

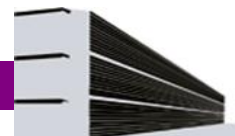
Documento: Servidor Intranet versión 1.0

Proyecto: Intranet Batoi

Modulo: CGS Desarrollo Aplicaciones Web

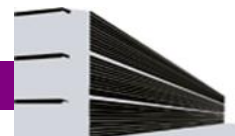
Profesores: Carlos Monllor, Rosa Aravid, Juan Segura
Cristina Tennes

Integrantes: Raúl Lara Rico
Carlos Huelmo Vaquero
José Luis Moltó Gimeno
José Vicente Martinez Mellado
Daniel Ferrándiz López



Índice

| | |
|--|----|
| Inicio..... | 3 |
| Primeros pasos..... | 3 |
| Instalación básica automatizada..... | 4 |
| Configuración..... | 6 |
| Configuración del sistema: toor con permisos de root..... | 6 |
| Carpetas para toor..... | 6 |
| Configuración avanzada..... | 7 |
| Apache..... | 7 |
| MySQL..... | 8 |
| Vsftpd (servidor ftp)..... | 10 |
| Actualizar: PhpMyAdmin(v.4.5.4) + PHP(5.5.30)..... | 11 |
| Configurar el recurso “/fotos”..... | 13 |
| Configurar acceso a las fotografías en “/srv/intranet/fotos” (Autenticación Básica)..... | 13 |
| Configurar acceso a las fotografías en “/srv/intranet/fotos” (Autenticación por MySQL i y ii)..... | 14 |
| Apache HTTPS/SSL..... | 17 |
| El fichero “intranet_mysql_ssl.conf”..... | 19 |
| Personalizar “ErrorDocument”..... | 20 |
| Importando el proyecto..... | 21 |
| Copia remota del proyecto..... | 21 |
| Copias de seguridad (backup's)..... | 22 |
| Uso de “cron”..... | 22 |
| Copiar backups de “/root” a “/home/toor/scripts/.backups_file”..... | 23 |
| Realizar backup de forma remota..... | 24 |
| Realizar backup de forma local..... | 25 |
| Comandos a utilizar para administrar nuestro servidor..... | 26 |
| Fuentes..... | 28 |



Inicio.

El centro nos proporciona un servidor “GNU/Linux” virtualizado por PROXMOX, un Debian 7.9 y se nos proporciona la IP (172.16.70.101) y acceso por ssh como usuario root.

Requisitos virtualizados:

- Disco duro: 5 GB.
- 1GB RAM.
- Arquitectura: 32 bits (i686).
- Tarjeta de red.
- Sin entorno gráfico, sólo por terminal (línea de comandos).

Primeros pasos.

Cambiaremos la contraseña de “root” que por defecto era “1234”. Para ello, remotamente accederemos vía “ssh” desde la máquina remota al debian virtualizado, para ello utilizaremos el siguiente comando:

```
$ ssh -X -Y root@172.16.70.101
```

Añadiremos un nuevo usuario “toor” con privilegios de “root”, por seguridad. Para ello escribiremos:

```
Archivo Editar Ver Buscar Terminal Ayuda  
root@dawl:~# useradd -g root -d /home/toor -m -s /bin/bash toor
```

Después proporcionaremos un password a nuestro nuevo usuario “toor” como “root”. Lo que hemos hecho ha sido crear un usuario nuevo, añadirlo al grupo de administradores, crear su carpeta “home” y su shell.

```
root@dawl:~# passwd toor  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully
```

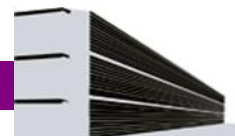
Ahora ya no dependeremos del usuario “root” para realizar conexiones ssh remotamente.

Otra parte es la actualización del fichero de repositorios de nuestro “Debian”, entendemos que sería aconsejable añadir más repositorios, por lo que tendremos dos archivos “sources.list.v[X]” que previamente habremos obtenido y que por ssh habremos enviado remotamente al recurso “/etc” utilizando el comando:

```
Archivo Editar Ver Buscar Terminal Ayuda  
vesprada@pcxx:~/Escritorio$ scp sources.list.v1 toor@172.16.70.101:/home/toor/  
toor@172.16.70.101's password:  
sources.list.v1 100% 557 0.5KB/s 00:00  
vesprada@pcxx:~/Escritorio$ scp sources.list.v2 toor@172.16.70.101:/home/toor/  
toor@172.16.70.101's password:  
sources.list.v2 100% 870 0.9KB/s 00:00
```

Después copiaremos los archivos a “/etc” y renombraremos uno de ellos (la v2 en nuestro caso, porque es más completo) y actualizaremos mediante el comando: `# apt-get update`.

NOTA: a partir de una máquina virtual Debian 7.8 actualizada al 8.3.0, hemos codigo su archivo “sources.list” y la hemos añadido a nuestro Debian para actualizar los paquetes. Por motivos ajenos finalmente tendremos una máquina Debian 7.X.



Instalación básica automatizada.

Hemos preparado un script para automatizar la instalación de paquetes básicos que necesitaremos para nuestro servidor y dar servicio en la “Intranet” del centro. El código del fichero “instalacion_basica.sh” se muestra a continuación:

```
1  #!/bin/bash
2  # cargar datos directamente en el propio servidor
3  clear
4  echo "*****"
5  echo "*   INSTALACION DE PAQUETES SERVIDOR DEBIAN   *"
6  echo "*****"
7  echo " "
8  # actualizar repositorios
9  apt-get update
10 # comprobar dependencias
11 aptitude update
12 # INSTALAR APACHE 2
13 apt-get install apache2 apache2-utils apache2-doc libapache2-mod-auth-mysql
14 echo "*****"
15 echo "Apache Instalado"
16 echo "*****"
17 # INSTALAR MYSQL
18 apt-get install mysql-server php5-mysql
19 echo "*****"
20 echo "MySQL Instalado"
21 echo "*****"
22 # INSTALAR PHP
23 apt-get install php5 php5-mcrypt php5-curl mcrypt
24 echo "*****"
25 echo "PHP Instalado"
26 echo "*****"
27 # INSTALAR phpMyAdmin
28 apt-get install phpmyadmin
29 echo "*****"
30 echo "PHPMYADMIN Instalado"
31 echo "*****"
32 # INSTALAR OPENSSH SERVER
33 apt-get install openssh-server
34 echo "*****"
35 echo "OpenSSH-Server Instalado"
36 echo "*****"
37 # INSTALAR SERVIDOR FTP
38 apt-get install vsftpd
39 echo "*****"
40 echo "SERVIDOR FTP - vsftpd Instalado"
41 echo "*****"
42 # INSTALAR COMPRESORES
43 apt-get install p7zip-full unzip zip
44 echo "*****"
45 echo "Compresores p7zip-full unzip zip Instalados"
46 echo "*****"
47 # PAQUETES NECESARIOS PARA INSTALAR WEBMIN
48 apt-get install libauthn-pam-perl libio-pty-perl apt-show-versions libapt-pkg-perl libnet-ssleay-perl
49 apt-get install iptraf htop
50 echo " "
51 echo "*****"
52 echo "          Instalación finalizada          "
53 echo "*****"
```

Imagen: script de instalación básica: Apache, PHP, MySQL, phpMyAdmin...

Ahora lo que tendremos que hacer será ejecutar dicho script, como “root” y después mediante los siguientes comandos: `# sh instalacion_basica_debian.sh`

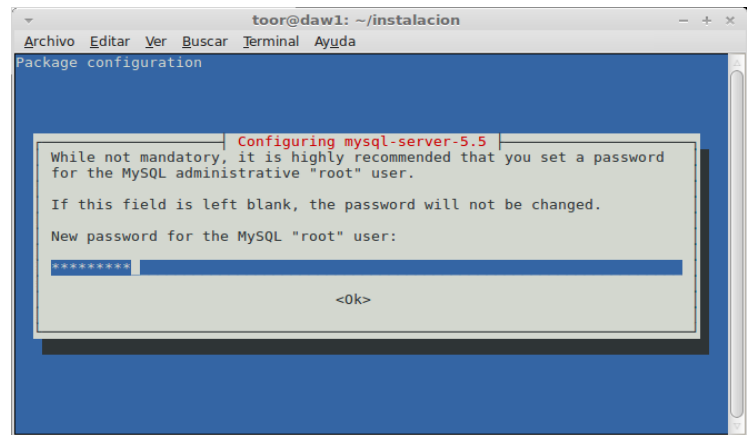
Podremos ver como va la ejecución de las instrucciones proporcionadas por el script:



```
toor@daw1: ~
Archivo Editar Ver Buscar Terminal Ayuda
*****
*  INSTALACION DE PAQUETES SERVIDOR DEBIAN  *
*****
Hit http://security.debian.org wheezy/updates Release.gpg
Hit http://security.debian.org wheezy/updates Release
Hit http://security.debian.org wheezy/updates/main Sources
```

Con nuestro script, lo que no tenemos controlado son las respuestas “Y/N” que tendrá que escribir el usuario para confirmar o cancelar la instalación de los distintos paquetes.

