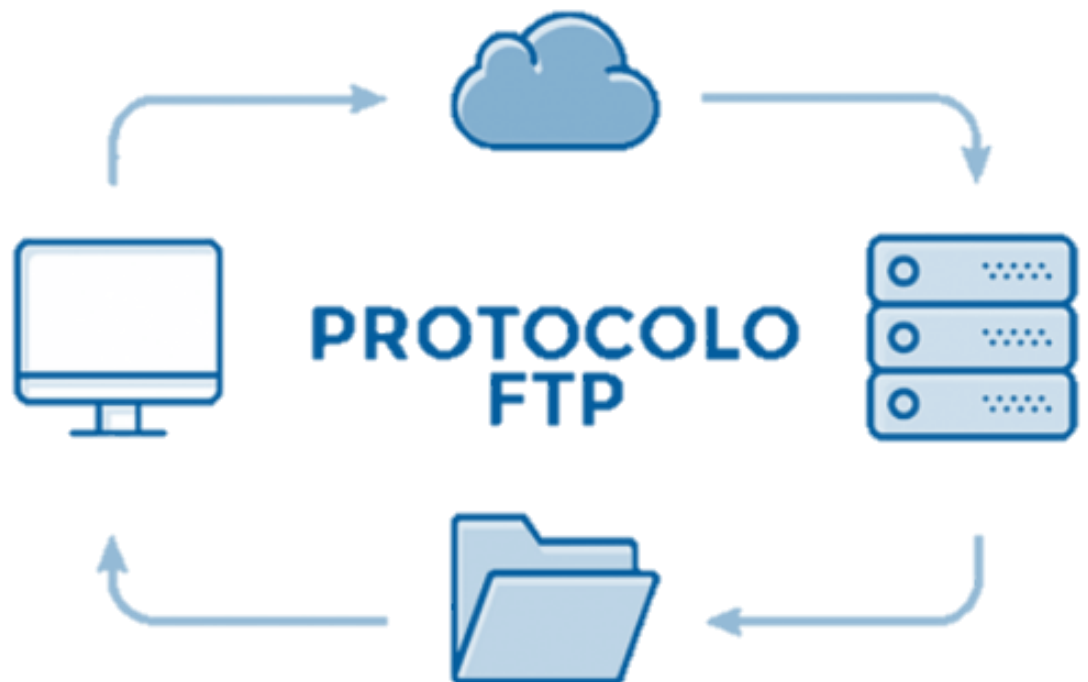

MANUAL LAMP Y FTP

Sara del Pino Cabrera Sánchez

2º DAW



ÍNDICE

Cambiar nombre de Usuario	3
Instalación Servidor LAMP	3
Apache.....	3
MySQL.....	5
PHP.....	5
Web por defecto.....	6
Configuración MySQL.....	7
Instalación VPS.....	9
Instalación PHPMyadmin.....	13
Instalación HTTPS	14
Certificado SSL.....	14
Configuración.....	15
Instalación FTP	17
Logs.....	18
Modificar mensaje.....	19
Carpetas por defecto.....	19
Creación de grupos y usuarios.....	20
Restringir usuarios.....	22
FTP privado y anónimo.....	23

MANUAL

Cambiar nombre de Usuario

Para cambiar el nombre de usuario que se ve desde nuestra terminal, tendremos que cambiar los ficheros *hosts* y *hostname* que se encuentran en el directorio */etc*. En ellos sobrescribiremos el antiguo nombre para poner el nuevo:

```
1  # Fichero hostname
2  sara_cabrera
3
4  # Fichero hosts
5  127.0.1.1    sara_cabrera
```



Debemos tener en cuenta reiniciar nuestra máquina tras aplicar estos cambios.

Instalación Servidor LAMP

Apache

Para este tutorial es importante saber que haremos uso de permisos como administrador, así que es interesante hacer primero un:

```
1  $ sudo su
```

Primeramente, abriremos nuestra terminal para actualizar nuestro sistema y, tras eso, instalaremos Apache2.

```
1  # Actualizamos el sistema
2  $ apt update
3
4  # Instalamos Apache
5  $ apt install apache2
```

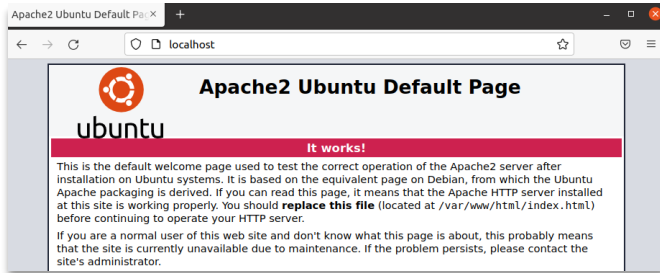


Es importante saber que *update* actualiza el software dentro de la versión que tenemos definida, y *upgrade* actualiza el software a la versión más reciente.

