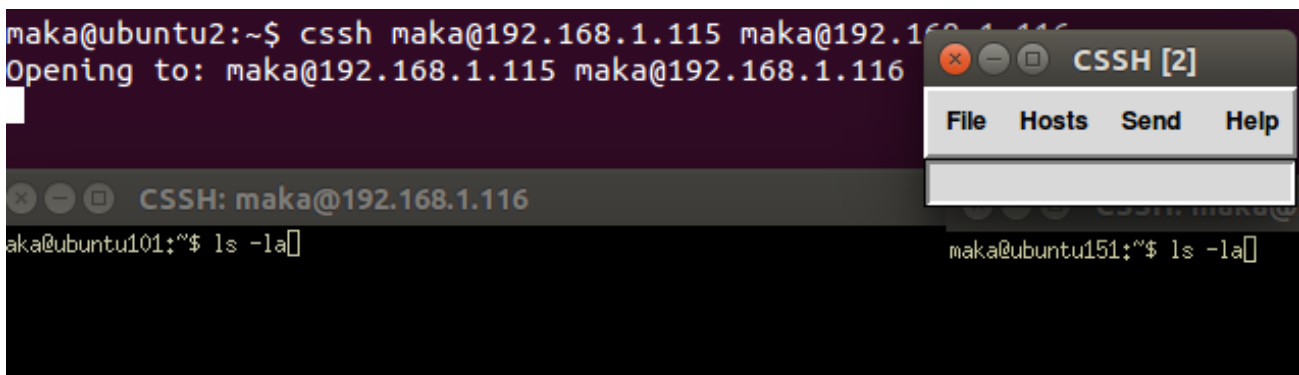
Para instalar `cssh` habrá que ejecutar la instrucción:

- `sudo apt-get install clusterssh`

Ahora ejecutamos el comando `cssh` para crear el directorio y sus ficheros de configuración.



ahora básicamente podremos conectarnos a varias máquinas a la vez, en mi caso me conectaré a otras dos máquinas Ubuntu



Como se puede ver en la imagen de arriba se me han abierto 3 ventanas, 2 más grandes y una pequeña, las dos grandes corresponden a los equipos que estamos conectados y la pequeña es para que escribamos los comandos

que se escribieran al mismo tiempo en las dos ventanas grandes y cuando presionemos la tecla **ENTER** se ejecutará el comando en las dos ventanas grandes.

Si las máquinas tuvieran el mismo usuario podríamos usar el siguiente comando:

```
■ cssh -l maka <ip><ip><ip>
```

2.11.1. TODO Clusters

Para no tener que estar escribiendo cada vez que queramos conectarnos todas la IPs, podemos crear un grupo de clústers para conectarnos directamente a todos los que estén en ese grupo.

Para ello creamos el fichero en `$HOME/maka/.clustersss/clusters`

NO SE SABE

2.11.2. TODO Tags

En justo lo inverso a cluster.

Crearemos el fichero en `$HOME/.clustersssh/tags`

2.12. Entorno gráfico

2.12.1. GNU/Linux

Para poder usar `ssh` con entorno gráfico, tenemos que habilitarlo en el servidor.

Tendremos que modificar el fichero `/etc/ssh/sshd_config` en la línea que contenga `X11Forwarding` `yes`

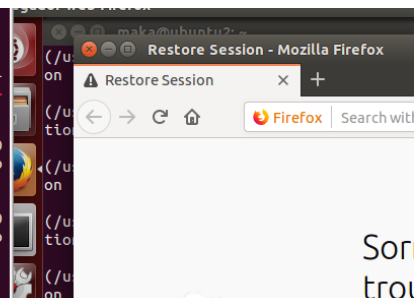
```
X11Forwarding yes
X11DisplayOffset 10
PrintMotd no
PrintLastLog yes
TCPKeepAlive yes
#UseLogin no
```

Ahora para conectarnos tendremos que usar el parámetro `-X` o también lo podemos hacer permanente modificando el fichero `/etc/ssh/ssh_config`

```
Host *
# ForwardAgent no
ForwardX11 yes
```