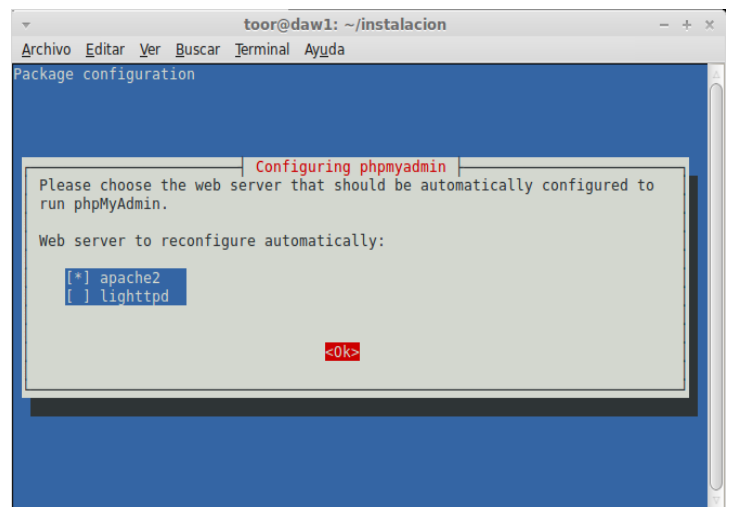
Después para el servidor MySQL, tendremos que indicarle la contraseña de root “1234567890”. Una vez introducida y confirmada dicha contraseña



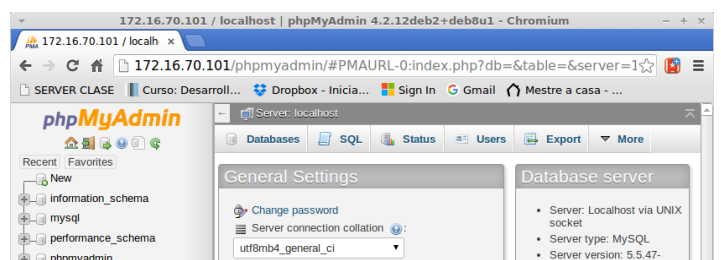
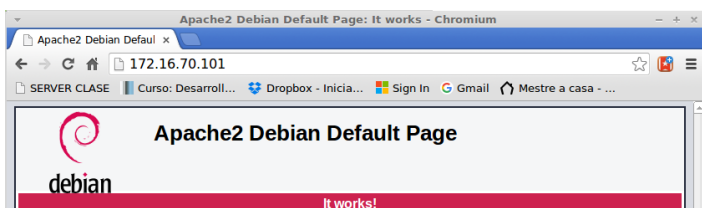
En la instalación de phpMyAdmin nos preguntará qué servidor web habrá que reconfigurar, al tener Apache, lo indicaremos y aceptaremos. Después nos preguntará si queremos “configurar la base de datos para phpmyadmin con dbconfig-common, contestaremos “Yes” e introduciremos la contraseña de root del servidor MySQL (recordemos “1234567890”).

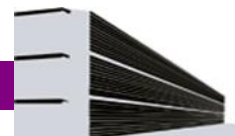
La instalación seguirá con “vsftpd”, el servidor FTP para linux de forma automática.

Después instalará los paquetes “iptraf” y “htop” más los compresores p7zip unzip zip.



Ahora desde nuestro navegador web de nuestro equipo remoto podemos comprobar que tenemos “Apache 2.2.22” y “phpMyAdmin” instalados en nuestro Debian proporcionado por el centro:





Instalaremos “webmin” más concretamente la versión 1.780 en nuestro Debian para una gestión gráfica de forma remota desde cualquier equipo dentro de la intranet. Copiaremos por “scp” el archivo con extensión DEB en el home de nuestro usuario “toor”:

```
vesprada@pcxx:~/Descargas$ scp webmin_1.780_all.deb toor@172.16.70.101:/home/toor/  
toor@172.16.70.101's password:  
webmin_1.780_all.deb 100% 27MB 13.3MB/s 00:02
```

Acto seguido instalaremos el paquete como “root” (recordemos que durante la instalación básica hemos descargado los paquetes necesarios para poder instalar Webmin):

```
Archivo Editar Ver Buscar Terminal Ayuda  
root@dawl:/home/toor# dpkg -i webmin_1.780_all.deb  
Selecting previously unselected package webmin.  
(Reading database ... 30854 files and directories currently installed.)  
Preparing to unpack webmin_1.780_all.deb ...  
Unpacking webmin (1.780) ...  
Setting up webmin (1.780) ...  
Webmin install complete. You can now login to https://dawl:10000/  
as root with your root password, or as any user who can use sudo  
to run commands as root.  
Processing triggers for systemd (215-17+deb8u3) ...  
root@dawl:/home/toor#
```

Configuración.

Configuración del sistema: toor con permisos de root.

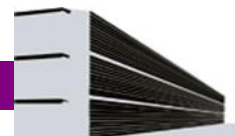
En principio nuestro usuario “toor” aunque lo hemos añadido al grupo de administradores del sistema “root”, no dispone de los permisos necesarios y tenemos que usar la contraseña de “root” para la instalación o cualquier otra acción, para ello instalaremos el paquete “sudo” con lo que nos aparecerá el archivo “/etc/sudoers” y dentro de este podremos darle los permisos que consideremos oportunos: `# apt-get install sudo`. Editaremos el archivo “sudoers” y con ello ya tendremos lo que necesitamos. Guardaremos los cambios.

```
toor@dawl: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
GNU nano 2.2.6 File: sudoers  
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"  
  
# Host alias specification  
  
# User alias specification  
  
# Cmnd alias specification  
  
# User privilege specification  
root    ALL=(ALL:ALL) ALL  
toor    ALL=(ALL:ALL) ALL  
# Allow members of group sudo to execute any command  
%sudo   ALL=(ALL:ALL) ALL
```

Carpetas para toor.

Dentro de su “/home” crearemos la carpeta scripts y guardaremos los “*.sh” y “*.sql” para reutilizar en caso necesario, recordemos que al tener SSH disponible podemos utilizar el comando “scp” (secure copy) para enviar los ficheros. Tampoco se descarta la posibilidad de utilizar el servidor FTP para lograr el mismo objetivo.

```
Terminal  
Archivo Editar Ver Buscar Terminal Ayuda  
vesprada@pcxx:~/Descargas$ scp -r crear_borrar_usuarios_mysql/ toor@172.16.70.101:/home/toor/scripts  
toor@172.16.70.101's password:  
borrar_usuarios_mysql.sql 100% 244 0.2KB/s 00:00  
crear_usuarios_mysql.sql 100% 861 0.8KB/s 00:00
```

Configuración avanzada.

Apache.

Como tenemos instalado la versión 2.2.22 en nuestro debian. Modificaremos el fichero “apache2.conf” que se encuentra en “/etc/apache2”, al final del archivo pondremos una línea donde indicaremos el nombre de nuestro servidor para que Apache sepa que es la propia máquina host **“ServerName daw1”**.

Comprobaremos que funciona cuando reiniciamos el servicio mediante **# /etc/init.d/apache2 restart**.

Además cambiaremos el tipo de archivo que mostrará el servidor apache cuando un usuario realiza una petición, queremos que en vez de buscar un **“index.html”** sea antes un archivo **“index.php”** por lo que tendremos que modificar el siguiente fichero que se encuentra en **“/etc/apache2/mods-enabled/dir.conf”** y pondremos **“index.php”** antes de todos los demás.

Primero accederemos al **“/etc/apache2/sites-available”** y crearemos nuestro propio sitio, en la configuración del archivo indicaremos la ruta y las directivas que contendrá. Para ello editaremos el archivo de configuración por defecto **“default”** por **“intranet_basic.conf”** quedando de la siguiente forma:

```
toor@daw1: /etc/apache2/sites-available
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 File: intranet_basic.conf
<VirtualHost *:80>
    ServerAdmin daw2016@localhost

    DocumentRoot /srv/intranet
    <Directory /srv/>
        Options All -Indexes
        AllowOverride None
        allow from all
    </Directory>

    <Directory /srv/intranet/>
        Options All -Indexes
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
```

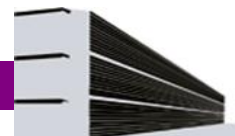
```
#Controlar el acceso al recurso por HTTP
<Directory /srv/intranet/fotos/>
    Options Indexes FollowSymLinks
    AuthType Basic
    AuthName "Acceso Restringido: FOTOS"
    AuthUserFile /etc/apache2/.passwd/.htpasswd
    Require valid-user
</Directory>

#Que archivos ejecutara dentro del sitio a cargar
<IfModule mod_dir.c>
    DirectoryIndex login.php index.php
</IfModule>

# mensajes de error
ErrorLog ${APACHE_LOG_DIR}/Intranet_error.log
LogLevel warn
CustomLog ${APACHE_LOG_DIR}/Intranet_access.log combined
</VirtualHost>
```

Crearemos en **“/srv”** la carpeta **“/intranet”** donde copiaremos el proyecto que descargaremos desde un equipo remoto y mediante **“secure copy”** enviaremos el archivo comprimido al servidor debian.