Una vez instalado, podremos comprobar el estado de nuestro servidor Apache haciendo uso de:

```
1 $ systemctl status apache2
```

O también podemos acceder desde nuestro navegador a localhost o 127.0.0.1



```
root@saracabrera:/home/sara# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2022-11-06 13:17:47; 1min 1s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 2649 (apache2)
      Tasks: 55 (limit: 11011)
     Memory: 4.8M
    CGroup: /system.slice/apache2.service
            └─2649 /usr/sbin/apache2 -k start
              └─3742 /usr/sbin/apache2 -k start
                └─3743 /usr/sbin/apache2 -k start

nov 06 13:17:37 saracabrera systemd[1]: Starting The Apache HTTP Server:
nov 06 13:17:47 saracabrera apachectl[2648]: AH00558: apache2: Syntax error in
nov 06 13:17:47 saracabrera systemd[1]: Started The Apache HTTP Server:
lines 1-15/15 (END)
```

Ahora tendremos que activar el cortafuegos (firewall) de nuestra máquina:

```
1 # Activamos el FireWall
2 $ ufw enable
3
4 # Muestra aquellas Apps con el FireWall activado
5 $ ufw app list
6
7 # Evitamos que nos proteja de las peticiones al servidor apache
8 $ ufw allow in "Apache"
```



Otros comandos sobre FireWall que pueden sernos útiles:

```
1 # Podemos desactivar el FireWall con:
2 $ ufw disable
3
4 # Para saber el estado de nuestro FireWall podemos usar:
5 $ ufw status
```



En caso de que queramos acceder a nuestra máquina desde fuera necesitaremos nuestra ip pública. Esta podemos conseguirla de la siguiente manera:

```
1 # Vemos nuestra configuración ip
2 $ ip add
3
4 # Para mostrar nuestra ip pública:
5 $ apt install curl
6 $ curl http://icanhazip.com
```

MySQL

A continuación, pasaremos a instalar mysql-server (sin seguridad):

```
1 # Instalamos Mysql
2 $ apt install mysql-server
3
4 # Comprobamos que funcione correctamente
5 $ mysql
```

```
root@saracabrera:/home/sara# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.31-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or
its affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql>
```

PHP

Ahora toca hacer la instalación de PHP en nuestro servidor, para ello instalaremos los módulos correspondientes:

```
1 # Instalamos los módulos necesarios
2 $ apt install php libapache2-mod-php php-mysql
```

Para comprobar que se ha instalado correctamente podremos preguntar por la versión de php que tenemos en nuestro servidor:

```
1 # Comprobamos la versión de php
2 $ php -v
```

```
root@saracabrera:/home/sara# php -v
PHP 7.4.3 (cli) (built: Aug 17 2022 13:29:56) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technolog
ies
```

Web por defecto

Para poder cambiar la página web que por defecto nos abre nuestro servidor tendremos que:

```
1 # Ir al directorio /var/www/html
2 $ cd /var/www/html
3
4 # Hacer una copia de seguridad de index.html
5 $ cp index.html index_copia.html
6
7 # Modificamos el index.html a nuestro gusto.
```



También podemos crear un index.php y que éste se abra por defecto en nuestro servidor, para ello tendremos que cambiar la configuración de Apache de la siguiente forma:

```
1 # Nos movemos al directorio /etc/apache2/mods-enable
2 $ cd /etc/apache2/mods-enable
3
4 # Editamos el fichero dir.conf
5 $ gedit dir.conf
6
7 # Añadimos el index.php como primera opción
8 # Reseteamos el Servicio de Apache
9 $ systemctl reload apache2
```



Es interesante conocer las diferentes formas de abrir un fichero en Linux. Tenemos la opción de *nano*, esta nos abrirá el fichero desde la terminal. Por otro lado, tenemos la opción de *gedit*, esta abrirá nuestro fichero en un archivo txt aparte, siendo mucho más cómodo de modificar y leer.