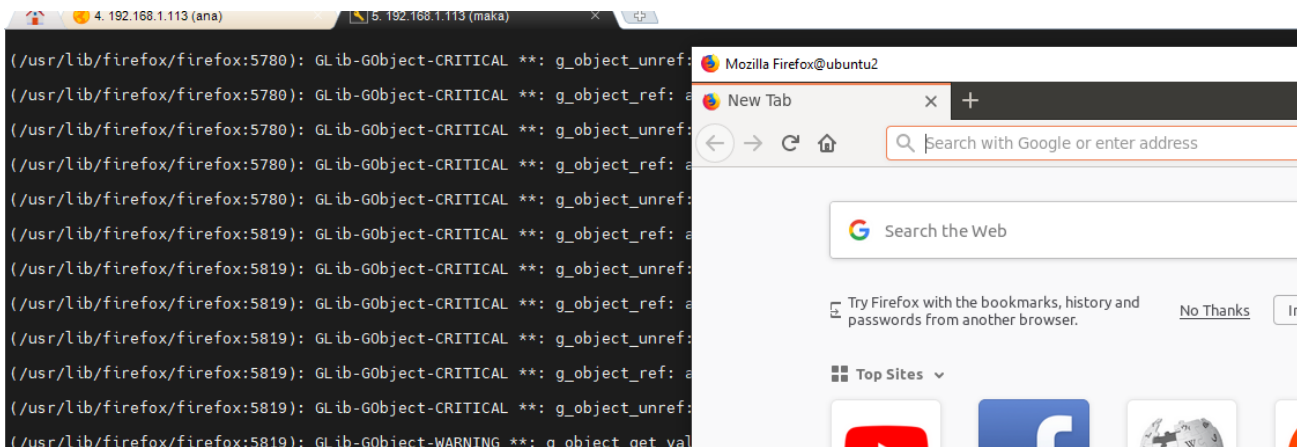
Con esto ya podremos conectarnos y ejecutar comandos de forma gráfica

```
root@ubuntu2:~# ps -ef | grep -i firefox
maka 5332 5314 56 01:03 pts/16 00:00:08 /usr/lib/firefox/firefox
maka 5402 5332 8 01:03 pts/16 00:00:01 /usr/lib/firefox/firefox -contentproc -childID 1
-isForBrowser -prefsLen 1 -prefMapSize 169710 -schedulerPrefs 0001,2 -parentBuildID 2019021500291
0 -greomni /usr/lib/firefox/omni.ja -appomni /usr/lib/firefox/browser/omni.ja -appdir /usr/lib/fi
firefox/browser 5332 true tab
maka 5471 5332 3 01:03 pts/16 00:00:00 /usr/lib/firefox/firefox -contentproc -childID 3
-isForBrowser -prefsLen 5686 -prefMapSize 169710 -schedulerPrefs 0001,2 -parentBuildID 2019021500
2910 -greomni /usr/lib/firefox/omni.ja -appomni /usr/lib/firefox/browser/omni.ja -appdir /usr/lib
/firefox/browser 5332 true tab
maka 5506 5332 1 01:03 pts/16 00:00:00 /usr/lib/firefox/firefox -contentproc -childID 4
-isForBrowser -prefsLen 6480 -prefMapSize 169710 -schedulerPrefs 0001,2 -parentBuildID 2019021500
2910 -greomni /usr/lib/firefox/omni.ja -appomni /usr/lib/firefox/browser/omni.ja -appdir /usr/lib
/firefox/browser 5332 true tab
root 5529 5113 0 01:03 pts/10 00:00:00 grep --color=auto -i firefox
root@ubuntu2:~#
```



2.12.2. Microsoft Windows

En este caso usaremos el programa MobaXterm.



2.13. Tunnelización

La instrucción a usar sería la siguiente:

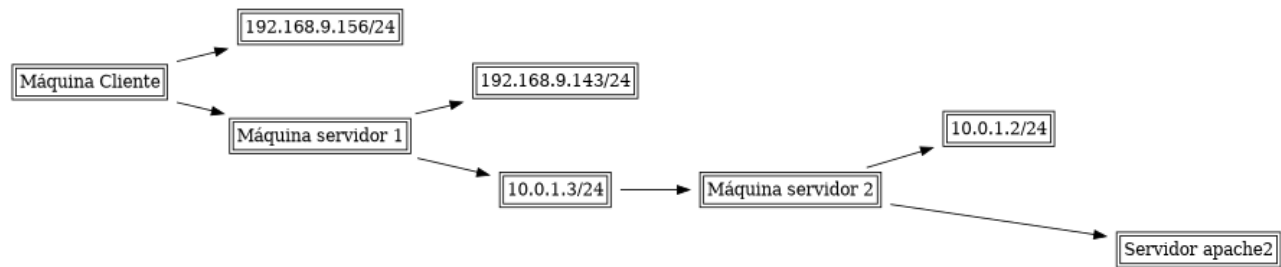
- `ssh -g -L [interfaz]:[puertoLocal]:[direcciónDelServicio]:[puertoServicio] [conexionSshPuente]`

Explicación:

- `-g`: *gateway ports*, permite que otros equipos de la red accedan a nuestro puerto de reenvío. Es decir, que puedan acceder a `http://NUESTRAIP:NUESTROPUERTO`. Si se usa `*` como interfaz de escucha, no sería necesario especificar este parámetro
- `-L`: *local port forwarding*, redirigir el tráfico de un puerto local a otro destino a través de `ssh`
- `[interfazLocal]`: interfaz de red local por la que queremos ofrecer el servicio de la conexión
- `[puertoLocal]`: puerto de la máquina local, es decir, la máquina desde la que se ejecuta el comando, por la que queremos que se acceda al servicio accedido
- `[direcciónServicio]`: dirección de red, ya sea IP o nombre de la máquina en la que se está sirviendo el servicio y al que queremos acceder
- `[puertoServicio]`: puerto por el que se está sirviendo el servicio en la máquina a la que queremos acceder
- `[conexionSSHpuente]`: dirección de la máquina puente a través de la cuál queremos acceder a la máquina que contiene el servicio al que queremos acceder