Guardaremos los cambios.



MySQL.

Finalizada la instalación de MySQL tendremos que realizar algunas operaciones más de forma que obtengamos nuestro entorno MySQL más seguro:

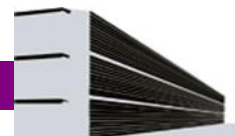
1. Ejecutaremos un script (que trae por defecto MySQL) para eliminar algunas configuraciones establecidas por defecto y bloquear el acceso a nuestro sistema de base de datos, para ello escribiremos en el terminal: `# mysql_secure_installation`.
 - Tendremos que introducir la contraseña que habíamos indicado antes durante la instalación y después nos preguntará si queremos cambiarla, responderemos “n” (NO).
 - Nos preguntará si eliminamos los usuarios anónimos que se puedan conectar a MySQL, de momento con el usuario “root” nos bastamos, en el futuro podremos crear más usuarios con privilegios si lo consideramos necesario, respondemos “Y” (Yes).
 - Desabilitaremos que el usuario root de MySQL pueda logearse remotamente, indicamos “Y”.
 - Quitaremos la base de datos “test” y su acceso, indicamos “Y” (Yes).
 - Finalmente, nos pide si recargamos la tabla de privilegios, indicamos “Y” (Yes).
2. Y con esto hemos finalizado la instalación y configuración del “Sistema Gestor de Base de Datos” MySQL.

Ahora crearemos los usuarios con los que trabajaremos con la base de datos, para ello dispondremos de un sql con las instrucciones necesarias, utilizaremos un navegador desde una máquina remota y accedemos al phpMyAdmin para poder ejecutarlo en el servidor debian, cargaremos el sql “crear_usuarios_mysql” importándolo, y nos aparecerá:

| Usuario | Servidor | Contraseña | Privilegios globales | Conceder | Acción |
|---|-----------|------------|----------------------|----------|----------------------------------|
| <input type="checkbox"/> admin_intranet | % | Sí | USAGE | No | Editar los privilegios Exportar |
| <input type="checkbox"/> admin_intranet | localhost | Sí | USAGE | No | Editar los privilegios Exportar |
| <input type="checkbox"/> debian-sys-maint | localhost | Sí | ALL PRIVILEGES | Sí | Editar los privilegios Exportar |
| <input type="checkbox"/> phpmyadmin | localhost | Sí | USAGE | No | Editar los privilegios Exportar |
| <input type="checkbox"/> root | 127.0.0.1 | Sí | ALL PRIVILEGES | Sí | Editar los privilegios Exportar |
| <input type="checkbox"/> root | :::1 | Sí | ALL PRIVILEGES | Sí | Editar los privilegios Exportar |
| <input type="checkbox"/> root | localhost | Sí | ALL PRIVILEGES | Sí | Editar los privilegios Exportar |
| <input type="checkbox"/> usuario_intranet | % | Sí | USAGE | No | Editar los privilegios Exportar |

Tendremos dos usuarios: **admin_intranet** con acceso remoto o local, todos los permisos; **usuario_intranet** con acceso remoto que podrá realizar “select y update” ambos usuarios sólo para la base de datos IntranetBatoi.

Ya desde el servidor remoto, siendo root, ejecutaremos el script: `# sh crear_usuarios_mysql.sql`.



Este es el script utilizado para crear los dos usuarios para la base de datos “IntranetBatoi”:

```
1  -- SELECCIONAR LA BASE DE DATOS
2  USE mysql;
3  -- USUARIO NORMAL
4  CREATE USER 'usuario_intranet'@'%' IDENTIFIED BY '1234567890';
5  GRANT SELECT,UPDATE,INSERT ON IntranetBatoi.* TO 'usuario_intranet'@'%';
6  -- USUARIO ADMINISTRADOR CON ACCESO A TODO / MISMO ROL QUE ROOT
7  -- PARA LOCALHOST
8  FLUSH PRIVILEGES;
9  CREATE USER 'admin_intranet'@'localhost' IDENTIFIED BY '1234567890' ;
10 GRANT ALL PRIVILEGES ON IntranetBatoi.* TO 'admin_intranet'@'localhost' WITH GRANT OPTION;
11 -- GRANT RELOAD,PROCESS ON IntranetBatoi.* TO 'admin_intranet'@'localhost';
12 FLUSH PRIVILEGES;
13 -- USUARIO ADMINISTRADOR CON ACCESO ROOT
14 -- PARA LA RED EN LA QUE SE ENCUENTRA
15 CREATE USER 'admin_intranet'@'%' IDENTIFIED BY '1234567890';
16 GRANT ALL PRIVILEGES ON IntranetBatoi.* TO 'admin_intranet'@'%' WITH GRANT OPTION;
17 -- GRANT RELOAD,PROCESS ON IntranetBatoi.* TO 'admin_intranet'@'%';
18 FLUSH PRIVILEGES;
```

En caso de que queramos cambiar la contraseña, fácilmente lo podremos realizar desde “PhpMyAdmin” pero con un usuario con los permisos suficientes (por ejemplo “root”).

Podemos realizar dicha acción mediante las siguientes sentencias sql:

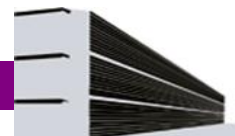
```
1  SET PASSWORD FOR 'usuario_intranet'@'%' = password('ejemplo');
2
3  SET PASSWORD FOR 'admin_intranet'@'%' = password('ejemplo2');
4
5  SET PASSWORD FOR 'admin_intranet'@'localhost' = password('ejemplo3');
```

Además para poder conectarnos por “Workbench” tendremos que modificar la configuración de MySQL, para ello iremos a “/etc/mysql/my.cnf” y tendremos que comentar el siguiente código:

```
# COMENTAR PARA QUE PERMITA ACCESO
#skip-external-locking
```

```
# COMENTAR PARA QUE ESCUCHE TODO
#bind-address = 127.0.0.1
```

Después reiniciaremos el servidor mysql: `# /etc/init.d/mysql restart`.

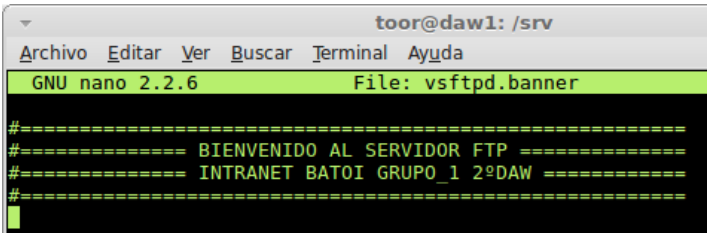


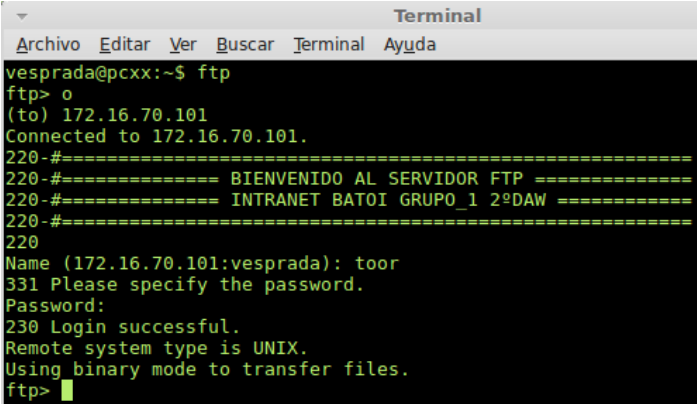
Vsftpd (servidor ftp).

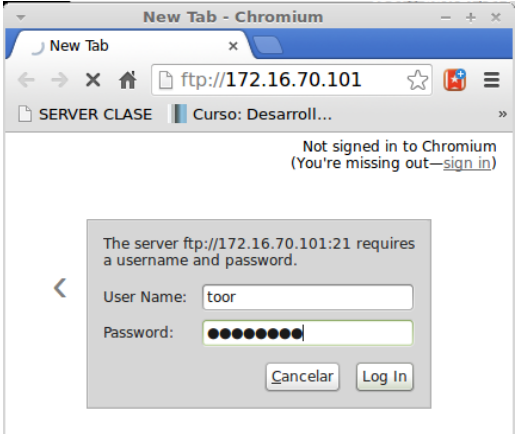
Aparte de tener el servidor “SSH”, realizar conexiones remotas y envío bidireccional de cualquier tipo de información (ficheros, carpetas, etc.), creemos que sería conveniente que el usuario “toor” dispusiera de un servidor “ftp” para subir archivos.