A continuación, añadiremos contenido de prueba al index.php y podremos visualizarlo en la web de nuestro servidor.

```
1 <?php phpinfo(); ?>
```



Configuración MySQL

Para configurar MySQL seguiremos los siguientes pasos (dentro de mysql):

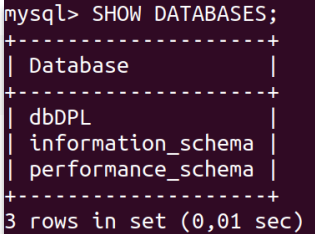
```
1 CREATE DATABASE dbDPL;
2
3 # Creamos el usuario y le introducimos la contraseña
4 CREATE USER 'SaraC'@'%' IDENTIFIED WITH MYSQL_NATIVE_PASSWORD BY '1234';
5
6 # Damos permisos al usuario creado sobre la base de datos
7 GRANT ALL ON dbDPL.* TO 'SaraC'@'%';
```

Una vez que hemos creado a nuestro usuario y le hemos dado los permisos correspondientes, iniciaremos mysql como dicho usuario:

```
1 $ mysql -u SaraC -p
```

Una vez insertemos la contraseña correctamente nos dejará iniciar mysql. En caso de que queramos comprobar que efectivamente hemos creado bien la base de datos podemos mostrarla de la siguiente forma:

```
1 SHOW DATABASES;
```



```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| dbDPL    |
| information_schema |
| performance_schema |
+-----+
3 rows in set (0,01 sec)
```

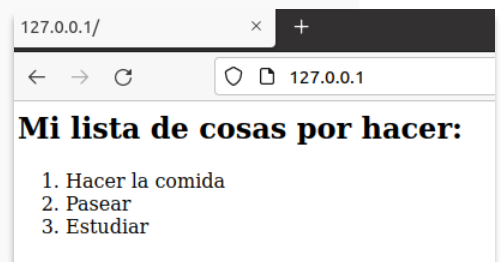
A continuación, crearemos una tabla y le insertaremos datos:

```
1  USE dbDPL;
2
3  CREATE TABLE TodoList(
4      item_id INT AUTO_INCREMENT,
5      contenido VARCHAR(255),
6      PRIMARY KEY (item_id)
7  );
8
9  INSERT INTO TodoList (contenido) VALUES
10     ("Hacer la comida"),
11     ("Pasear"),
12     ("Estudiar");
13
14  SELECT * FROM TodoList;
```

```
mysql> SELECT * FROM TodoList;
+-----+-----+
| item_id | contenido |
+-----+-----+
|      1 | Hacer la comida |
|      2 | Pasear |
|      3 | Estudiar |
+-----+-----+
3 rows in set (0,00 sec)
```

Y cambiaremos el index.php a lo siguiente:

```
1  <?php
2      $user="SaraC";
3      $password="1234";
4      $database="dbDPL";
5      $table="TodoList";
6
7      try {
8          $db = new PDO("mysql:host = localhost; dbname=$database", $user, $password);
9          echo "<h2> Mi lista de cosas por hacer: </h2><ol>";
10         foreach ($db->query("SELECT contenido FROM $table") as $row) {
11             echo "<li>" . $row["contenido"] . "</li>";
12         } echo "</ol>";
13     } catch (PDOException $e) {
14         print "Error!" . $e->getMessage() . "<br/>";
15         die();
16     }
17     ?>
```





Podemos ver los permisos de un usuario en concreto haciendo uso de:

```
1 # Para ver los permisos de un usuario.  
2 SHOW GRANTS FOR "SaraC"@"%";
```

Instalación VPS

Debemos dirigirnos al directorio /var/www y crear una carpeta para nuestro proyecto:

```
1 # Creamos la carpeta y entramos en ella  
2 $ mkdir proyectoDPL.son  
3 $ cd proyectoDPL.son  
4  
5 # Creamos un fichero html  
6 $ gedit index.html
```

La estructura de nuestro html quedará de la siguiente forma:

```
1 <!DOCTYPE html>  
2 <html lang="es">  
3   <head>  
4     <meta charset="UTF-8">  
5     <title>Proyecto DPL</title>  
6   </head>  
7   <body>  
8     <h1>Proyecto DPL</h1>  
9   </body>  
10 </html>
```

Ahora iremos al directorio /etc/apache2/sites-available y haremos una copia del fichero 000-default.conf cambiándole el nombre:

```
1 $ cd /etc/apache2/sites-available  
2 $ cp 000-default.conf proyectoDPL.son.conf
```

