



MINISTERIO
DE EDUCACIÓN
Y CIENCIA

SECRETARÍA GENERAL
DE EDUCACIÓN
Y FORMACIÓN PROFESIONAL

DIRECCIÓN GENERAL
DE EDUCACIÓN,
FORMACIÓN PROFESIONAL
E INNOVACIÓN EDUCATIVA

CENTRO NACIONAL
DE INFORMACIÓN Y
COMUNICACIÓN EDUCATIVA

Redes de área local Aplicaciones y Servicios Linux

Otros servicios



C/ TORRELAGUNA, 58
28027 - MADRID

Índice de contenido

Instalación del servidor de shell seguro - SSH.....	3
Introducción.....	3
Instalación del servidor y el cliente ssh.....	3
Arranque y parada manual del servidor ssh.....	3
Arranque automático del servidor ssh al iniciar el sistema.....	3
Conexión al servidor mediante ssh.....	3
Servicios adicionales.....	4
Ejecución remota de aplicaciones gráficas.....	4
Servidor de ftp seguro.....	6
Copia remota de archivos.....	6
Identificación por certificado.....	7
Crear un certificado en el PC cliente.....	7
Copiar el certificado en el PC servidor.....	8
Instalación y configuración de PHP.....	8
Introducción.....	8
Instalación de PHP.....	8
Configuración de PHP.....	9
Probando PHP.....	9
Instalación y configuración de MySQL.....	9
Introducción.....	9
Instalación de mysql.....	10
Arranque y parada del SGBD mysql.....	10
Arranque automático del servidor MySQL al iniciar el sistema.....	11
Configuración del SGBD mysql.....	11
Administración del SGBD mysql.....	11
Administración del SGBD mysql vía web.....	12
Instalación y configuración de phpmyadmin.....	12
Introducción.....	12
Instalación de phpmyadmin.....	12
Configuración de phpmyadmin.....	14
host.....	15
port.....	15
auth_type.....	15
user y password.....	15
Práctica: Mis Favoritos on line.....	15
Paso 1: Creación de la base de datos.....	16
Paso 2: Creación de la tabla.....	16
Paso 3: Asignación de permisos a un usuario.....	16
Paso 4: Creación del script.....	17
Paso 5: Prueba del script.....	22
Instalación y configuración de proftpd.....	22
Introducción.....	22
Instalación de proftpd.....	22
Configuración de proftpd.....	23
Práctica - Probar el servidor ftp.....	24

Instalación del servidor de shell seguro - SSH

Introducción

El servidor de shell seguro o SSH (Secure SHell) es un servicio muy similar al servicio telnet ya que permite que un usuario acceda de forma remota a un sistema Linux pero con la particularidad de que, al contrario que telnet, las comunicaciones entre el cliente y servidor viajan encriptadas desde el primer momento de forma que si un usuario malintencionado intercepta los paquetes de datos entre el cliente y el servidor, será muy difícil que pueda extraer la información ya que se utilizan sofisticados algoritmos de encriptación.

La popularidad de ssh ha llegado a tal punto que el servicio telnet prácticamente no se utiliza. Se recomienda no utilizar nunca telnet y utilizar ssh en su lugar.

Para que un usuario se conecte a un sistema mediante ssh, deberá disponer de un cliente ssh. Desde la primera conexión, y mediante encriptación asimétrica, las comunicaciones se encriptan incluido el proceso de autenticación del usuario cuando proporciona su nombre y su contraseña. También se proporciona una clave de encriptación simétrica para encriptar las comunicaciones del resto de la sesión mediante encriptación simétrica por su menor necesidad de procesamiento.

Instalación del servidor y el cliente ssh

Para instalar el servidor y el cliente ssh debemos instalar mediante apt-get el paquete ssh que contiene tanto la aplicación servidora como la aplicación cliente:

```
// Instalación de servidor ssh y cliente ssh
root@cnice-desktop:~# apt-get install ssh
```

Los archivos de configuración son:

- /etc/ssh/ssh_config: Archivo de configuración del cliente ssh
- /etc/ssh/sshd_config: Archivo de configuración del servidor ssh

Arranque y parada manual del servidor ssh

El servidor ssh, al igual que todos los servicios en Debian, dispone de un script de arranque y parada en la carpeta /etc/init.d.

```
// Iniciar o Reiniciar el servidor ssh
root@cnice-desktop:~# /etc/init.d/ssh restart
```

```
// Parar el servidor ssh
root@cnice-desktop:~# /etc/init.d/ssh stop
```

Arranque automático del servidor ssh al iniciar el sistema.

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado [Arranque automático de servicios al iniciar el sistema.](#)

Conexión al servidor mediante ssh

Para conectar desde un PC cliente al servidor mediante ssh, debemos ejecutar el comando ssh seguido del nombre ó dirección IP del servidor. La conexión se realizará con el mismo nombre de usuario que estemos

utilizando en el PC cliente. Ejemplo, supongamos que jessica, desde el PC llamado aula5pc3, quiere conectarse al servidor cuya IP es 192.168.1.239:

```
// Conexión por ssh
jessica@aula5pc3:~$ ssh 192.168.1.239

The authenticity of host '192.168.1.239 (192.168.1.239)' can't be
established.

RSA key fingerprint is
51:70:3f:9c:ac:49:52:74:88:f5:45:a6:ae:f0:9c:8a.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '192.168.1.239' (RSA) to the list of known
hosts.

Password: // Introducir contraseña de jessica

jessica@cnice-desktop:~$ // Ya estamos en el servidor
```

La primera vez que se conecte alguien desde dicho PC cliente, se instalará el certificado de autenticación del servidor, lo cual es normal si se trata de la primera vez. A la pregunta 'Are you sure you want to continue connecting (yes/no)?' debemos responder 'yes' ya que de lo contrario la comunicación se cortará. Si ya nos hemos conectado anteriormente otras veces y vuelve a realizar ésta pregunta, significa que alguien se está haciendo pasar por el servidor (nuestro servidor ha sido hackeado) o que se ha reconfigurado el servidor (cambio de nombre, IP, etc...)