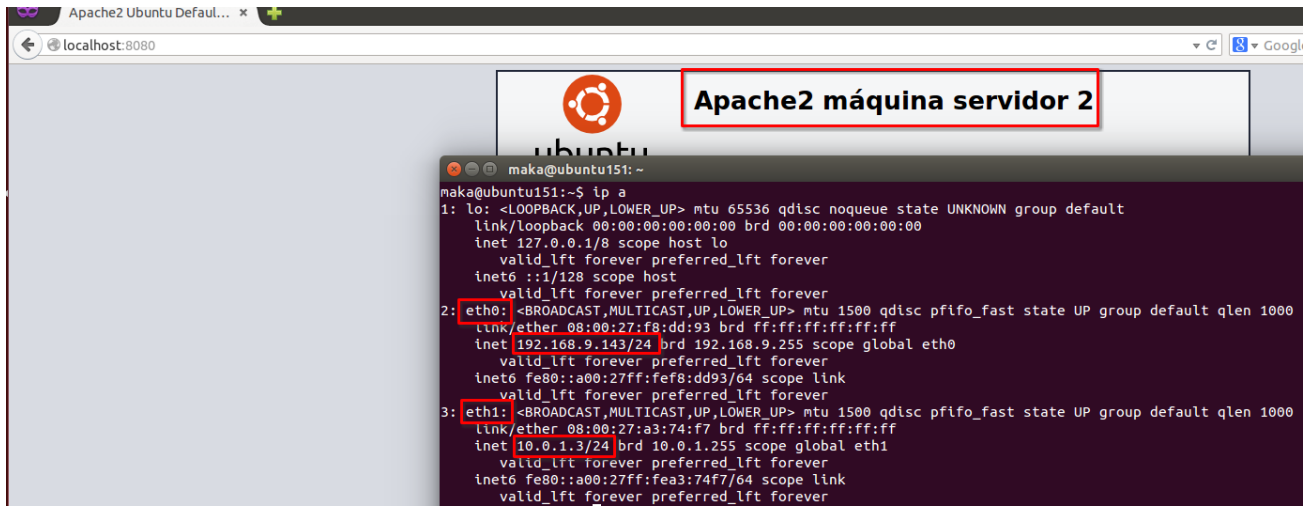
2.13.1. Primer ejemplo

Tenemos un servidor `apache2` y otro servidor con `ssh` además de un cliente, lo que vamos a hacer es pasar el `apache` del servidor 2 mediante el servidor 1 ya que el servidor 2 no está en nuestra red.

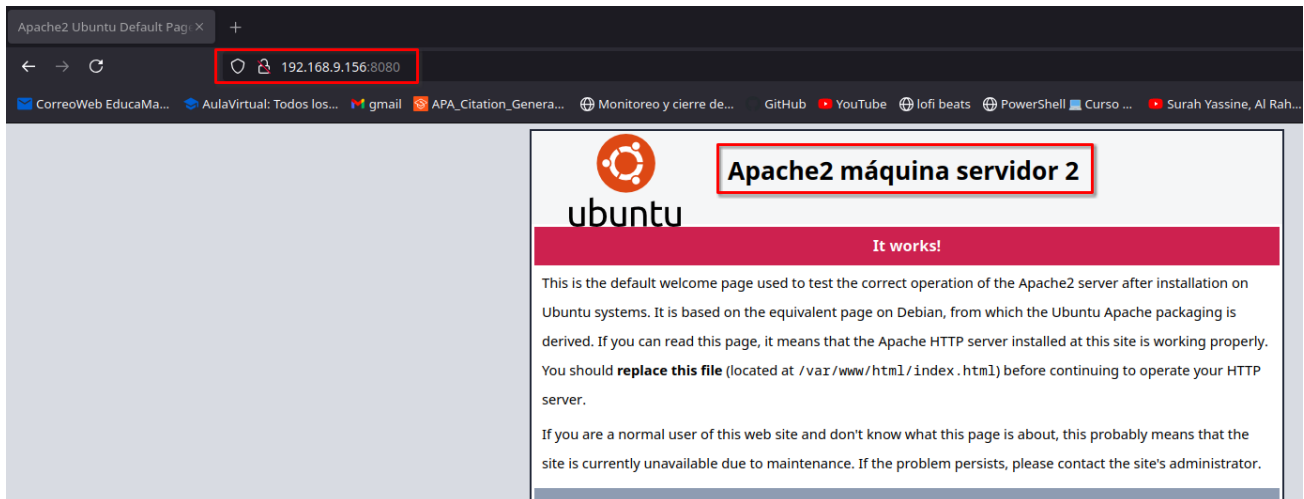


En el cliente ejecutaremos la siguiente instrucción:

- `ssh -L 8080:10.1.1.3 maka@192.168.9.143`



Además si usamos el mismo comando pero con el parámetro `-g` podremos ver el `apache` a través de mi máquina real poniendo la IP de la máquina cliente