En el nuevo fichero tendremos que añadir lo siguiente:

```
1 <VirtualHost *:80>
2     ServerName proyectoDPL.son
3     ServerAlias www.proyectoDPL.son
4     ServerAdmin webmaster@dominioDPL.com
5     DocumentRoot /var/www/proyectoDPL.son
6
7     <Directory /var/www/proyectoDPL.son>
8         Options -Indexes +FollowSymLinks
9         AllowOverride All
10    </Directory>
11
12    ErrorLog ${APACHE_LOG_DIR}/proyectoDPL.son-error.log
13    CustomLog ${APACHE_LOG_DIR}/proyectoDPL-access.log combined
14 </VirtualHost>
```



Es interesante saber como ver los puertos abiertos en nuestro servidor:

```
1 # Opción 1
2 $ netstat -tplugin
3 # En caso de querer ver el estado de un puerto concreto
4 $ netstat -tplugin | grep:puerto
5
6 # Opción 2
7 $ lsof -i
8 # En caso de querer ver el estado de un puerto concreto
9 $ lsof -i | grep:puerto
```



Para indicarle a Apache cual es el puerto por el que debe escuchar, iremos a `/etc/apache2/ports.conf` y añadiremos la línea `Listen :puerto`.

```
1 | # Iremos al siguiente directorio
2 | $ /etc/apache2
3 |
4 | # Abriremos el fichero
5 | $ gedit ports.conf
6 |
7 | # Añadiremos la siguiente línea
8 | Listen :puerto
```

Una vez tengamos todo esto, debemos habilitar nuestro sitio web y recargar apache para que se apliquen los cambios:

```
1 | # Habilitamos el sitio
2 | $ a2ensite proyectoDPL.son.conf
3 |
4 | # Recargamos Apache
5 | $ systemctl reload apache2
```



Es bueno conocer también la directiva para deshabilitar un sitio web:

```
1 | # Deshabilitamos el sitio
2 | $ a2dissite proyectoDPL.son.conf
```

Ahora iremos al directorio `/etc/apache2` y abrimos el fichero de `apache2.conf` para añadir el `ServerName` de nuestro proyecto.

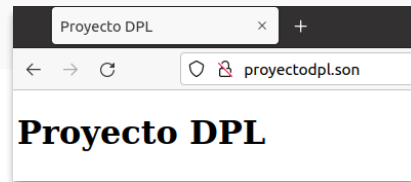
```
1 | # Al final del fichero
2 | ServerName 127.0.0.1
```

Reseteamos el servidor y comprobamos que todo esté correctamente:

```
1 # Reseteamos el servidor
2 $ systemctl restart apache2
3
4 # Comprobamos que la configuración esté bien
5 $ apachectl configtest
```

Por último, debemos direccionar nuestra ip local al nuevo dominio que hemos creado, para ello modificamos el fichero hosts dentro del directorio /etc:

```
1 # Al principio del fichero
2 127.0.0.1    proyectoDPL.son
```



Ahora ya podremos acceder a <http://proyectoDPL.son> y comprobar que funciona correctamente.



En caso de tener algún fallo, podremos verlos desde el fichero `proyectoDPL.son-error.log` dentro del directorio `/var/log/apache2`:

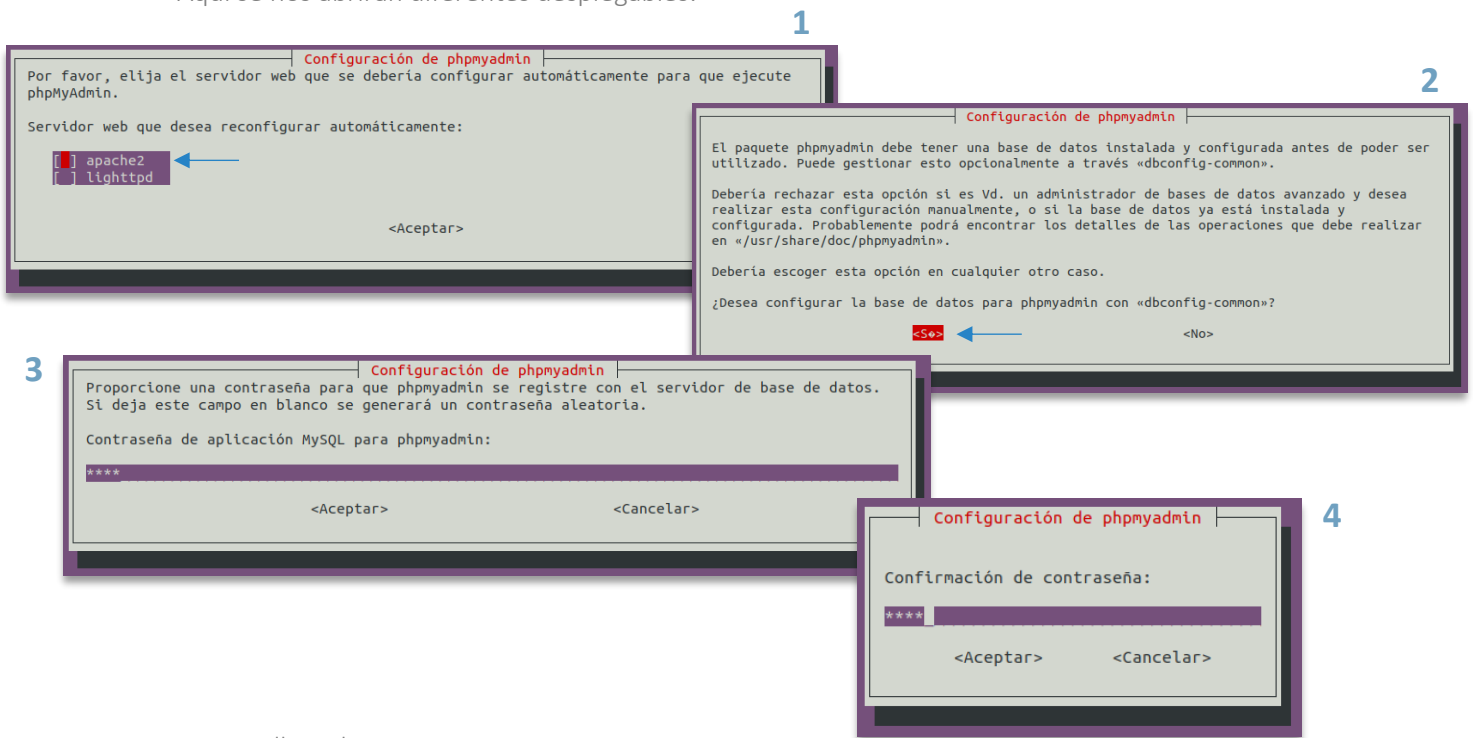
```
1 # Para ver las últimas 10 líneas del fichero
2 $ tail -10 proyectoDPL.son-error.log
3
4 # Para ver las primeras 10 líneas del fichero
5 $ head -10 proyectoDPL.son-error.log
```

Instalación PHPMyadmin

Para la instalación de PHPMyadmin tendremos que seguir los siguientes pasos:

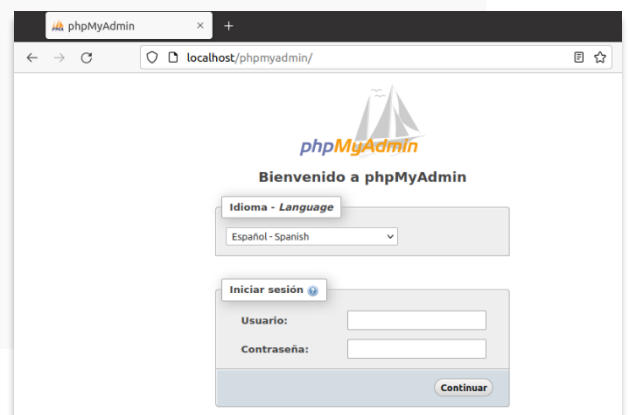
```
1 | $ apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```

Aquí se nos abrirán diferentes desplegables:



Y tras rellenarlo continuamos:

```
1 | # Vamos a la carpeta de configuración del proyecto y abrimos apache2.conf
2 | $ cd /etc/apache2
3 | $ gedit apache2.conf
4 |
5 | # Añadimos el directorio de phpmyadmin al final del fichero apache2.conf
6 | Include /etc/phpmyadmin/apache.conf
7 |
8 | # Habilitamos PHP y PHPMyAdmin
9 | $ phpenmod mbstring
10 |
11 | # Reseteamos apache
12 | $ systemctl restart apache2
```



Instalación HTTPS

Para instalar HTTPS en nuestras páginas primero tendremos que desactivar el cortafuegos:

```
1 | $ ufw disable
```

A continuación, añadiremos el módulo SSL:

```
1 | # Habilitamos el módulo ssl
2 | $ a2enmod ssl
3 |
4 | # Reseteamos apache
5 | $ systemctl restart apache2
```

Certificado SSL

Ahora generaremos el certificado SSL firmado por nosotros:

```
1 | $ openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
   /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
2 |
3 | # Rellenamos los campos que nos saldrán a continuación:
4 | Country Name (2 letter code) [XX]: ES
5 | State or Province Name (full name) []: LPA
6 | Locality Name (eg, city) [Default City]: LPA
7 | Organization Name (eg, company) [Default Company Ltd]: sarcabsan.sl
8 | Organizational Unit Name (eg, section) []: section
9 | Common Name (eg, your name or your server's hostname) []: www.proyecto.son
10 | Email Address []: correo
```

Configuración

Cambiamos la configuración en el archivo proyectoDPL.son.conf dentro del directorio /etc/apache2/sites-available:

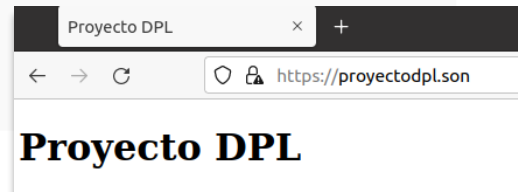
```
1  # Abrimos el fichero
2  $ gedit proyectoDPL.son.conf
3
4  # Cambiamos el puerto por el 443 (Puerto por defecto de HTTPS)
5  <VirtualHost *:443>
6      ...
7      # En el fichero escribimos esto al final del todo
8      SSLEngine on
9      SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
10     SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
11 </VirtualHost>
```

Deshabilitamos el proyecto y reseteamos apache:

```
1  # Deshabilitamos el proyecto
2  $ a2dissite proyectoDPL.son.conf
3
4  # Reseteamos apache
5  $ systemctl restart apache2
```

Volvemos a habilitar el proyecto, comprobamos que la configuración esté correcta y recargamos apache:

```
1 # Habilitamos el proyecto
2 $ a2ensite proyectoDPL.son.conf
3
4 # Comprobamos que la configuración esté bien hecha
5 $ apache2ctl configtest
6 # En caso de que todo esté correctamente nos saldrá un mensaje de "Syntax OK"
7
8 # Recargamos apache
9 $ systemctl reload apache2
```



En caso de que esto no nos funcione podemos probar a abrir los puertos 443 y 80:

```
1 # Abrimos los puertos 443 y 80
2 $ ufw allow "Apache Full"
```



Por último, es interesante redireccionar el http al https, para ello iremos al fichero de configuración proyectoDPL.son.conf en el directorio /etc/apache2/sites-available:

```
1 # Añadimos un virtualhost para el puerto 80 fuera del que ya tenemos
2 <VirtualHost *:80>
3     ServerName proyectoDPL.son
4     Redirect / https://proyectoDPL.son
5 </VirtualHost>
6
7 # Recargamos apache
8 $ systemctl reload apache2
```


Instalación FTP



Primeramente, es interesante conocer los principales comandos de FTP:

- 1 `help o ?` -> enumera todos los comandos de FTP disponibles.
- 2 `cd` -> cambia el directorio en la máquina remota.
- 3 `lcd` -> cambiar el directorio en la máquina local.
- 4 `ls` -> muestra los nombres de los archivos y directorios en el directorio remoto actual.
- 5 `mkdir` -> crea un nuevo directorio dentro del directorio remoto.
- 6 `pwd` -> imprime el directorio de trabajo actual en la máquina remota.
- 7 `delete` -> elimina un archivo en el directorio remoto actual.
- 8 `rmdir` -> elimina un directorio en el directorio remoto actual.
- 9 `get` -> copia un archivo del servidor remoto a la máquina local.
- 10 `mget` -> permite copiar múltiples archivos del servidor remoto a la máquina local.
- 11 `put` -> copia un archivo de la máquina local a la máquina remota.
- 12 `mput` -> copia un archivo de la máquina remota a la máquina local.

Para instalar el FTP haremos uso de las siguientes directivas:

- 1 `# Actualizamos el software del equipo`
- 2 `$ apt update`
- 3
- 4 `# Instalamos el módulo`
- 5 `$ apt install proftpd`



Tras la instalación se nos habrán generado una serie de ficheros de configuración, los más importantes son:

- `proftpd.conf`: archivo de configuración de proftpd, se encuentra en el directorio `/etc/proftpd/`
- `ftputils`: fichero que contiene una lista de usuarios que no tienen permiso de acceso por FTP. Por razones de seguridad al menos los siguientes usuarios deberían estar listados en este fichero: root, bin, uucp, news. Se encuentra en el directorio `/etc`

Para conectarnos al FTP lo haremos de la siguiente forma:

```
1 # ftp "dirección_ip"
2 $ ftp localhost
```

Al conectarnos nos pedirá un usuario y una contraseña que, por defecto, serán las de nuestro equipo.