Si deseamos conectarnos al servidor utilizando un nombre de usuario diferente, debemos incluir el nombre de usuario antes del nombre o IP del servidor y separado por una arroba '@'. Ejemplo, supongamos que jessica, desde el PC llamado aula5pc3, quiere conectarse como miguel al servidor cuya IP es 192.168.1.239:

```
// Conexión por ssh como otro usuario
jessica@cliente:~$ ssh miguel@192.168.1.239

Password: // Introducir contraseña de miguel en el servidor

miguel@servidor:~$ // Ya estamos en el servidor como miguel
```

Desde PCs con Windows es posible conectarse por ssh a servidores Linux mediante el programa **Putty**. Se trata de un cliente ssh para Windows que permite acceder en modo texto al sistema Linux desde sistemas Windows.

Servicios adicionales

El paquete ssh no solamente nos proporciona conexión remota sino que proporciona otros servicios como:

Ejecución remota de aplicaciones gráficas

Mediante ssh existe la posibilidad de ejecutar aplicaciones gráficas en el servidor y manejarlas y visualizarlas en el cliente. El servidor ssh deberá tener activada la redirección del protocolo X, es decir, deberá tener el siguiente parámetro en el archivo de configuración /etc/ssh/ssh_config:

```
// Habilitar la redirección X en /etc/ssh/sshd_config
X11Forwarding yes
```

Ejemplo: supongamos que en nuestro terminal tenemos Damn Small Linux (que no dispone del gimp) y deseamos conectarnos a otro PC que sí que tiene instalado el editor gráfico gimp, los pasos que haremos serán:

```
// Ejecutar aplicaciones gráficas
jessica@cliente:~$ ssh -X cnice@192.168.1.239 // -X para redirigir
Xwindows.

cnice@cnice-desktop:~$ gimp // Ejecutamos el gimp
```

El resultado será que desde nuestro Linux sin gimp estamos manejando el gimp que se está ejecutando en el PC remoto:

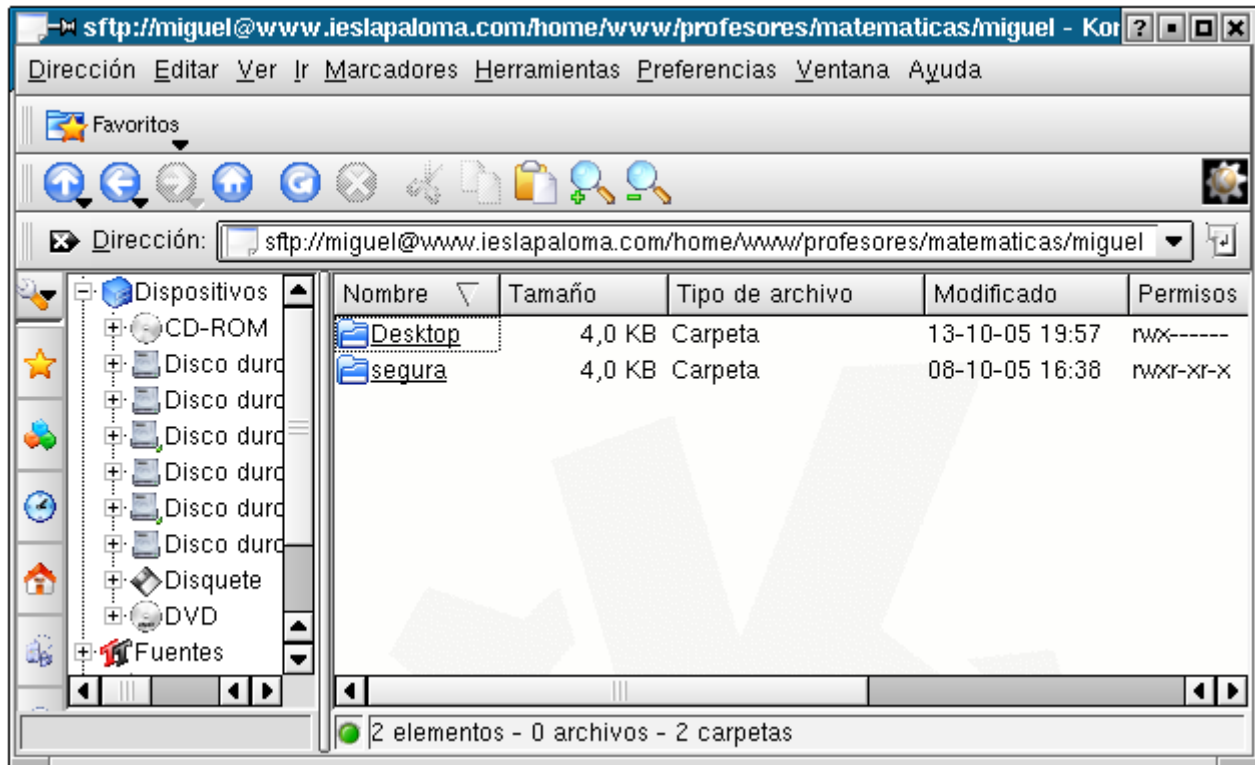


Desde PCs con Windows es posible conectarse por ssh a servidores Linux de forma gráfica mediante **Cygwin**. Se trata de un conjunto de programas libres que simulan un 'Unix para Windows' con servidor gráfico X y cliente ssh para Windows entre otras cosas, que permite acceder en modo gráfico al sistema Linux desde sistemas Windows. Otros servidores X gratuitos para Windows son Xming y Mocha.

Servidor de ftp seguro

El paquete ssh también incorpora un servidor ftp seguro y un cliente ftp seguro. Para activar el servidor ftp seguro tan solo hay que tener arrancado el servidor ssh.