2.13.2. Túnel remoto

Es igual que antes pero poniendo la opción `-R`, de esta forma estaríamos sirviéndolo en la máquina que estamos usando de "puente"

- `ssh -R localhost:8080:10.0.1.2:80 maka@192.168.1.143`

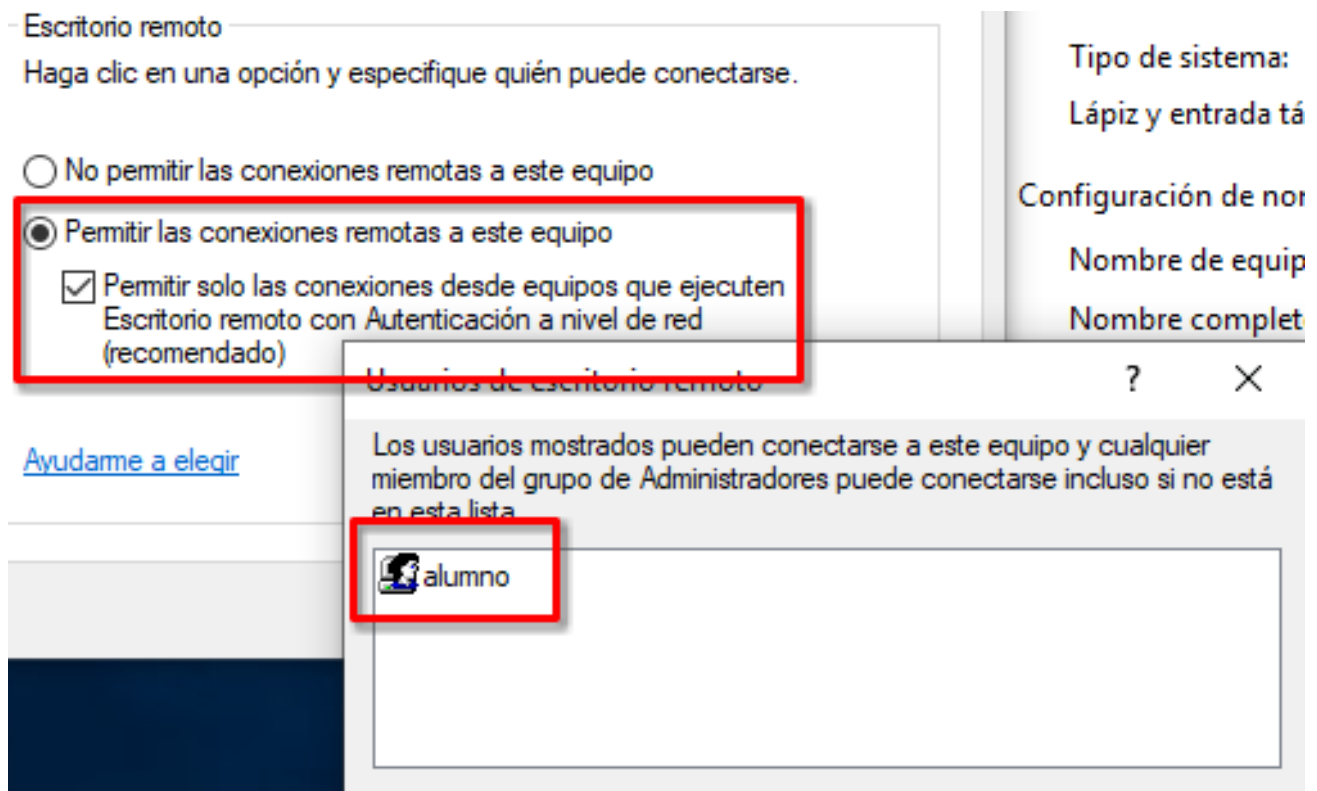
Con este comando estamos sirviendo el contenido de la página pero en vez de en mi máquina cliente que ejecuta el comando del servidor