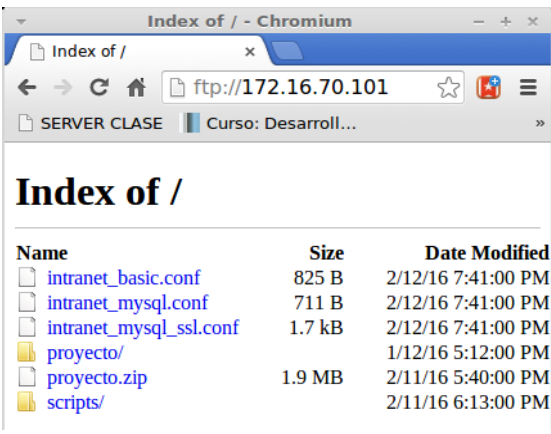
Al haber instalado el servidor, se habrá creado en “/srv” la carpeta “/ftp” con permisos 755, como propietario “root” y grupo “ftp”. Modificaremos el fichero “vsftpd.conf” y realizaremos los siguientes cambios:

| | |
|--|----------------------------|
| No permitiremos el usuario anonymous, puesto que el servicio ftp será utilizado por “toor”. | anonymous_enable=NO |
| Sólo los usuarios locales del SO podrán logearse. | local_enable=YES |
| Habilitaremos que sea posible escribir. | write_enable=YES |
| Cerraremos la sesión del usuario si se queda “ocioso” estableciendo 300 segundos. | idle_session_timeout=300 |
| Controlaremos el tiempo en el caso de que no se produzca ningún progreso cuando se produzca la transferencia de archivos, por lo que se cerrará la conexión en 60 segundos. | data_connection_timeout=60 |
| Crearemos el fichero “/etc/vsftpd.banner” para mostrar mensaje de bienvenida y así que el usuario sepa en qué servidor se encuentra. Como “root” crearemos el fichero, el cual tendrá permisos 644 y como propietario y grupo “root”. Después reiniciaremos el servicio con “service vsftpd restart” o “/etc/init.d/vsftpd restart”. | |

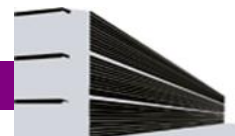








NOTA: la navegación en Opera da problemas con la identificación, con Chrome muestra la carpeta del usuario toor, mientras que en Firefox podemos navegar por las carpetas de todo el sistema.



Actualizar: PhpMyAdmin(v.4.5.4) + PHP(5.5.30).

1. Para ello lo primero será descargarnos la versión para todos los lenguajes para ello, como “root” escribiremos el siguiente código y después de descargar el fichero, descomprimos:

```
# wget https://files.phpmyadmin.net/phpMyAdmin/4.5.4/phpMyAdmin-4.5.4-all-languages.zip
# unzip phpMyAdmin-4.5.4-all-languages.zip
```

2. Una vez descargado, buscaremos dónde se encuentra el directorio de instalación de “phpMyAdmin”, en nuestro caso, que usamos un Debian 7.9., usaremos el comando `# find / -name “phpmyadmin”` y nos indicará que se encuentra en “/usr/share/phpmyadmin”.
3. Nos situaremos en “/usr/share/phpmyadmin” y borramos el contenido con `# rm -Rf *`.
4. Volvemos a la carpeta descomprimida de la nueva versión de “phpMyAdmin”, desde allí moveremos todo a la carpeta de instalación: `# mv * /usr/share/phpmyadmin`.
5. Reiniciaremos los servicios apache2.22 y MySQL.

6. Al acceder a phpmyadmin desde un navegador web remoto, nos indicará que necesita una versión de **php 5.5+** por lo que tendremos que añadir a nuestro fichero “sources.list” el repositorio correspondiente:

```
# actualizar la version PHP
deb http://packages.dotdeb.org wheezy-php55 all
deb-src http://packages.dotdeb.org wheezy-php55 all
```

7. Después actualizaremos los repositorios con: `# apt-get update`. Pero al final dará un error de clave pública por lo que tendremos que añadir otro repositorio y añadir su clave pública:

```
root@daw1:/etc/apt# wget http://www.dotdeb.org/dotdeb.gpg
root@daw1:/etc/apt# apt-key add dotdeb.gpg
```

8. Una vez finalizado el paso 7, actualizaremos los repositorios y podremos instalar la nueva versión de **php 5.5.30** (en vez de php 5.4.45).

```
Archivo Editar Ver Buscar Terminal Ayuda
root@daw1:/etc/apt# php5 -v
PHP 5.5.30-1~dotdeb+7.1 (cli) (built: Oct 1 2015 18:17:01)
Copyright (c) 1997-2015 The PHP Group
Zend Engine v2.5.0, Copyright (c) 1998-2015 Zend Technologies
with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2015, by Zend Technologies
```

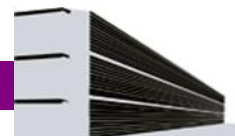
9. Nos queda reiniciar apache2 y acceder desde el navegador web, y podremos ver que tenemos phpMyAdmin actualizado y php5.5, tal y como se muestra en la imagen.

Servidor web

- Apache/2.2.22 (Debian)
- Versión del cliente de base de datos: libmysql - 5.5.47
- extensión PHP: mysql
- Versión de PHP: 5.5.30-1~dotdeb+7.1

phpMyAdmin

- Acerca de esta versión: 4.5.4, versión estable más reciente: 4.5.4.1
- Documentación
- Wiki
- Página oficial de phpMyAdmin
- Contribuir
- Obtener soporte
- Lista de cambios



10. Ahora tendremos dos problemas, (1) el almacenamiento de configuración phpMyAdmin no está completamente configurado, algunas funcionalidades extendidas fueron deshabilitadas, (2) El archivo de configuración ahora necesita una frase secreta (blowfish_secret).

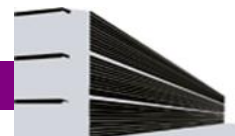
11. Para el primer caso, nos logearemos como “root” (o cualquier usuario con permisos de root) en phpmyadmin y crearemos la base de datos “phpmyadmin” (el proceso es automático).

12. Para lo del “blowfish secret”, primero crearemos una copia del fichero “config.inc.php” y renombramos el fichero a “config.sample.php” y buscar “\$cfg['blowfish_secret']” y añadir un texto cualquiera:

```
/**
 * This is needed for cookie based authentication
 * cookie
 */
$cfg['blowfish_secret'] = 'dawlcurso2016';
```



Reiniciamos Apache2 y solucionado.



Configurar el recurso “/fotos”.

Configurar acceso a las fotografías en “/srv/intranet/fotos” (Autenticación Básica).

Si queremos que un usuario tenga acceso a dicho recurso, podemos utilizar el usuario y la contraseña almacenada en un fichero. Nos situaremos en “/etc/apache2/”:

- Crearemos la carpeta “.passwd”: `# mkdir .passwd`.
- El grupo y el usuario de ésta por defecto será “root” con permisos 755, **NO tendremos que cambiarlo por:** `# chmod 644 .passwd/`.
- Crearemos el fichero “.htpasswd” con: `# htpasswd -c /etc/apache2/.passwd/.htpasswd admin`, utilizamos el parámetro “-c” para crear el fichero por primera vez, lo quitaremos para añadir más usuarios. No tendremos que cambiar los permisos del fichero (644 al crearse).

Podemos realizar la protección de la carpeta por “htpasswd” creando en “/etc/apache2/sites-available” la configuración de nuestro sitio mediante el siguiente código (imagen a la derecha).

Después lo habilitaríamos mediante el comando “**sudo a2ensite [fichero]**”, después reiniciaremos el servidor apache:

```
$sudo service apache2 restart
```

Se recomienda deshabilitar la configuración por defecto del sitio (**a2dissite**) ya que por defecto que nos proporciona apache: **000-default.conf** para que se quede funcionando la configuración de nuestro sitio.

Ya reiniciado el servidor apache, desde un navegador web accederemos a dicho recurso y podremos ver el resultado:

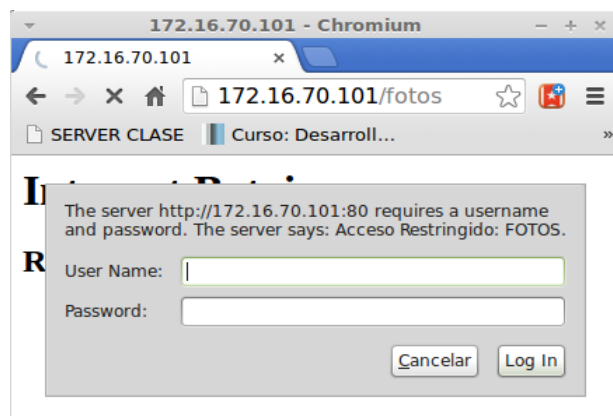
```
GNU nano 2.2.6      File: intranet_basic.conf

    allow from all
</Directory>

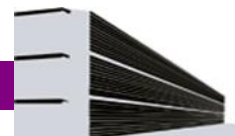
#Controlar el acceso al recurso por HTTP
<Directory /srv/intranet/fotos/>
    #Options Indexes FollowSymLinks
    AuthType Basic
    AuthName "Acceso Restringido: FOTOS"
    AuthUserFile /etc/apache2/.passwd/.htpasswd
    Require valid-user
</Directory>

#Que archivos ejecutara dentro del sitio a cargar
<IfModule mod_dir.c>
    DirectoryIndex login.php index.php
</IfModule>

# mensajes de error
ErrorLog ${APACHE_LOG_DIR}/Intranet_error.log
LogLevel warn
CustomLog ${APACHE_LOG_DIR}/Intranet_access.log combined
</VirtualHost>
```



Para la carpeta “/srv/intranet/fotos” pondremos todos los permisos 777 para que sea posible “subir” la fotografía tomada por la webcam o dispositivo similar. Recordemos también que dicha carpeta está protegida si alguien quiere acceder al recurso desde un navegador web.