El cliente ftp seguro es el comando sftp que funciona igual que el comando ftp. También podemos utilizarlo desde un navegador como Konqueror escribiendo sftp://nombre-del-usuario@nombre-del-servidor por ejemplo en la url: sftp://miguel@www.ieslapaloma.com



Copia remota de archivos

También se dispone de el comando scp que permite copiar archivos desde y hacia el servidor remoto desde el cliente. Ejemplo, si deseamos copiar el archivo /etc/hosts del servidor cuya IP es 192.168.1.239 e identificándonos como jessica en la carpeta actual de nuestro PC, ejecutaremos el siguiente comando:

```
// Copiar un archivo del servidor a nuestro PC
root@cliente:~# scp jessica@192.168.1.239:/etc/hosts .

Password: // Introducimos la contraseña de jessica en el servidor

hosts          100% 443      0.4KB/s   00:00 // Archivo
copiado

root@cliente:~#

// Copiar un archivo de nuestro PC al servidor

// La carpeta de destino debe existir en el servidor
root@cliente:~# scp miarchivo.txt
jessica@192.168.1.239:/home/jessica/pruebas/

Password: // Introducimos la contraseña de jessica en el servidor
```

```

miarchivo.txt                100%  443      1.6KB/s   00:00 //
Archivo copiado

root@cliente:~#

// Copiar una carpeta y subcarpetas de nuestro PC al servidor
root@cliente:~# scp -r /datos/*.*
jessica@192.168.1.239:/pruebas/datos/

Password: // Introducimos la contraseña de jessica en el servidor

datos/*.*                    100%  443      50.6KB/s   00:03 // Archivos
copiados

root@cliente:~#

```

Desde PCs con Windows es posible utilizar el programa **WinSCP** que permite copiar archivos desde y hacia el servidor. Se trata de un cliente que utiliza el protocolo ssh para acceder al sistema de archivos del servidor Linux desde sistemas Windows.

Identificación por certificado

Para evitar tener que introducir continuamente la contraseña cuando deseamos conectar con un servidor remoto por ssh, existe la posibilidad de autenticarse por certificado, para ello debemos:

1. Crear un certificado de usuario en el PC cliente
2. Copiar el certificado en el PC servidor

Para que el servidor ssh acepte la autenticación por medio de certificado, deberá tener activada la opción PubkeyAuthentication yes, es decir, deberá tener el siguiente parámetro en el archivo de configuración /etc/ssh/sshd_config:

```

// Permitir autenticación por certificado
PubkeyAuthentication yes

```

Crear un certificado en el PC cliente

Para crear un certificado que permita autenticar al usuario, debemos ejecutar el comando ssh-keygen. Dicho comando creará dentro de nuestra carpeta home, en una carpeta llamada '.ssh', dos archivos: uno llamado id_rsa que será la clave privada de nuestro certificado y otro llamado id_rsa.pub que será la clave pública de nuestro certificado. Éste último archivo será el que hay que copiar en el servidor remoto.

```

// Creación de un certificado
miguel@cliente:~$ ssh-keygen -t rsa

Generating public/private rsa key pair.

Enter file in which to save the key (/home/miguel/.ssh/id_rsa):

// Archivo del certificado. Podemos dejar el que viene por defecto

Created directory '/home/miguel/.ssh'.

```

```
Enter passphrase (empty for no passphrase): // Opcional

Enter same passphrase again:

Your identification has been saved in /home/miguel/.ssh/id_rsa.

Your public key has been saved in /home/miguel/.ssh/id_rsa.pub.

The key fingerprint is:

c8:a4:fe:0c:19:78:8e:7d:05:5b:13:df:37:17:e8:ea
miguel@dsl.ieslapaloma.com

miguel@dsl:~$
```

Copiar el certificado en el PC servidor

Para poder identificarse en el servidor como miguel desde el cliente, debemos copiar el archivo id_rsa.pub que hemos creado en el cliente, en la carpeta home de miguel en el servidor dentro de una carpeta llamada '.ssh' en un archivo llamado authorized_keys. Para copiar dicho archivo del cliente al servidor, podemos hacerlo con scp. Supongamos que el cliente se llama 'cliente' y el servidor se llama 'servidor':

```
// Copia del certificado y prueba de la conexión
// Nota: el símbolo ~ en Linux es la carpeta home del usuario
miguel@cliente:~$ scp ~/.ssh/id_rsa.pub
miguel@servidor:~/.ssh/authorized_keys

Password: // Va a ser la última vez que introduzcamos la contraseña

id_rsa.pub                100% 242      0.2KB/s   00:00 // Copiado

miguel@cliente:~$ ssh miguel@servidor // Probamos la conexión

miguel@servidor:~$ // Ya estamos en el servidor sin necesidad de
contraseña
```

Instalación y configuración de PHP

Introducción

PHP es, junto con mysql, el complemento ideal del servidor web apache ya que dota al servidor de un lenguaje script de ejecución en el servidor lo que facilita la creación de aplicaciones web y sitios web dinámicos.

Instalación de PHP

Para instalar PHP en nuestro servidor podemos utilizar apt-get. El paquete a instalar depende de la versión que deseemos instalar y la versión de apache. Lo normal es que utilicemos la versión 2 de apache y que instalemos la versión 5 de php. En tal caso deberíamos instalar libapache2-mod-php5:

```
// Instalación de php5 para apache 2
```



```
# apt-get install libapache2-mod-php5
```

Al instalar libapache2-mod-php5 mediante apt-get, automáticamente se configura para integrarse perfectamente en apache, creando los archivos necesarios en la carpeta de módulos disponibles de apache (/etc/apache2/mods-available) y creando los enlaces necesarios para habilitarlos en la carpeta de módulos habilitados de apache (/etc/apache2/mods-enabled)

Configuración de PHP

El archivo de configuración de php4 es el archivo:

```
// Archivo de configuración de php4
/etc/php5/apache2/php.ini
```

Los parámetros más destacables a configurar son:

- **Safe Mode = Off** (Modo Seguro. Si el Modo seguro está desactivado, se habilitan todas las funciones del php. Para un uso educativo es mejor ser funcional y no activar el modo seguro. Si el Modo seguro está activado, se deshabilitan todas las funciones del php consideradas peligrosas. Para servicios de hosting se recomienda activar el modo seguro.)
- **Display errors = On** (Mostrar Errores. Muestra los errores en las mismas páginas, cuando les haya. Cuando hay errores en los scripts, es más fácil encontrarlos si se muestran en las páginas)