2.14. Escritorio remoto

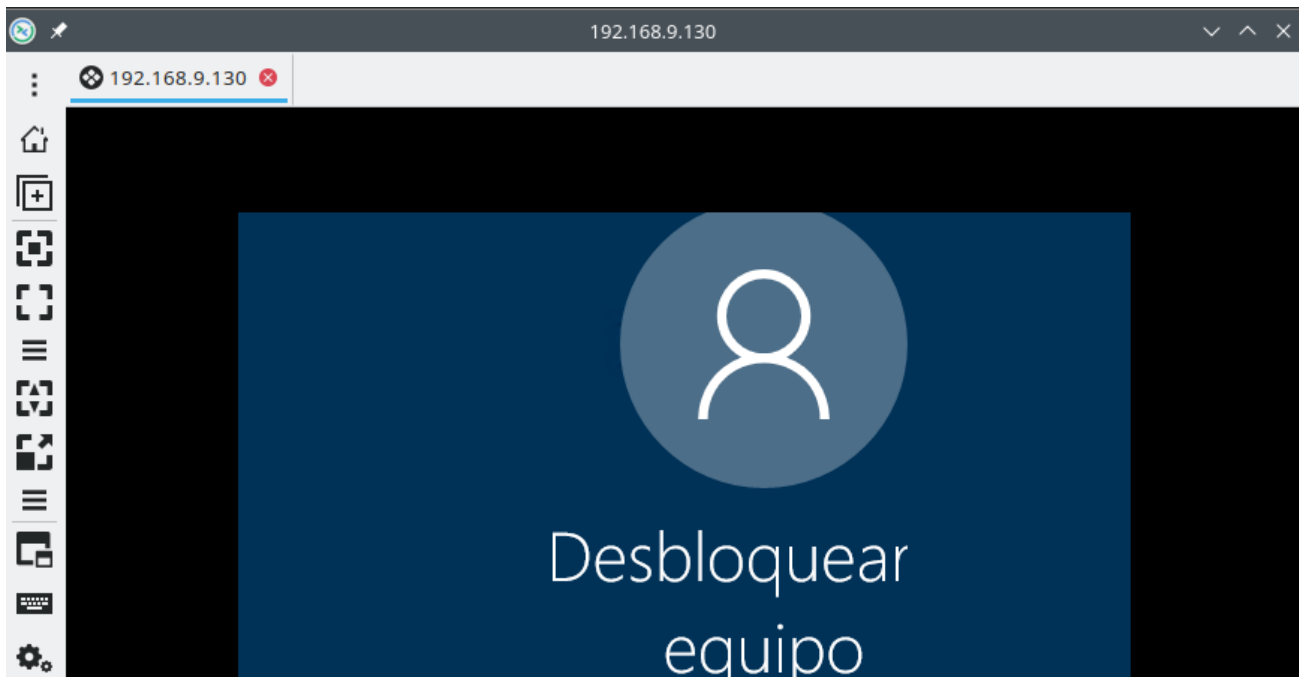
2.14.1. RDP

Para activar RDP en *windows* seguiremos la siguiente ruta:

1. Panel de control
2. Sistema de seguridad
3. Sistema
4. Configuración de acceso remoto



Ahora nos podremos conectar desde otro Windows o GNU/Linux



Desde otro S.O Windows podremos conectarnos también a través del cliente nativo o un programa como por ejemplo MobaXterm.

2.14.2. VNC

Desde linux o *windows* habrá que instalar un *software* como por ejemplo TightVNC o TigerVNC para tener el servidor y así poder conectarnos.

3. Servicio web, apache2

3.1. Instalación

Para instalar el servicio web `apache2` tendremos que instalar lo siguiente:

- `sudo apt-get install apache2`
- `sudo apt-get install libapache2-mod-php5`
- `sudo apt-get install apache2-utils`
- `sudo apt-get install mysql-server mysql-client mysql-common`
- `sudo apt-get install phpmyadmin`

3.2. Apuntes teóricos

- `/var/www/html`: ruta por defecto de apache donde se sirve el contenido web y los fichero `.html`
- `/etc/apache2/apache2.conf`: configuración general de `apache2`
- `/etc/apache2/sites-available`: fichero de configuración sobre los sitios de `apache`
- `/etc/apache2/sites-enabled`: sitios habilitados
- `/etc/apache2/mods-available`: módulos de `apache`

- `/etc/apache2/mods-enabled`: módulos de apache habilitados

3.2.1. Comandos útiles

- `apache2ctl -S`: muestra información importante de
 - puertos
 - IPs
 - sitios en los que se está haciendo algo
- `apache2 config-test`: comprueba la sintaxis de los ficheros

3.3. Práctica

3.3.1. Cambiar página por defecto

Podemos modificar la página por defecto de apache en el fichero `/var/www/html/index.html`



Lo que se puede apreciar arriba es el por defecto.

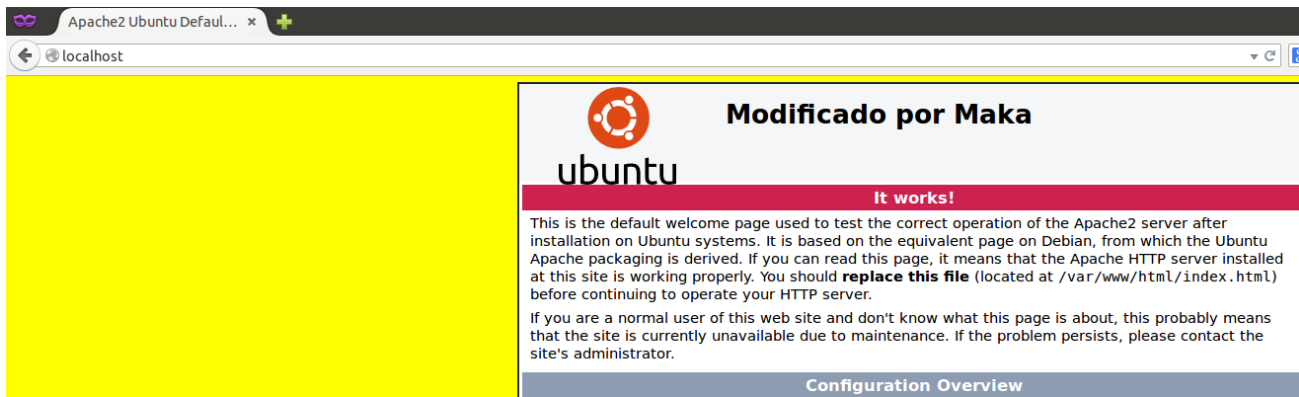
Modificaré el fichero `.html` para que se vea con otro título y fondo

```
body, html {
    padding: 3px 3px 3px 3px;

    background-color: yellow;

    font-family: Verdana, sans-serif;
    font-size: 11pt;
    text-align: center;
}
```

```
<body>
  <div class="main_page">
    <div class="page_header floating_element">
      
      <span class="floating_element">
        Modificado por Maka
      </span>
    </div>
  </div>
</body>
```