Configurar acceso a las fotografías en “/srv/intranet/fotos” (Autenticación por MySQL i y ii).

La solución anterior puede servirnos, pero si utilizamos un “sniffer” el usuario y la contraseña se podrán ver en texto plano por lo que podemos utilizar la autenticación a través de la creación de una base de datos MySQL que posea los usuarios y las contraseñas que se podrán validar para visitar nuestro sitio web. Al ser un método nativo de Apache nos proporcionará el intercambio de usuarios y contraseñas de forma segura.

Instalación y activación del módulo:

```
$sudo apt-get install libapache2-mod-auth-mysql
```

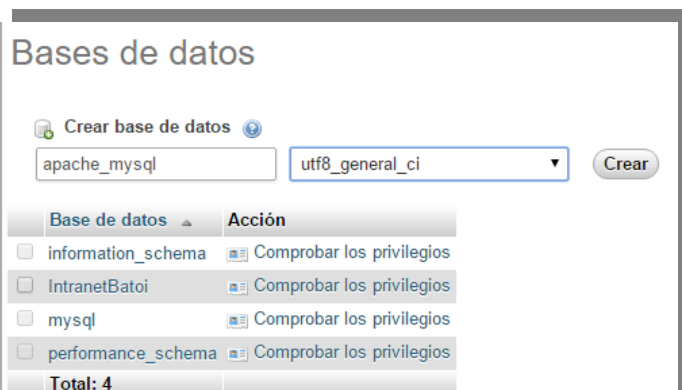
```
$sudo a2enmod auth_mysql
```

```
$sudo service apache2 restart
```

Crearemos la base de datos MySQL para la autenticación, podremos usar “phpMyAdmin” o “MySQL Workbench” de forma remota (si ya lo tenemos configurado en nuestro servidor para conexiones remotas). Crearemos la base de datos indicando el tipo de idioma adecuado “utf8_general_ci”.

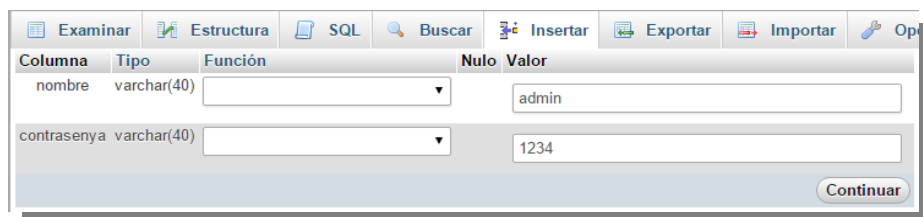
Crearemos la tabla “usuarios” e indicaremos que contendrá dos tablas.

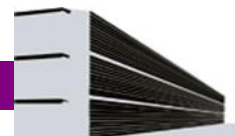
Después crearemos dos tablas, por ejemplo “usuario” y “contrasena” de tipo “VARCHAR(40)” (o como queramos), nosotros proponemos la siguiente configuración que mostramos en la captura siguiente:



Utilizaremos como motor de almacenamiento “MyISAM” ya que no tendremos otras bases de datos enlazadas.

Guardaremos los cambios. Insertaremos el usuario “admin” con password “1234” por ejemplo y lo guardaremos en dicha tabla.





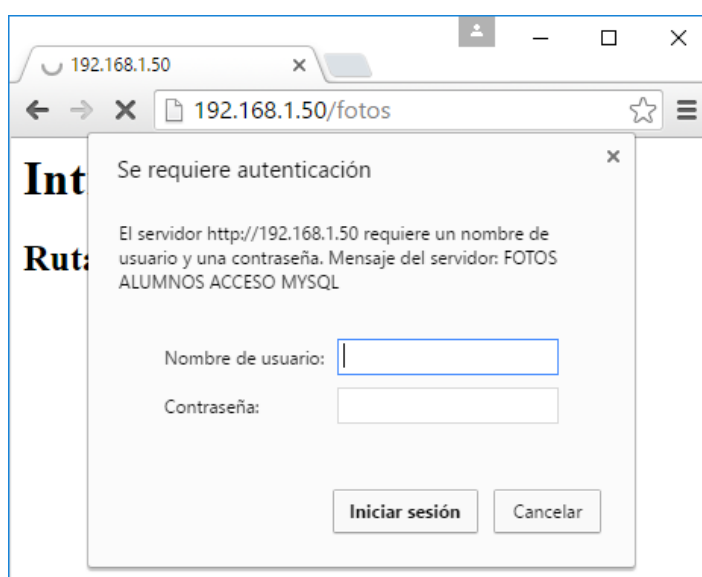
Además necesitaremos indicar al sistema que Apache puede utilizar y acceder a dicha base de datos para ver sus contenidos, crearemos un usuario que contenga derechos de acceso al cual llamaremos **“apacheadmin”**. Para ello desde “PHPMyAdmin” los crearemos, indicando el servidor y con privilegio global “SELECT” y contraseña “1234”:



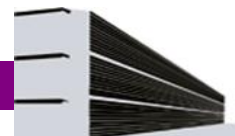
Ahora podremos ver la lista de usuarios entre ellos **“apacheadmin”**:

| | Usuario | Servidor | Contraseña | Privilegios globales | Conceder | Acción |
|--------------------------|----------------|-----------|------------|----------------------|----------|----------------------------------|
| <input type="checkbox"/> | admin_intranet | % | Sí | USAGE | No | Editar los privilegios Exportar |
| <input type="checkbox"/> | admin_intranet | localhost | Sí | USAGE | No | Editar los privilegios Exportar |
| <input type="checkbox"/> | apacheadmin | localhost | Sí | SELECT | No | Editar los privilegios Exportar |
| <input type="checkbox"/> | chvdaw | % | Sí | ALL PRIVILEGES | No | Editar los privilegios Exportar |
| <input type="checkbox"/> | chvdaw | ::1 | Sí | ALL PRIVILEGES | Sí | Editar los privilegios Exportar |
| <input type="checkbox"/> | chvdaw | localhost | Sí | ALL PRIVILEGES | Sí | Editar los privilegios Exportar |
| <input type="checkbox"/> | dani | % | Sí | USAGE | No | Editar los privilegios Exportar |

Ahora crearemos nuestro archivo de configuración en **“/etc/apache2/sites-available/”** **intranet_mysql.conf** en donde escribiremos el siguiente código para configurar el sitio web:



Pero no funciona, la solución es comentar “AuthMySQL_Authoritative” o poner valor “Off”.



Pero por otra parte podemos crear dentro de la carpeta “fotos” el archivo “.htaccess” mientras que en la configuración del sitio web “intranet_mysql.conf” indicamos que la configuración sea por defecto.

```
<Directory /srv/intranet/fotos/>
  AllowOverride AuthConfig
  Options All
</Directory>
```

Entonces el contenido del fichero “.htaccess” será el siguiente.

Con lo que podremos controlar el acceso a nuestro recurso a partir de los datos contenidos en la base de datos indicada.

Nótese que hemos indicado que la encriptación primeramente ha sido en texto plano, pero podría ser fácilmente visible mediante un “sniffer” por lo que cambiaríamos a “Crypt”, pero tendremos que “encriptar” las contraseñas de los usuarios.

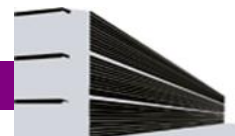
Para poder encriptar las contraseñas de los usuarios de la base de datos “apachemysql”

cuando insertamos podemos seleccionar a parte del tipo de dato, también el tipo de encryption:

| Columna | Tipo | Función | Nulo | Valor |
|-------------|--------------|---------|------|-------|
| nombre | varchar(40) | | | jose |
| contrasenia | varchar(130) | ENCRYPT | | 1234 |

Es decir, para “Crypt” en Apache, usaremos “ENCRYPT” en PHPMYADMIN

Con lo que conseguimos algo de “seguridad” al no mostrar la contraseña y sin utilizar texto plano.



Apache HTTPS/SSL.

Hasta ahora nuestro servidor permitía la navegación por el puerto “80” ya he habilitábamos el sitio web “intranet_mysql.conf”, ahora lo que queremos es deshabilitar este sitio que se encuentra en la ruta “/etc/apache2/sites-available/intranet_mysql.conf” y habilitar otro para que atienda peticiones por el puerto “443” llamado “intranet_mysql_ssl.conf”.