Probando PHP

Una vez instalado y configurado, antes de probar debemos reiniciar el servidor web apache:

```
// Reiniciando apache
/etc/init.d/apache restart
```

Ahora crearemos una página php que utilice la función phpinfo que además de comprobar que apache y php están funcionando, nos mostrará una información de la versión. Crearemos el siguiente archivo:

```
// Probando PHP. Crear archivo /var/www/phpinfo.php - permisos 644

<HTML>

    <h1>Probando PHP</h1>
    Salida del comando phpinfo:

    <?
        phpinfo();
    ?>

</HTML>
```

Ahora tan solo necesitamos arrancar el navegador e ir a la URL: <http://ip-del-servidor/phpinfo.php>. Si nos aparece la información de la versión de php significa que está correctamente instalado.

Instalación y configuración de MySQL

Introducción

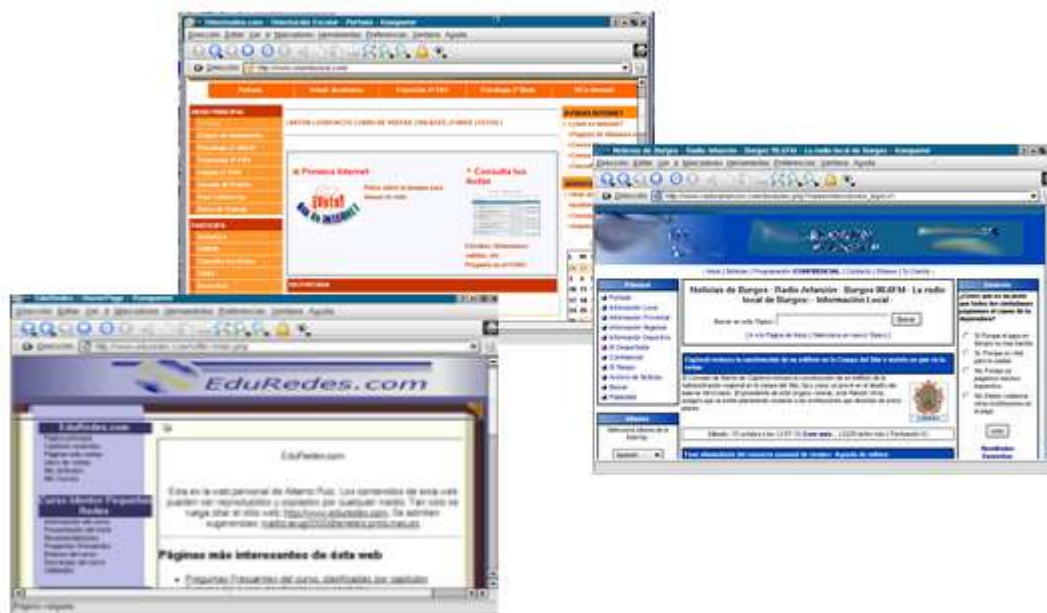
MySQL es un SGBD (Sistema Gestor de Bases de Datos) relacionales muy completo y muy utilizado tanto

en entornos Linux como en entornos Windows, principalmente para el desarrollo de aplicaciones web. Entre sus principales prestaciones destacamos:

- Fácil instalación
- Fácil administración
- Rápido
- Completo
- Multiplataforma

Por éstas razones, casi todas las aplicaciones web desarrolladas en lenguaje php que requieran de base de datos, utilizan mysql.

Si disponemos de un servidor web con soporte php y base de datos mysql, tendremos la arquitectura ideal para crear un portal dinámico utilizando gestores de contenidos como PHPNuke, drupal o Tikiwiki y herramientas orientadas a crear sitio web para entornos educativos como Mambo o Claroline, así como aplicaciones web orientadas al trabajo colaborativo y al desarrollo rápido de contenidos como Wikis y Blogs.



Instalación de mysql

Para la instalación del servidor y el cliente de mysql, debemos instalar los paquetes mysql-server, mysql-common y mysql-client mediante apt-get. Se instalará la versión 5 de mysql:

```
// Instalación de mysql
# apt-get install mysql-server mysql-common mysql-client
```

Arranque y parada del SGBD mysql

El servidor de datos mysql, al igual que todos los servicios en Debian, dispone de un script de arranque y parada en la carpeta /etc/init.d.

```
// Iniciar o reiniciar el servidor mysql
# /etc/init.d/mysql restart
```

```
// Parar el servidor mysql
# /etc/init.d/mysql stop
```

Arranque automático del servidor MySQL al iniciar el sistema.

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado [Arranque automático de servicios al iniciar el sistema](#).

Configuración del SGBD mysql

El archivo de configuración de mysql es el archivo:

```
// Archivo de configuración de mysql
/etc/mysql/my.cnf
```

En dicho archivo se configuran aspectos generales como la contraseña, el puerto a utilizar y algún otro aspecto pero para hacer un uso normal de mysql, no es necesario realizar ninguna modificación del archivo original.

Administración del SGBD mysql

Mysql es un SGBD completo que permite crear usuarios y establecer permisos sobre bases de datos, tablas y campos deseados a dichos usuarios. Los permisos pueden ser de consulta, inserción, modificación y borrado de datos, creación, modificación y eliminación de tablas y bases de datos y de administración de usuarios y permisos, lo que hace a mysql ser un SGBD muy flexible y muy completo.

Quizás la primera acción que se debería hacer nada más arrancar el SGBD mysql sería poner una contraseña al usuario root ya que inicialmente no tiene contraseña. Para ello debemos iniciar mysql con el comando:

```
// Iniciar el servidor de bases de datos mysql
# /etc/init.d/mysql start
```

Posteriormente iniciamos el cliente de mysql como root y cuando aparezca el prompt de mysql (mysql>) ejecutamos una orden grant para establecer la contraseña de root:

```
// Ejecutar cliente de mysql y cambiar contraseña de root
# mysql -u root // Accedemos sin contraseña
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 5 to server version: 4.0.20-log
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> grant all privileges on *.* to root@localhost identified by
'secreta' with grant option;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> quit
```

Bye

De ésta manera habremos puesto como contraseña de root la palabra 'secreta'. La próxima vez que entremos, debemos añadir la opción -p para que nos pida la contraseña ya que de lo contrario no nos dejará entrar:

```
// Entrando como root con contraseña  
# mysql -u root -p
```

Ahora debemos introducir la contraseña para acceder y tener acceso global al SGBD mysql.