3.3.2. *option indexes*

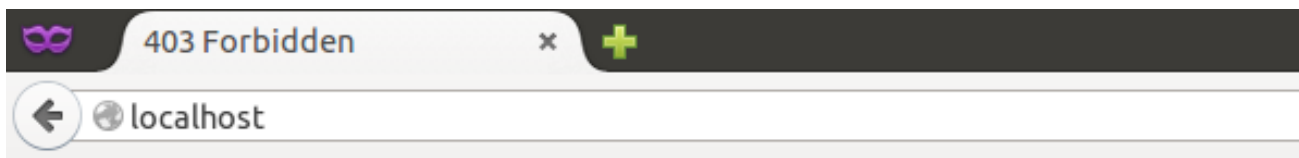
Podemos quitar la opción para ver el contenido de la siguiente forma:

1. Ir al fichero `/etc/apache2/apache2.conf`
2. Donde pone `<Directory /var/www>`
 - quitar el `options indexes FollowSymLinks`

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

Ahora también tenemos que cambiar el nombre del fichero `index.html` porque si no seguirá cargándolo por defecto. Luego recargaremos `apache2`

```
root@ubuntu101:/var/www/html# mv index.html index-cambiado.html
root@ubuntu101:/var/www/html# ls -la
total 28
drwxr-xr-x 3 root root 4096 feb 19 13:12 .
drwxr-xr-x 3 root root 4096 nov 7 08:58 ..
-rw-r--r-- 1 root root 11494 feb 19 13:08 index-cambiado.html
-rw-r--r-- 1 root root 21 feb 11 09:29 prueba.php
drwxr-xr-x 2 root www-data 4096 nov 7 09:08 webmail
root@ubuntu101:/var/www/html# service apache2 reload
* Reloading web server apache2
```



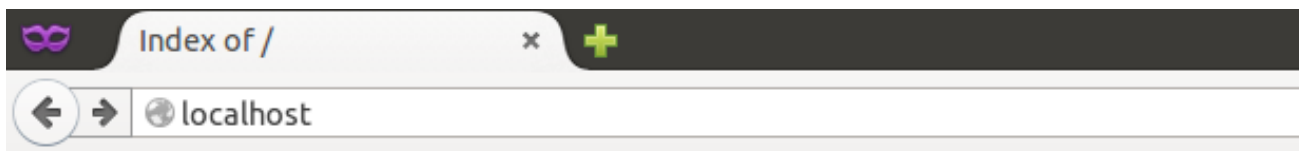
Forbidden

You don't have permission to access / on this server.




Apache/2.4.7 (Ubuntu) Server at localhost Port 80

Si queremos volver a habilitar esto, podemos hacerlo directamente en cada sitio de forma independiente. Añadiéndolo en su sitio de configuración como por ejemplo el 000-default.conf

Ahora volvemos a dejar todo como estaba. Como tenemos el fichero renombrado saldrá así:



Index of /

	<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
	index-cambiado.html	2025-02-19 13:08	11K	
	prueba.php	2025-02-11 09:29	21	
	webmail/	2024-11-07 09:08	-	

Apache/2.4.7 (Ubuntu) Server at localhost Port 80

3.3.3. Crear un sitio personalizado

Primero tendremos que crear tanto un fichero .conf como uno .html

El fichero /etc/apache2/sites-available/sitioIsmael.conf va a ser nuestro nuevo sitio:

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
```

```
# However, you must set it for any further virtual host explicitly.
ServerName www.sitioismael.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/sitioismael

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

<Directory /var/www/sitioismael>
    Options Indexes FollowSymLinks
    AllowOverride none
    Require all granted
</Directory>
```

Luego tendremos que habilitar el sitio mediante la instrucción `a2ensite` y seguidamente crearemos el fichero `.html`

- `sudo a2ensite sitioismael.conf`
- `mkdir /var/www/sitioismael`
- `emacs -q /var/www/sitioismael/index.html`

```
<h1>Bienvenido a www.sitioismael.com</h1>
```

Tendremos que añadir esto al fichero `/etc/hosts`