```
root@saracabrera:/etc/apache2/sites-available# ftp localhost
Connected to localhost.
220 ProFTPD Server (Debian) [::ffff:127.0.0.1]
Name (localhost:sara):
331 Contraseña necesaria para sara
Password:
230 Usuario sara conectado
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

 Algunas de las directivas más básicas a la hora de configurar nuestro proftpd.conf son:

- **ServerName**: nombre de nuestro servidor.
- **DeferWelcome**: con esto activado, no mostraremos información sobre nuestro equipo hasta que el usuario haga login.
- **ShowSymlinks**: habilita/deshabilita los enlaces simbólicos.
- **TimeoutIdle**: configura la cantidad máxima de segundos que proftpd permitirá a los clientes permanecer conectados sin recibir ningún dato.
- **Port**: puerto de nuestro servidor (21 por defecto).
- **TransferLog**: especifica la ruta al registro de transferencia.
- **SystemLog**: ruta del fichero en donde guardamos los logs.

Logs

Para mostrar el contenido de los logs de proftpd iremos a la ruta /var/log/proftpd y ejecutaremos el fichero proftpf.log:

```
1 # Vamos al siguiente directorio:
2 $ cd /var/log/proftpd
3
4 # Muestra los logs
5 $ cat proftpd.log
```

Modificar mensaje

Para modificar el mensaje de bienvenida y de error iremos a la ruta `/etc/proftpd` y editaremos el fichero `proftpd.conf`:

```
1 # Vamos al siguiente directorio:
2 $ cd /etc/proftpd
3
4 # Abrimos el fichero proftpd
5 $ gedit proftpd.conf
6
7 # Insertamos las siguientes líneas al final del fichero:
8 AccessGrantMsg "Bienvenido a mi servidor"
9 AccessDenyMsg "ERROR al entrar al servidor"
10
11 # Reseteamos el FTP
12 $ service proftpd restart
```

```
root@saracabrera:/etc/proftpd# ftp localhost
Connected to localhost.
220 ProFTPD Server (Debian) [::ffff:127.0.0.1]
Name (localhost:sara):
331 Contraseña necesaria para sara
Password:
230 Bienvenido a mi servidor
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

```
root@saracabrera:/etc/proftpd# ftp localhost
Connected to localhost.
220 ProFTPD Server (Debian) [::ffff:127.0.0.1]
Name (localhost:sara):
331 Contraseña necesaria para sara
Password:
530 ERROR al entrar al servidor
Login failed.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Carpetas por defecto

Existe una directiva llamada `DefaultRoot` con la que podemos definir a que carpeta entrarán, por defecto, nuestros usuarios al FTP (va colocada al final del fichero `proftpd.conf` del directorio `/etc/proftpd`):

```
1 # Sintaxis de la directiva:
2 DefaultRoot [directorio] [grupo de usuarios que tendrán este directorio por defecto]
   [grupo de usuarios a los que no se les aplicará el directorio]
3
4 # Por defecto definimos home
5 DefaultRoot ~
6
7 # Por defecto definimos /var/ftpXX
8 DefaultRoot /var/ftpXX
```

Creación de grupos y usuarios

Para la creación de usuarios usaremos:

```
1 # Creamos al usuario user1dpl
2 $ adduser user1dpl
```



Podemos eliminar un usuario con *userdel* seguido del nombre del usuario a eliminar.



Debemos tener en cuenta las diferencias entre *adduser* y *useradd*:

- **useradd**: crea un nuevo usuario, pero no le crea su respectiva carpeta en */home*. En caso de que queramos crearla tendremos que indicarlo con:

```
1 $ useradd -m usernuevo
```

- **adduser**: crea un nuevo usuario y su respectiva carpeta en */home*. La desventaja es que este comando no es compatible con todas las distribuciones de Linux.

Podemos comprobar que un usuario ha sido creado correctamente haciendo uso de:

```
1 # Muestra todos los usuarios de nuestro equipo
2 $ cat /etc/passwd
```

También podemos conocer la configuración de un usuario en concreto con:

```
1 $ id user1dpl
```

```
root@saracabrera:/var# id user1dpl
uid=1001(user1dpl) gid=1001(user1dpl) grupos=1001(user1dpl)
```

Esto nos indicará, entre otras cosas, a que grupos pertenece el usuario.