Administración del SGBD mysql vía web

La herramienta de administración de mysql vía web es phpmyadmin. Para más información consultar el apartado [Instalacion_y_configuracion_de_PHPMyAdmin](#)

Instalación y configuracion de phpmyadmin

Introducción.

Phpmyadmin es una excelente herramienta de administración de mysql vía web. Para poder utilizar phpmyadmin se requiere disponer de un servidor web con soporte php.

La herramienta permite que cualquier usuario de la base de datos que disponga de algún permiso, acceda y haga uso de dichos permisos. Identificándose con el usuario administrador de la base de datos (root) o con cualquier otro usuario que disponga de todos los privilegios, estarán habilitadas todas las características de la herramienta.

Instalación de phpmyadmin

La instalación de phpmyadmin se puede realizar de forma automática con apt-get, pero se trata de un paquete que no se encuentra en el 'repositorio principal' (main) de ubuntu, sino que se encuentra en el 'repositorio universo' (universe). Para que apt-get pueda instalar paquetes del repositorio universo, es necesario editar el archivo /etc/apt/sources.list y quitar la almohadilla de las líneas:

```
// Quitar almohadilla delante de las líneas, para descomentarlas  
deb http://es.archive.ubuntu.com/ubuntu/ drapper universe  
deb-src http://es.archive.ubuntu.com/ubuntu/ drapper universe
```

Posteriormente, tendremos que actualizar en nuestro PC, el contenido de los repositorios:

```
// Actualizar contenido de los repositorios  
# apt-get update
```

Ahora sí, podremos instalar el paquete 'phpmyadmin' mediante el comando:

```
// Instalación de phpmyadmin  
# apt-get install phpmyadmin
```

El programa de instalación crea un enlace simbólico en el DocumentRoot del servidor web para que la aplicación pueda ser accesible desde la url: <http://ip-del-servidor-web/phpmyadmin/index.php>. Si no se viera la aplicación en dicha url, quizás sea por algún aspecto de la configuración de apache. En tal caso, lo más sencillo sería mover la carpeta de phpmyadmin directamente dentro del DocumentRoot del servidor, mediante el comando:

```
// Colocar phpmyadmin en el servidor web y asignar propietario www
# mv /usr/share/phpmyadmin /var/www/ (en nuestro caso)

# chown -R www /var/www/phpmyadmin
```

De ésta forma, es seguro que accediendo a <http://ip-del-servidor-web/phpmyadmin/index.php> nos aparecerá la página de autenticación de phpmyadmin:


phpMyAdmin 2.8.0.3-Debian-1 - Mozilla Firefox

Archivo Editar Ver Ir Marcadores Herramientas Ayuda

← → ↺ × 🏠 Ir

🔴 Getting Started 📡 Latest BBC Headlines

🌐 Explotación de Redes 📄 phpMyAdmin 2.8.0.3-De... 📄 Ubuntu -- Package Download ...


Bienvenido a phpMyAdmin 2.8.0.3-Debian-1

Language ⓘ

Iniciar sesión

Usuario:

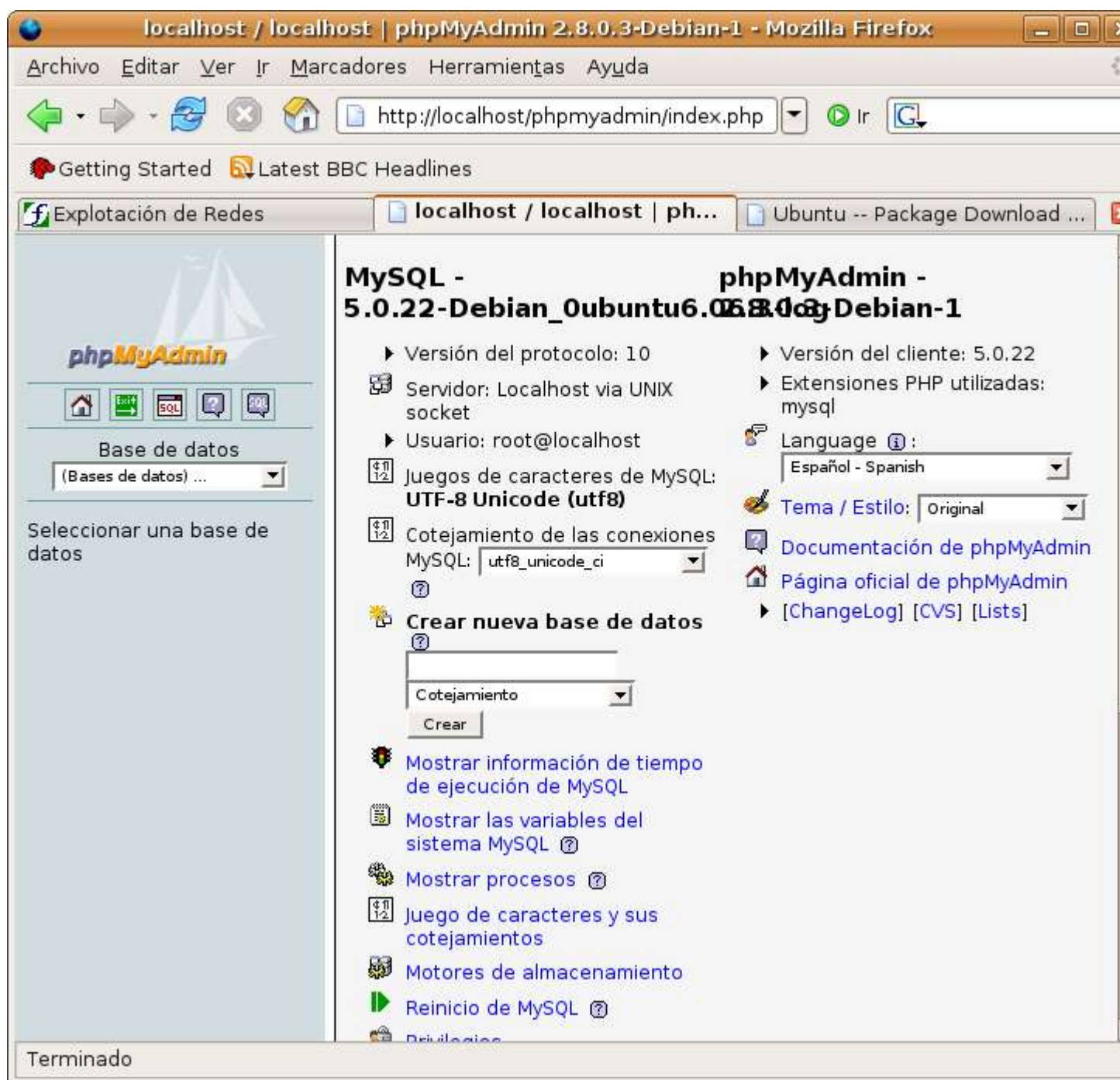
Contraseña:

📘 Las "cookies" deben estar habilitadas.

Terminado

Página de autenticación de phpmyadmin

Una vez que nos identifiquemos con un usuario y contraseña válidos, accederemos a la página principal de phpmyadmin desde la que podremos crear una nueva base de datos o realizar consultas y modificaciones sobre bases de datos ya existentes:



Página principal de phpmyadmin

Configuración de phpmyadmin

El archivo de configuración de phpmyadmin es el archivo:

```
// Archivo de configuración de phpmyadmin
/etc/phpmyadmin/config.inc.php
```

En dicho archivo de configuración hay que establecer los parámetros que permitirán a phpmyadmin conectar con mysql, que son:

host

En este parámetro habrá que indicar la IP del servidor mysql. Si el servidor web y el servidor mysql son la misma máquina, se deberá poner 'localhost' ó 127.0.0.1. En el caso de que sean máquinas diferentes, se deberá poner la IP del servidor mysql.

port

Aquí se especifica el puerto de conexión al servidor mysql. El puerto por defecto por el que sirve datos el servidor mysql es el 3306. Si en la configuración de mysql (archivo /etc/mysql/my.cnf) no se ha cambiado, no es necesario especificarlo ya que se usará el puerto 3306 por defecto.

auth_type

Para que phpmyadmin pueda acceder a mysql, es necesario autenticarse. Se admiten tres formas de autenticación:

- **config:** permite que el nombre de usuario y la contraseña se especifiquen en el archivo config.inc.php
- **http:** el usuario deberá introducir nombre y contraseña para acceder a la ruta web
- **cookie:** el usuario deberá introducir nombre y contraseña para acceder a la aplicación

user y password

En el caso de que hayamos elegido tipo de autenticación 'config', será necesario proporcionar el nombre de usuario y la contraseña con el que phpmyadmin accederá a mysql. En tal caso, la línea del password puede quedar comentada.

```
// Configuración por defecto en config.inc.php
Corresponden a las siguientes líneas en el archivo config.inc.php:

$cfg['Servers'][$i]['host']          = 'localhost'; // MySQL hostname
or IP

$cfg['Servers'][$i]['port']          = '';          // MySQL port-
blank default port

$cfg['Servers'][$i]['auth_type']     = 'cookie';    // Authentication
method

$cfg['Servers'][$i]['user']          = 'root';      // MySQL user

$cfg['Servers'][$i]['password']      = '';         // MySQL pass (only
'config')
```

Práctica: Mis Favoritos on line

En la siguiente práctica pondremos en marcha una aplicación web basada en php y mysql. Dicha aplicación la hemos bautizado como **Mis Favoritos on-line** y permitirá añadir, eliminar y visualizar mis direcciones de Internet favoritas. Los datos se almacenarán en una base de datos mysql.

Paso 1: Creación de la base de datos

Crearemos en mysql una base de datos llamada 'favoritos' y nos situaremos en ella.

```
// Crear base de datos
mysql> CREATE DATABASE favoritos;

Query OK, 1 row affected (0.00 sec)
```

```
mysql> USE favoritos;
```

```
Database changed
```

Paso 2: Creación de la tabla

Crearemos una tabla llamada 'favoritos' en la base de datos 'favoritos'.

```
// Crear tabla (ojo: usar comilla simple invertida en nombres)
mysql> CREATE TABLE `favoritos` (

  `numero` INT NOT NULL PRIMARY KEY ,

  `url` VARCHAR( 50 ) ,

  `descripcion` VARCHAR( 50 )

);

Query OK, 0 row affected (0.02 sec)


// Insertamos un registro en la tabla para que no esté vacía:
mysql> INSERT INTO `favoritos` ( `numero` , `url` , `descripcion` )

VALUES ( '1', 'http://www.pntic.mec.es', 'Web del cnice' );

Query OK, 1 row affected (0.00 sec)
```

Paso 3: Asignación de permisos a un usuario

Crearemos un usuario en mysql asignandole permisos sobre nuestra base de datos. Nuestro usuario se llamará 'cnice' y su contraseña será 'cnice'. Se podrá conectar desde el mismo equipo (localhost) y tendrá todos los privilegios sobre todas las tablas de nuestra base de datos 'favoritos':

```
// Conceder permisos
mysql> GRANT ALL ON favoritos.* to cnice@localhost identified by
"cnice";
```



```
Query OK, 0 row affected (0.01 sec)
```

```
mysql> flush privileges; // actualizar permisos
```

```
Query OK, 0 row affected (0.01 sec)
```