Activaremos el módulo SSL con el siguiente comando: **# a2enmod ssl.**

Reiniciaremos el servidor Apache y comprobaremos los puertos por los que escucha apache:

Después crearemos nuestro propio certificado autofirmado con OpenSSL, así que nos situaremos desde la carpeta “/etc/ssl” y siendo “root”: **# openssl genrsa 2048 > /etc/ssl/private/ssl-cert.key**

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@dawl:/etc/ssl# openssl genrsa 2048 > /etc/ssl/private/ssl-cert.key
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
root@dawl:/etc/ssl#
```

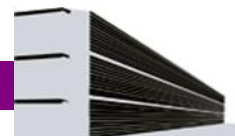
Ahora modificaremos los propietarios y los permisos (dentro de “/etc/ssl/private”):

```
# chown root:ssl-cert ssl-cert.key
# chmod 640 ssl-cert.key
```

Como ya tenemos nuestra clave privada, crearemos un certificado X.509 de un año de validez, dentro del directorio “/etc/ssl/private” (porque tenemos el fichero creado “ssl-cert.key”):

```
# openssl req -new -x509 -nodes -sha1 -days 365 -key ssl-cert.key > IntranetBatoi.pem
```

Nos queda mover el fichero “IntranetBatoi.pem” a “/etc/ssl/certs”, que es su lugar.



Ahora tenemos que crear nuestro “sites-available” para ello copiaremos la configuración por defecto que trae el apache para ssl, el cual se llama “default-ssl” y lo cambiaremos por “intranet_mysql_ssl.conf”, es decir, a la configuración ya disponible la adecuaremos para que sólo acepte peticiones por el puerto seguro “443”.

Modificaremos las líneas correspondientes al certificado y la clave generados anteriormente:

```
toor@daw1: ~/scripts
GNU nano 2.2.6 File: intranet_mysql_ssl.conf Modified

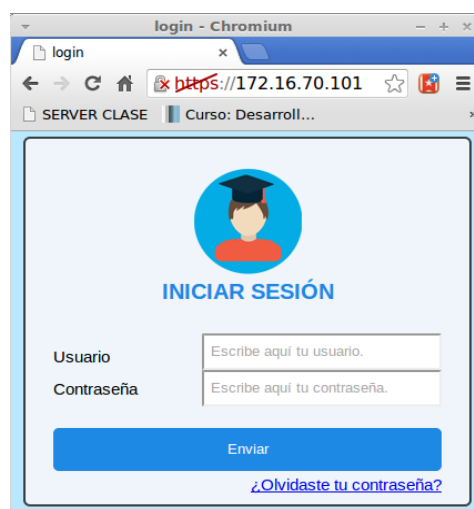
CustomLog ${APACHE_LOG_DIR}/intranet_ssl_access.log combined

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2.2-common/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/IntranetBatoi.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert.key
```

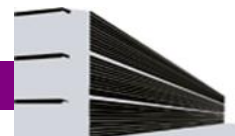
Eliminaremos las líneas comentadas, porque probado su funcionamiento en otras prácticas no afectará a la ejecución de nuestro servidor Apache.

El siguiente paso será deshabilitar “intranet_mysql.conf” (y todas las configuraciones anteriores), habilitar “intranet_mysql_ssl.conf” y reiniciar el servidor apache para que surtan efecto los cambios realizados.



Al acceder desde un navegador remoto nos envía al DocumentRoot “por defecto” en “/var/www”. Pero si añadimos “https://” a la ip de nuestro servidor, nos redirige al DocumentRoot de nuestro sitio.

NOTA: no es necesario modificar el fichero “/etc/apache2/ports.conf” y añadir el puerto 443 a continuación del puerto 80 puesto que al habilitar el módulo “ssl” en apache, hemos indicado en la configuración de nuestro nuevo sitio que “si el módulo está activo” (`<IfModule mod_ssl.c>`) se podrá acceder al recurso, carpetas, etc. **Habrà que bien poner un index.html que redirija o buscar otra forma para que cargue de forma automática (ejemplo: moodle).**



El fichero “intranet_mysql_ssl.conf”.

Ahora pasamos a desglosar el contenido de dicho fichero, como queremos que el sitio sea accesible por SSL, recordemos que hemos habilitado el módulo “ssl” (comentado anteriormente) mediante “IfModule” que condicionamos la activación, entonces “se ejecutará” todo el contenido, es decir tendremos nuestro host virtual con sus directivas y configuraciones.

Línea 2: el host virtual escucha por el puerto “443”.

Línea 5: indicamos que el directorio raíz se encuentra en “/srv/intranet”.

Líneas 6 – 10: Aplicamos directivas a la carpeta “/srv”, activamos todas las opciones menos que se pueda indexar el contenido de la carpeta, evitamos así que se pueda ver el contenido. Permitimos el acceso a todos los usuarios sin restricciones.

```
intranet_mysql_ssl.conf x
1  <IfModule mod_ssl.c>
2  <VirtualHost *:443>
3      ServerAdmin daw2016@localhost
4
5      DocumentRoot /srv/intranet
6      <Directory /srv/>
7          Options All -Indexes
8          AllowOverride None
9          allow from all
10     </Directory>
11
```

```
12     <Directory /srv/intranet/>
13         ErrorDocument 401 /errorApache/error401.html
14         ErrorDocument 403 /errorApache/error403.html
15         ErrorDocument 404 /errorApache/error404.html
16         ErrorDocument 500 /errorApache/error500.html
17         Options All -Indexes
18         AllowOverride None
19         Order allow,deny
20         Allow from all
21     </Directory>
```

Líneas 12 – 22: Indicamos que apache muestre nuestras propias páginas de error personalizadas para que apache no muestre por defecto sus propios html de error (o es posible utilizar php). También se permite el acceso a todos.

Líneas 23 – 27: Para el directorio dónde se encuentran las fotos, estableceremos la directiva de que con “AuthConfig” indicamos que utilice el fichero “.htaccess” que se encuentra en la carpeta “/srv/intranet/fotos” pero oculto. Activaremos todas las opciones.

```
23     #Controlar el acceso al recurso por MYSQL
24     <Directory /srv/intranet/fotos>
25         AllowOverride AuthConfig
26         Options All
27     </Directory>
```

Líneas 30 – 32: Mediante la directiva DirectoryIndex, indicamos que cargue el fichero con el nombre y la extensión que queramos. Conseguimos cambiar el que está establecido por defecto.

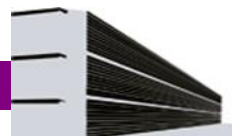
```
29     #Que archivo carga al acceder al sitio
30     <IfModule mod_dir.c>
31         DirectoryIndex login.php index.php
32     </IfModule>
```

```
43
44     ErrorLog ${APACHE_LOG_DIR}/intranet_ssl_error.log
45
46     LogLevel warn
47
48     CustomLog ${APACHE_LOG_DIR}/intranet_ssl_access.log combined
```

Líneas 44 – 48: Indicamos el nombre de los ficheros de error y de acceso creados mientras el sitio esté habilitado, nivel de “log” y que siga utilizando por defecto el directorio utilizado por apache para tal fin.

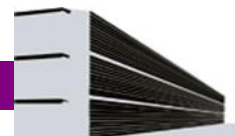
```
50     # SSL Engine Switch:
51     # Enable/Disable SSL for this virtual host.
52     SSLEngine on
53
54     SSLCertificateFile /etc/ssl/certs/IntranetBatoi.pem
55     SSLCertificateKeyFile /etc/ssl/private/ssl-cert.key
```

Líneas 52 – 55: Habilitamos SSL, indicamos dónde se encuentran la clave pública y el certificado generado (visto anteriormente).



Personalizar “ErrorDocument”.

Dentro de “/srv/intranet/” crearemos la carpeta “**errorApache**” dentro contendrá las páginas “**html**” o bien “**php**” correspondientes a los errores que da apache por defecto: 401, 403, 404 y 500. Las páginas contendrán en principio y por su sencillez: CSS, LESS y PHP.



Importando el proyecto.

Copia remota del proyecto.

Nos hemos descargado desde “dropbox” el proyecto comprimido en nuestra máquina remota, después mediante el siguiente comando lo enviaremos al servidor debian:

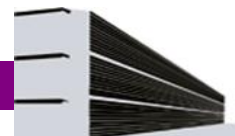
```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
vesprada@pcxx:~/Descargas$ scp proyecto.zip toor@172.16.70.101:/home/toor
toor@172.16.70.101's password:
proyecto.zip                                100% 2060KB  2.0MB/s  00:00
```

Descomprimiremos el fichero (siendo usuario toor) y podremos ver que los permisos serán 744, siendo propietario toor y grupo root. Después tendremos que ser root, situarnos dentro de la carpeta “/proyecto” y utilizar el siguiente comando: `root@dawl:/home/toor/proyecto# cp * -R /srv/intranet`

Después será necesario cambiar los permisos, por eso nos situaremos en la carpeta “/srv” e introducir el siguiente comando: `# chmod 755 * -R intranet/`.

No olvidemos cambiar el grupo también al proyecto: `# chown root.www-data -R intranet/`.

Para la carpeta “/srv/intranet/fotos” pondremos todos los permisos 777 para que sea posible “subir” la fotografía tomada por la webcam o dispositivo similar. Recordemos también que dicha carpeta está protegida si alguien quiere acceder al recurso desde un navegador web.



Copias de seguridad (backup's).

Uso de “cron”.

Desde nuestro servidor podemos automatizar las copias de seguridad de forma automática a través de un script que podría ser como este:

```
backup_BD_y_DATOS_auto.sh x
1 #!/bin/bash
2 # CREAR BASE DATOS + DATOS
3 # -----
4 # Obtener fecha y hora
5 FECHAHORA=$(date +%Y%m%d_%H:%M:%S)
6 USUARIO="admin intranet"
7 PASS="1234567890"
8 #-----
9 # crear el backup de la base de datos
10 # ESTRUCTURA DE LA BASE DE DATOS Y SUS DATOS
11 mysqldump -u $USUARIO -p$PASS --databases IntranetBatoi > $FECHAHORA"backup_IntranetBatoi_DB_DATA.bak"
12 #mysqldump -u $USUARIO -p$PASS --databases IntranetBatoi > $FECHAHORA"backup_IntranetBatoi_DB_DATA.sql"
13
14 # SOLO ESTRUCTURA DE LA BASE DE DATOS
15 mysqldump -d -u $USUARIO -p$PASS -d IntranetBatoi > $FECHAHORA" _IntranetBatoi_DB.bak"
16 #mysqldump -d -u $USUARIO -p$PASS -d IntranetBatoi > $FECHAHORA" _IntranetBatoi_DB.sql"
17
18 # SOLO LOS DATOS
19 mysqldump -t -u $USUARIO -p$PASS -t IntranetBatoi > $FECHAHORA" _IntranetBatoi_DATA.bak"
20 #mysqldump -t -u $USUARIO -p$PASS -t IntranetBatoi > $FECHAHORA" _IntranetBatoi_DATA.sql"
21 #
22 # FIN DEL FICHERO
23
```

Script en linux para realizar de forma autónoma, ya que proporcionamos el usuario y la contraseña correspondiente. Recordemos que dicho usuario debe tener todos los permisos necesarios para poder realizar la copia de la base de datos a través de la shell de linux.

Problema: aparece el nombre del usuario y la contraseña (legible) en el fichero, podemos ocultarlo y además también podemos indicar los permisos a la carpeta que contiene el script.

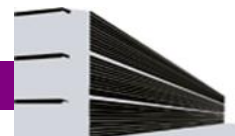
Pasos a seguir:

1. Siendo “toor” crearemos dentro de “/home/toor/scripts/” la carpeta “./backups” con permisos “760”. Dicha carpeta tendrá como propietario “toor” y “root” para el grupo.
2. Copiaremos nuestro script “backup_BD_y_DATOS_auto.sh” dentro de la carpeta que hemos creado con permisos “760” tendrá como propietario “toor” y “root” como grupo.
3. Como root accederemos al directorio “/etc”, ejecutaremos: `# crontab -l` (para ver la lista de tareas) y con el comando `# crontab -e` las editaremos añadiendo la línea correspondiente para que ejecute el script indicado:

```
#
# m h dom mon dow   command
* * * * * sh /home/toor/scripts/.backups/backup_BD_y_DATOS_auto.sh
```

Ejemplo: cada minuto se ejecutará el script.

4. Después guardaremos los cambios y reiniciaremos “cron”: `# /etc/init.d/cron [restart |stop |start]`.
5. Es posible que al detener el servicio, tengamos un “email” en el archivo “/var/mail/root” que nos indicará si se han producido errores, mediante el comando “cat” podremos ver el contenido.
6. **Problema,** los “bak” se guardarán en “/root” como: `644 root root`.

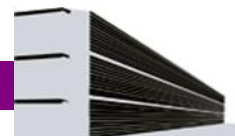


Copiar backups de “/root” a “/home/toor/scripts/.backups_file”.

Como no queremos que se guarden todos los archivos “.bak” en “/root” añadiremos un par de líneas más a nuestro cron, para ello como “root” y situados en “/etc”, ejecutaremos el comando “crontab -e” y añadiremos:

```
# m h dom mon dow  command
30 * * * * sh /home/toor/scripts/.backups/backup_BD_y_DATOS_auto.sh
31 * * * * cp *.bak /home/toor/scripts/.backups_file/
32 * * * * rm *.bak
```

- Cada “31 minutos” se realizará una copia de todos los ficheros con extensión “bak” y se copiarán a la carpeta de nuestro usuario “toor” localizada en “/home/toor/scripts/.backups_file/”.
- Un minuto después, se borrarán todos los ficheros “.bak” y así quedará la carpeta “/root” limpia. Guardaremos los cambios y reinicaremos el servicio “cron” para que surtan efecto los cambios.
- Como mejora también podríamos realizar una copia de la carpeta de la intranet “/srv/intranet” y realizar el mismo proceso, por ejemplo.



Realizar backup de forma remota.

Para ello dispondremos de un script que utilizado desde otro equipo (dentro de la intranet del centro) lo ejecute y pueda obtener tres tipos distintos de copia de seguridad:

- Base de datos + datos.
- Sólo la base de datos.
- Sólo los datos.

El script empleado para tal fin es el siguiente:

```
backup_BD_y_DATOS_remoto.sh x
1 #!/bin/bash/
2 # CREAR BASE DATOS + DATOS REMOTAMENTE
3 clear
4 echo "*****"
5 echo "**** BACKUP BASE DE DATOS INTRANET BATOI ****"
6 echo "*****"
7 echo " "
8 echo "Introduzca el usuario con permisos de root para acceder a MYSQL"
9 read USUARIO
10 echo " "
11 echo "Introduzca la IP del servidor"
12 read IPSERVER
13 echo " "
14 # Obtener fecha y hora
15 FECHAHORA=$(date +%Y%m%d_%H:%M:%S)
16 #-----
17 # Indicar la contraseña/modificar
18 PASS="1234567890"
19 # crear el backup de la base de datos
20 # ESTRUCTURA DE LA BASE DE DATOS Y SUS DATOS
21 mysqldump -u $USUARIO -p$PASS -h $IPSERVER --databases IntranetBatoi > $FECHAHORA "_backup_IntranetBatoi_DB_DATA.bak"
22 #mysqldump -u $USUARIO -p$PASS --databases IntranetBatoi > $FECHAHORA "_backup_IntranetBatoi_DB_DATA.sql"
23
24 # SOLO ESTRUCTURA DE LA BASE DE DATOS
25 mysqldump -d -u $USUARIO -p$PASS -h $IPSERVER -d IntranetBatoi > $FECHAHORA "IntranetBatoi_DB.bak"
26 #mysqldump -d -u $USUARIO -p$PASS -h $IPSERVER -d IntranetBatoi > $FECHAHORA "IntranetBatoi_DB.sql"
27
28 # SOLO LOS DATOS
29 mysqldump -t -u $USUARIO -p$PASS -h $IPSERVER -t IntranetBatoi > $FECHAHORA "IntranetBatoi_DATA.bak"
30 #mysqldump -t -u $USUARIO -p$PASS -h $IPSERVER -t IntranetBatoi > $FECHAHORA "IntranetBatoi_DATA.sql"
31
32 echo " "
33 echo "-----"
34 echo "Backup - Proceso finalizado"
35 echo "-----"
36
```

Entonces el encargado de realizar la copia de seguridad, deberá tener disponible el script, así como el usuario y la IP para poder realizar dicha acción. Entonces al ejecutarlo se crearan los ficheros con extensión **“bak”** (se pueden cargar en PhpMyAdmin o en Workbench) con la fecha y la hora del sistema.

```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
*****
**** BACKUP BASE DE DATOS INTRANET BATOI ****
*****

Introduzca el usuario con permisos de root para acceder a MYSQL
admin_intranet

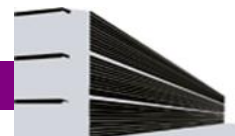
Introduzca la IP del servidor
172.16.70.101

-----
Backup - Proceso finalizado
-----
vesprada@pcxx:~/Descargas$
```

| Nombre | Tamaño |
|---|---------|
| 20160210_16:56:45_IntranetBatoi_DB.bak | 14,0 kB |
| 20160210_16:56:45_IntranetBatoi_DATA.bak | 43,5 kB |
| 20160210_16:56:45_backup_IntranetBatoi_DB_DATA... | 56,3 kB |
| backup_BD_y_DATOS_remoto.sh | 1,4 kB |

```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
vesprada@pcxx:~/Descargas$ ls
0XqU+Fr5.-
20160210_16:56:45_backup_IntranetBatoi_DB_DATA.bak
20160210_16:56:45_IntranetBatoi_DATA.bak
20160210_16:56:45_IntranetBatoi_DB.bak
backup_BD_y_DATOS_remoto.sh
vesprada@pcxx:~/Descargas$
```

El problema es que está visible la contraseña del USUARIO.



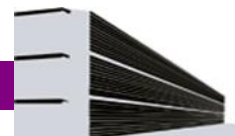
Realizar backup de forma local.

Es posible que en cualquier momento queramos realizar una copia por lo que podemos disponer del script correspondiente (muy parecido al que utilizamos para realizar la copia de forma remota) por ejemplo dentro de “/home/toor/scripts/.backups” o en una carpeta oculta para el usuario “toor” llamada también igual.