```
root@ubuntu101:/var/www/sitioismael# cat /etc/hosts
127.0.0.1        localhost
127.0.1.1        ubuntu1.myguest.virtualbox.org      ubuntu101
172.26.0.101     ubuntu101
192.168.9.156    www.sitioismael.com

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0  ip6-localnet
ff00::0  ip6-mcastprefix
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
```

Reiniciaremos el servicio web mediante la instrucción `service apache2 reload`



3.3.4. Contraseñas y restricción de acceso

Primero lo haremos mediante *passwd-basic*, para ello tendremos que acceder al fichero `/etc/apache2/sites-enabled/sitio`

Le pediré la autenticación para la carpeta interna que tenemos que crear después

```
<Directory /var/www/sitioismael/interna>
  AuthType basic
  AuthName "Acceso restringido a sitio interno"
  AuthBasicProvider file
  AuthUserFile /etc/apache2/passwd-basic
  Require user pedro
</Directory>
```

Ahora tendremos que crear el usuario `pedro` de una manera un poco más especial.

Aviso passwd-basic

Da igual que no exista el fichero, se crea con la instrucción `htpasswd`

```
root@ubuntu101:/var/www/sitioismael# htpasswd -c /etc/apache2/passwd-basic pedro
New password:
Re-type new password:
Adding password for user pedro
```

Reiniciamos `apache2` mediante la instrucción `service apache2 reload`



3.3.5. Páginas por puertos e IPs

Primero lo haremos mediante puertos.

Vamos a crear 2 directorios, uno para el puerto 8080, y otro para el puerto 8081

```
root@ubuntu101:/var/www# mkdir puerto8080
root@ubuntu101:/var/www# mkdir puerto8081
```

Ahora crearemos sus sites-availables

```
root@ubuntu101:/etc/apache2/sites-available# cp 000-default.conf puerto8080.conf
root@ubuntu101:/etc/apache2/sites-available# cp 000-default.conf puerto8081.conf
```

Ahora en cada uno tendremos que modificar el VirtualHost al puerto correspondiente de cada uno

```
<VirtualHost *:8080>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName www.puertos.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/puerto8080

    <Directory /var/www/puerto8080>
        Options Indexes FollowSymLinks
        AllowOverride none
        Require all granted
    </Directory>
```

```
<VirtualHost *:8081>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName www.puertos.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/puerto8081

    <Directory /var/www/puerto8081>
        Options Indexes FollowSymLinks
        AllowOverride none
        Require all granted
```

```
</Directory>
```

Luego tendremos que modificar el fichero `ports.conf` de la siguiente manera

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

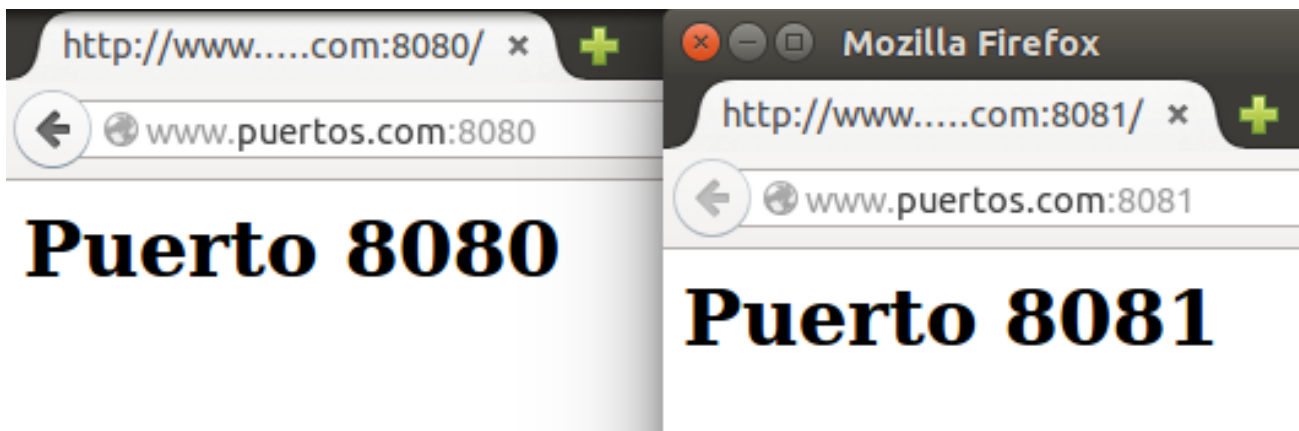
Listen 80
Listen 8080
Listen 8081
```

Luego habilitaremos los sitios mediante la instrucción `a2ensite`

Luego modificaremos el fichero `/etc/hosts` para que quede de la siguiente manera

```
root@ubuntu101:/etc/apache2/sites-available# cat /etc/hosts
127.0.0.1          localhost
127.0.1.1          ubuntu1.myguest.virtualbox.org      ubuntu101
172.26.0.101       ubuntu101
192.168.9.156      www.sitioismael.com
192.168.9.156      www.puertos.com
```

Por último, usando la instrucción `service apache2 reload` reiniciaremos el servicio web apache



Ahora lo haremos mediante la IP.