Paso 4: Creación del script

Crearemos un script en php que accederá a mi base de datos y permitirá al usuario insertar registros y consultar el contenido de la tabla.

```
// Script PHP para el acceso a datos MySQL
```

```
<?
```

```
////////////////////////////////////  
////////
```

```
//
```

```
//      MisFavoritos on-line. (C) 2007 - CNICE.
```

```
//      Nombre del script: index.php
```

```
//
```

```
////////////////////////////////////  
////////
```

```
// Parámetros de conexión con la base de datos
```

```
define( "DB_HOST",      "localhost" );
```

```
define( "DB_USER",      "cnice" );
```

```
define( "DB_PASSWD",    "cnice" );
```

```
define( "DB_NAME",      "favoritos" );
```

```
error_reporting( 0 ); //Para que no muestre warnings ni errores
```

```
?>
```

```

<HTML>

<HEAD><TITLE> Favoritos - mysql </TITLE></HEAD>

<H1>Favoritos</H1>

<STYLE type="text/css">

<!--A {font-family: Arial; color: #00FF00}-->

</STYLE></HEAD>

<BODY>

Favoritos on-line. Acceso a datos mysql desde páginas PHP.<HR>

Elija la operación que desee efectuar:<BR>

<TABLE BORDER>

    <TD><A HREF="index.php?Accion=Ayuda">Ayuda</A></TD>

    <TD><A HREF="index.php?Accion=Nuevo">Nuevo</A></TD>

    <TD><A HREF="index.php?Accion=Borrar">Borrar</A></TD>

</TABLE>

<HR>

<?

$IdCon = mysql_connect( DB_HOST, DB_USER, DB_PASSWD ) or die( "Error
en la conexión: " . mysql_error());

mysql_select_db( DB_NAME, $IdCon );

echo "<TABLE BORDER>";

echo "<TR><TD>Número</TD><TD>URL</TD><TD>Descripción</TD></TR>";

//Mostramos el contenido de la tabla

$cSql = "SELECT * FROM favoritos";

$IdQry = mysql_query( $cSql, $IdCon );

while ( $IdRec = mysql_fetch_array( $IdQry ) ) {

printf('<TR><TD>%s</TD><TD><A
HREF="%s">%s</A></TD><TD>%s</TD></TR>', $IdRec[0], $IdRec[1],

```

```

$IdRec[1], $IdRec[2]);

}

echo "</TABLE>";


$Accion= $_GET['Accion'];

$numero=$_GET['numero'];

$url=$_GET['url'];

$descripcion=$_GET['descripcion'];


switch ($Accion)
{
//----- Visualizar Ayuda -----

    case ('Ayuda'):

        echo "<HR>Las acciones disponibles son: <BR> <BR>";

        echo "<TABLE BORDER>";

        echo "<TR><TD>Ayuda: Muestra éste mensaje de ayuda
</TD></TR>";

        echo "<TR><TD>Nuevo: Crea un nuevo registro </TD></TR>";

        echo "<TR><TD>Borrar: Elimina un registro </TD></TR>";

        echo "</TABLE>";

        echo "<HR>";

        break;

//----- Nuevo -----

    case ('Nuevo'):

        //Si no enviamos una url, pintamos el formulario

        if (isset($url)==FALSE){

```

```

        echo '<FORM METHOD="GET" ACTION="index.php">';

        echo '<INPUT TYPE=HIDDEN NAME="Accion"
VALUE="Nuevo">';

        echo 'Num.: <INPUT TYPE="text" NAME="numero"><BR>';

        echo 'URL: <INPUT TYPE="text" NAME="url"><BR>';

        echo 'Descripción: <INPUT TYPE="text"
NAME="descripcion">';

        echo '<INPUT TYPE="Submit" VALUE="Insertar
Registro">';

        echo '</FORM>';

    }

    else{

        $cSql = "INSERT INTO favoritos values
($numero,'$url','$descripcion')";

        $idQry = mysql_query( $cSql, $idCon );

        if ($idQry==FALSE){

            echo "Error al añadir un registro";

        }

        else{

            echo "<BR>Registro $numero, $url, $descripcion
añadido satisfactoriamente.";

            echo 'Clic <a href="index.php">aquí</a> para
refrescar.';

        }

    }

    break;

//----- Borrar -----

case ('Borrar'):

    //Si no enviamos un NumReg, pintamos el formulario

```

```

        if (isset($numero)==FALSE){

            echo '<FORM METHOD="GET" ACTION="index.php">';

            echo '<INPUT TYPE=HIDDEN NAME="Accion"
VALUE="Borrar">';

            echo 'Introduzca Número de Registro a eliminar: <INPUT
TYPE="text" NAME="numero">';

            echo '<INPUT TYPE="Submit" VALUE="Aceptar">';

            echo '</FORM>';

        }

        else{

            //Primero comprobamos si existe dicho registro

            $cSql = "SELECT * FROM favoritos WHERE numero =
$numero";

            $idQry = mysql_query( $cSql, $idCon );

            $nfilas = mysql_num_rows($idQry);

            echo "Filas=$nfilas";

            //Si no existe, no podemos borrarlo

            if ($nfilas==0){

                echo "No existe el registro $numero. Imposible
eliminar dicho registro.";

            }

            else{

                $cSql = "DELETE FROM favoritos WHERE numero =
$numero";

                $idQry = mysql_query( $cSql, $idCon );

                if ($idQry==FALSE) echo "Error al eliminar el
registro $numero";

                else{

                    echo "Registro $numero eliminado
satisfactoriamente.";

                    echo 'Clic <a
href="index.php">aquí</a> para refrescar.';

```

```

        }

    }

    break;

}

?>

</BODY>

</HTML>

```

Paso 5: Prueba del script

Una vez creado el script, debemos subirlo a nuestro servidor dentro del 'Raíz de documentos' del servidor web y acceder desde el navegador. Ejemplo, si dentro de nuestra carpeta /var/www hemos creado una carpeta llamada 'favoritos' y hemos subido el script con el nombre 'index.php', para probarlo debemos poner en el navegador: <http://ip-del-servidor/favoritos/index.php>

Instalación y configuración de proftpd

Introducción

Proftpd es un servidor de ftp rápido, de fácil instalación y flexible configuración con un esquema similar a la configuración de apache. Además permite diferentes posibilidades de autenticación: mediante usuarios del sistema unix, mediante base de datos mysql o autenticación mediante servidor LDAP.