```
toor@daw1: ~/scripts/.backups
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
***** BACKUP BASE DE DATOS INTRANET BATOI *****
*****
Introduzca el usuario con permisos de root para acceder a MYSQL
admin_intranet

-----
Backup - Proceso finalizado
-----
toor@daw1:~/scripts/.backups$
```

```
toor@daw1: ~/scripts/.backups
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
toor@daw1:~/scripts/.backups$ ls
20160210_15:48:15_IntranetBatoi_DATA.bak
20160210_15:48:15_IntranetBatoi_DB.bak
20160210_15:48:15backup_IntranetBatoi_DB_DATA.bak
toor@daw1:~/scripts/.backups$
```



Comandos a utilizar para administrar nuestro servidor.

1. Obtener una “lista” de los usuarios logeados dentro de nuestro servidor como de forma remota:

Uso del comando “\$ last” (desde el servidor como por un terminal de forma remota por SSH) veremos conexiones desde el propio servidor como de forma remota, así las operaciones realizadas como “reboot”:

```
toor@daw1: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
toor@daw1:~$ last  
toor pts/0 172.16.62.200 Mon Feb 15 16:27 still logged in  
toor pts/1 172.16.62.200 Mon Feb 15 14:12 - 16:08 (01:55)  
toor pts/0 172.16.62.200 Mon Feb 15 14:10 - 16:08 (01:57)  
toor pts/1 172.16.62.200 Mon Feb 15 11:55 - 12:01 (00:05)  
toor pts/0 172.16.62.200 Mon Feb 15 08:13 - 12:01 (03:47)  
toor pts/1 172.16.62.200 Fri Feb 12 17:45 - 18:40 (00:54)  
toor pts/0 172.16.62.200 Fri Feb 12 17:40 - 18:41 (01:01)
```

Podemos utilizar el comando “\$ last > usuarios_on_of.txt” para guardar las conexiones.

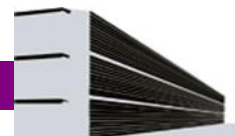
2. Ver los recursos utilizados de nuestro servidor dedicado:

Usando el comando “\$ htop” podremos ver los procesos, uso de la CPU, RAM utilizada y SWAP:

```
toor@daw1: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
1 [ 0.7%] Tasks: 28, 19 thr; 1 running  
2 [ 0.0%] Load average: 0.00 0.00 0.00  
Mem[|||||] 119/1024MB Uptime: 6 days, 22:05:01  
Swp[ ] 2/1024MB  


| PID  | USER     | PRI | NI | VIRT | RES   | SHR  | S | CPU% | MEM% | TIME+   | Command           |
|------|----------|-----|----|------|-------|------|---|------|------|---------|-------------------|
| 6763 | www-data | 20  | 0  | 114M | 15732 | 8376 | S | 0.0  | 1.5  | 0:00.14 | /usr/sbin/apache2 |
| 1    | root     | 20  | 0  | 2152 | 628   | 536  | S | 0.0  | 0.1  | 0:03.56 | init [2]          |
| 574  | root     | 20  | 0  | 2180 | 868   | 684  | S | 0.0  | 0.1  | 0:00.00 | /usr/sbin/cron    |
| 1555 | root     | 20  | 0  | 1968 | 748   | 616  | S | 0.0  | 0.1  | 0:00.00 | /sbin/getty 38400 |
| 1949 | root     | 20  | 0  | 1804 | 540   | 440  | S | 0.0  | 0.1  | 0:00.00 | /bin/sh /usr/bin/ |
| 2255 | mysql    | 20  | 0  | 347M | 39908 | 6760 | S | 0.0  | 3.8  | 0:00.10 | /usr/sbin/mysqld  |
| 2256 | mysql    | 20  | 0  | 347M | 39908 | 6760 | S | 0.0  | 3.8  | 0:00.14 | /usr/sbin/mysqld  |
| 2257 | mysql    | 20  | 0  | 347M | 39908 | 6760 | S | 0.0  | 3.8  | 0:00.10 | /usr/sbin/mysqld  |
| 2258 | mysql    | 20  | 0  | 347M | 39908 | 6760 | S | 0.0  | 3.8  | 0:00.10 | /usr/sbin/mysqld  |
| 2259 | mysql    | 20  | 0  | 347M | 39908 | 6760 | S | 0.0  | 3.8  | 0:00.11 | /usr/sbin/mysqld  |
| 2260 | mysql    | 20  | 0  | 347M | 39908 | 6760 | S | 0.0  | 3.8  | 0:00.11 | /usr/sbin/mysqld  |
| 2261 | mysql    | 20  | 0  | 347M | 39908 | 6760 | S | 0.0  | 3.8  | 0:00.11 | /usr/sbin/mysqld  |
| 2262 | mysql    | 20  | 0  | 347M | 39908 | 6760 | S | 0.0  | 3.8  | 0:00.11 | /usr/sbin/mysqld  |
| 2263 | mysql    | 20  | 0  | 347M | 39908 | 6760 | S | 0.0  | 3.8  | 0:00.11 | /usr/sbin/mysqld  |
| 2264 | mysql    | 20  | 0  | 347M | 39908 | 6760 | S | 0.0  | 3.8  | 0:00.10 | /usr/sbin/mysqld  |
| 2278 | mysql    | 20  | 0  | 347M | 39908 | 6760 | S | 0.0  | 3.8  | 0:00.26 | /usr/sbin/mysqld  |

  
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit
```

3. Otros comandos interesantes:

“\$ **uptime**”, devuelve el tiempo transcurrido desde la última vez que se arrancó el sistema, la cantidad de usuarios trabajando en el sistema (local y remotamente) y la carga del sistema (load average).

```
toor@dawl: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
toor@dawl:~$ uptime  
16:30:38 up 6 days, 22:05, 1 user, load average: 0.00, 0.00, 0.00  
toor@dawl:~$
```

“\$ **uname -a**”, información sobre el sistema operativo de nuestro servidor dedicado.

```
toor@dawl: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
toor@dawl:~$ uname -a  
Linux dawl 2.6.32-27-pve #1 SMP Tue Feb 11 16:18:29 CET 2014 i686 GNU/Linux  
toor@dawl:~$
```

“\$ **free -tm**”, información sobre la cantidad de memoria disponible y usada.

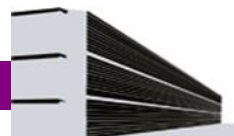
| | total | used | free | shared | buffers | cached |
|--------------------|-------|------|------|--------|---------|--------|
| Mem: | 1024 | 535 | 488 | 0 | 0 | 417 |
| -/+ buffers/cache: | | 118 | 905 | | | |
| Swap: | 1024 | 2 | 1021 | | | |
| Total: | 2048 | 538 | 1509 | | | |

“\$ **df -h**”, información sobre los sistemas de ficheros y tamaño así como los dispositivos montados en la máquina.

| | Size | Used | Avail | Use% | Mounted on |
|------------|------|------|-------|------|------------|
| Filesystem | | | | | |
| /dev/simfs | 5.0G | 1.1G | 4.0G | 21% | / |
| tmpfs | 103M | 44K | 103M | 1% | /run |
| tmpfs | 5.0M | 0 | 5.0M | 0% | /run/lock |
| tmpfs | 410M | 0 | 410M | 0% | /run/shm |

Disponemos de más en el siguiente enlace:

<https://nandodelmal.wordpress.com/tag/comandos-para-ubuntu-server/>



Fuentes

<https://help.ubuntu.com/community/Tasksel>

Usuarios y grupos:

http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m1/administracin_de_usuarios_y_grupos.html

Convertir archivo CSV a SQL:

<http://www.convertcsv.com/csv-to-sql.htm>

Creación de scripts MySQL:

<http://linuxconfig.org/bash-scripting-tutorial>

<http://stackoverflow.com/questions/8055694/how-to-execute-a-mysql-command-from-a-shell-script>

Autenticación por MySQL (APACHE):

<http://www.linuxuserexpo.com/autenticacion-con-mysql-en-apache2/>

Uso de CRONTAB (para copias de seguridad):

<http://www.sololinux.es/post/Tareas-cron-usando-crontab>

<http://voragine.net/linux/crontab-usuarios-sistema-tareas-periodicas-cron-linux>

Error Document Apache:

<https://www.addedbytes.com/articles/for-beginners/error-documents-for-beginners/>

<https://www.addedbytes.com/articles/for-beginners/http-status-codes/>

Actualizar PhpMyAdmin:

<http://www.2daygeek.com/upgrade-phpmyadmin-on-ubuntu-centos-debian-fedora-mint-rhel-opensuse/#>

<http://villatux.blogspot.com.es/2013/09/error-configuration-file-now-needs.html>

Actualizar php 5.4 a 5.5+:

<http://stackoverflow.com/questions/27413509/debian-wheezy-upgrade-php-5-4-to-5-5>