El proceso es el mismo que por puerto menos en un pequeño aspecto, en el `<VirtualHost>` en vez de tener un `*`, tendremos que tener la dirección IP y añadir está misma al fichero `/etc/hosts`

Primero añadiremos 2 tarjetas alías de la siguiente manera

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto eth0
allow-hotplug eth0
iface eth0 inet dhcp
```

```

auto eth0:1
allow-hotplug eth0:1
iface eth0:1 inet static
    address 10.0.1.10
    netmask 255.255.255.0

auto eth0:2
allow-hotplug eth0:2
iface eth0:2 inet static
    address 172.26.1.10
    netmask 255.255.0.0

```

Luego de aplicar los cambios y verlos reflejados a través de la instrucción `ip` a ya podremos modificar los fichero `/etc/apache2/sites-available/puerto8080.conf`

```

<VirtualHost 10.0.1.10:8080>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName www.puertos.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/puerto8080

    <Directory /var/www/puerto8080>
        Options Indexes FollowSymLinks
        AllowOverride none
        Require all granted
    </Directory>

```

```

<VirtualHost 172.26.1.10:8081>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName www.puertos.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/puerto8081

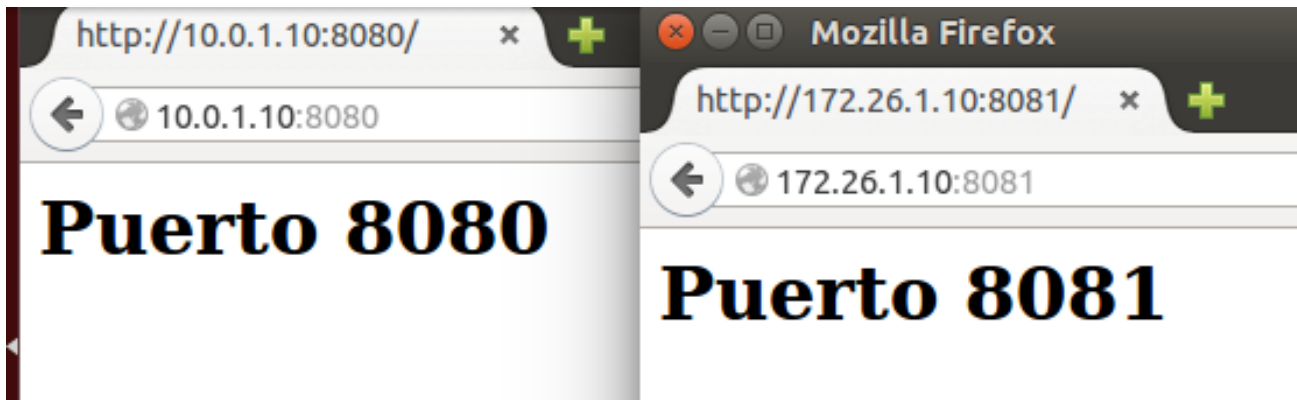
    <Directory /var/www/puerto8081>
        Options Indexes FollowSymLinks

```

```

        AllowOverride none
        Require all granted
    </Directory>

```



3.3.6. Sitio virtual, directorios *userdir*

Para esto comenzaremos como siempre, creando un fichero en `sites-available`, en este caso se llamará `arbol.conf`

```

<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName www.arbol.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/arbol

    <Directory /var/www/puerto8080>
        Options Indexes FollowSymLinks
        AllowOverride none
        Require all granted
    </Directory>

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    UserDir public_html

```

Ahora tenemos que habilitar tanto el sitio como el módulo `userdir` mediante las siguientes instrucciones:

- `a2ensite arbol.conf`

- a2enmod userdir

```
<IfModule mod_userdir.c>
    UserDir public_html
    UserDir disabled root

    <Directory /home/*/public_html>
        AllowOverride FileInfo AuthConfig Limit Indexes
        Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
        <Limit GET POST OPTIONS>
            Require all granted
        </Limit>
        <LimitExcept GET POST OPTIONS>
            Require all denied
        </LimitExcept>
    </Directory>
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Lo siguiente sería crear en la carpeta de la \$HOME una carpeta public-html con un fichero .html de la siguiente manera:

- mkdir /home/maka/public-html
- echo «h1>Mi PAGINA DE MAK</h1>» /home/maka/public-html/index.html

Luego de esto solo faltaría modificar el fichero /etc/hosts de la siguiente manera

```
127.0.0.1      localhost
127.0.1.1      ubuntu1.myguest.virtualbox.org  ubuntu101
172.26.0.101   ubuntu101
192.168.9.156  www.sitioismael.com
192.168.9.156  www.puertos.com
192.168.9.156  www.arbol.com
```

3.3.7. Securización

Simplemente tendremos que habilitar el sitio /etc/apache2/sites-available/default-ssl.conf y el módulo ssl con las siguientes instrucciones:

- a2ensite default-ssl.conf
- a2enmod ssl
- sudo service apache2 reload