Para crear un grupo y asignárselo a un usuario seguiremos los siguientes pasos:

```
1 # Creamos el grupo
2 $ addgroup grupo1
3
4 # Añadimos a un usuario al grupo
5 $ usermod -a -G grupo1 user1dpl
```




Podemos eliminar un grupo con *delgroup* seguido del nombre del grupo a eliminar.

```
root@saracabrera:/var# id user1dpl
uid=1001(user1dpl) gid=1001(user1dpl) grupos=1001(user1dpl),1002(grupo1)
```

También podemos cambiar los permisos y los grupos de diferentes carpetas, para ello podemos hacer uso de los siguientes comandos:

```
1 # Para ver los grupos y permisos sobre una serie de carpetas:
2 $ ls -l
3
4 # Para poner a un grupo como propietario de una carpeta:
5 $ chgrp grupo1 ftpXX
6
7 # Para darle todos los permisos al grupo sobre dicha carpeta:
8 $ chmod 775 ftpXX
```

 Para la hora de dar permisos sobre diferentes ficheros es muy importante conocer cómo se establecen estos:

r	w	x	Número	Permisos
0	0	0	0	---
0	0	1	1	--x
0	1	0	2	-w-
0	1	1	3	-wx
1	0	0	4	r--
1	0	1	5	r-x
1	1	0	6	rw-
1	1	1	7	rwX

Restringir usuarios

Para poder restringir a ciertos usuarios a una única carpeta, tendremos que ir al directorio `/etc/proftpd` y modificar el fichero `proftpd.conf` añadiendo al final:

```
1  DefaultRoot /var/ftpXX grupo1
2
3  # Reseteamos el FTP
4  $ service proftpd restart
```

Una vez hecho esto, podremos iniciar sesión con un usuario que pertenezca a dicho grupo y comprobaremos que se ha hecho todo correctamente.

Para permitir o denegar usuarios en nuestro FTP tendremos que modificar el fichero `proftpd.conf` de nuestro directorio `/etc/proftpd`:

```
1  # Añadimos al final de fichero:
2  <Limit LOGIN>
3      AllowUser user1dpl
4      DenyALL
5  </Limit>
```



Estos son los diferentes parámetros que podremos usar:

- `AllowUser [nombre del usuario]`: permitimos el acceso a un usuario en específico.
- `DenyUser [nombre del usuario]`: bloqueamos el acceso a un usuario en específico.
- `DenyAll`: bloqueamos a todos los usuarios (salvo los que permitamos con `AllowUser`).
- `AllowAll`: permitimos que los usuarios como “anonymous” se conecten al FTP.

FTP privado y anónimo

Para configurar nuestro FTP como privado y anónimo tendremos que descomentar las siguientes líneas del fichero proftpd.conf de nuestro directorio /etc/proftpd:

```
148
149 # <Anonymous ~ftp>
150 #   User ftp
151 #   Group nogroup
152 #   # We want clients to be able to login with "anonymous" as well as "ftp"
153 #   UserAlias anonymous ftp
154 #   # Cosmetic changes, all files belongs to ftp user
155 #   DirFakeUser on ftp
156 #   DirFakeGroup on ftp
157 #
158 #   RequireValidShell off
159 #
160 #   # Limit the maximum number of anonymous logins
161 #   MaxClients 10
162 #
163 #   # We want 'welcome.msg' displayed at login, and '.message' displayed
164 #   # in each newly chdir'd directory.
165 #   DisplayLogin welcome.msg
166 #   DisplayChdir .message
167 #
168 #   # Limit WRITE everywhere in the anonymous chroot
169 #   <Directory *>
170 #     <Limit WRITE>
171 #       DenyAll
172 #     </Limit>
173 #   </Directory>
174 #
175 #   # Uncomment this if you're brave.
176 #   # <Directory incoming>
177 #     #   # Umask 022 is a good standard umask to prevent new files and dirs
178 #     #   # (second parm) from being group and world writable.
179 #     #   Umask 022 022
180 #     #     <Limit READ WRITE>
181 #     #       DenyAll
182 #     #     </Limit>
183 #     #     <Limit STOR>
184 #     #       AllowAll
185 #     #     </Limit>
186 #   # </Directory>
187 #
188 # </Anonymous>
189
```

Y con esto, si entramos como usuario anónimo veremos que no nos deja hacer nada.