Instalación de proftpd

Proftpd se puede instalar automáticamente mediante apt-get:

```
// Instalación de proftpd
# apt-get install proftpd
```

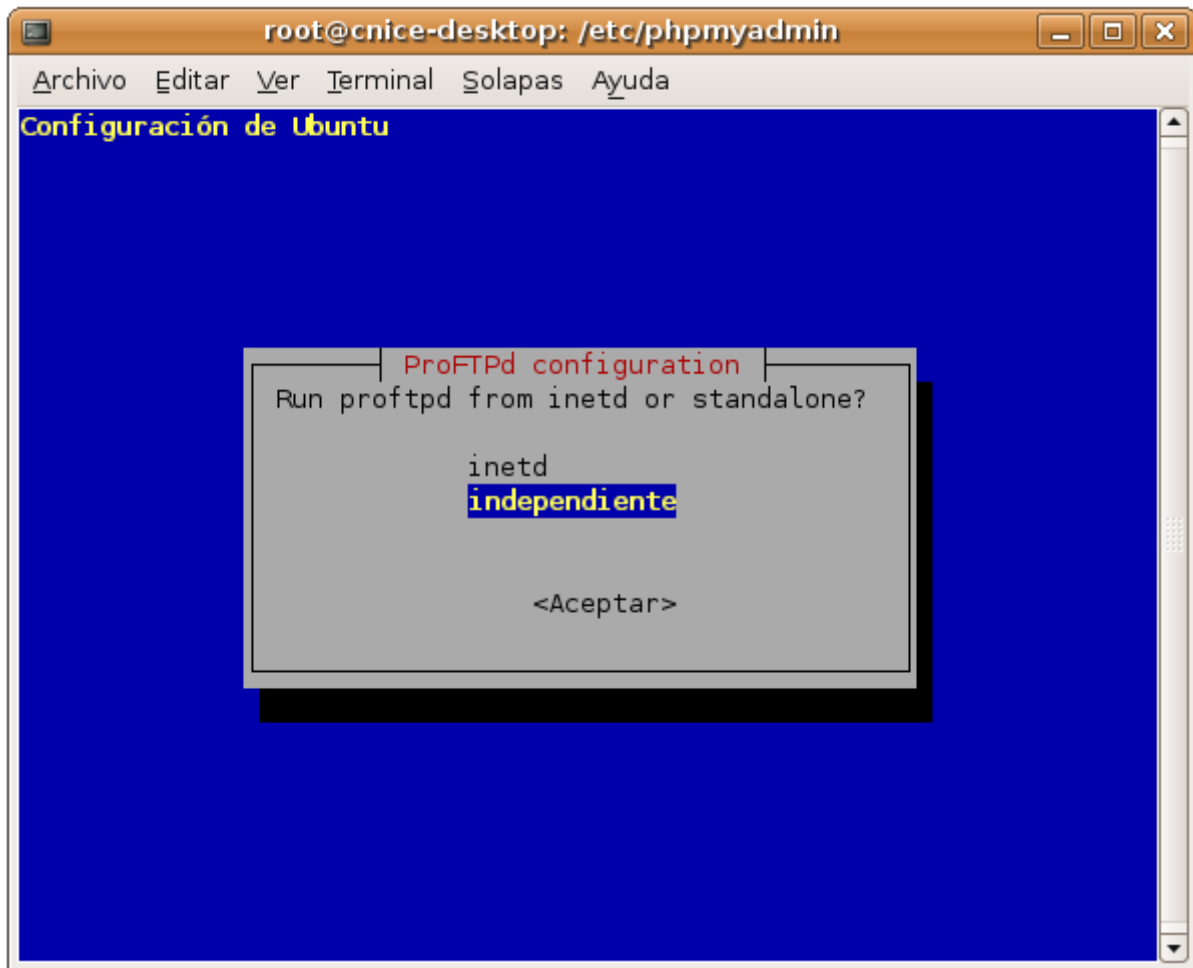
Si disponemos de un servidor LDAP, podemos instalar la versión apta para ldap 'proftpd-ldap'.

Configuración de proftpd

Al instalar el paquete proftpd-ldap se iniciará el asistente de configuración de proftpd. Si más adelante deseamos lanzar de nuevo el asistente, debemos ejecutar:

```
// Lanzar el asistente de configuración de proftpd
# dpkg-reconfigure proftpd
```

Este asistente únicamente nos hace una pregunta que es si deseamos ejecutar el servidor desde inetd (solo se carga en memoria cuando existan peticiones) o como un servicio independiente (permanentemente en memoria). El funcionamiento como servicio independiente es más eficiente.



Para que proftpd arranque automáticamente al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado [Arranque automático de servicios al iniciar el sistema](#).

El archivo de configuración de proftpd es el archivo:

```
// Archivo de configuración de proftpd
/etc/proftpd.conf
```

No es necesario modificar ningún parámetro del archivo /etc/proftpd.conf para un uso normal del servidor ftp en el centro educativo.

Si hemos instalado proftpd-ldap porque en la red disponemos de un servidor LDAP con las cuentas de usuarios de nuestro sistema, para que proftpd autentique a los usuarios contra nuestro servidor LDAP es necesario configurar tres parámetros: quien es el servidor LDAP (LDAPServer), cual es el usuario

administrador y la contraseña de LDAP (LDAPDNInfo) y qué unidad organizativa tiene la información de los usuarios (LDAPDoAuth). También configuraremos como máscara de creación de archivos y carpetas la máscara 002 porque utilizamos grupos privados de usuario:

```
// Parámetros destacables a configurar
# Para que autentifique contra nuestro servidor LDAP

AuthPAM on

LDAPServer localhost

LDAPDNInfo cn=admin,dc=ieslapaloma,dc=com ldapadmin // ldapadmin =
contraseña

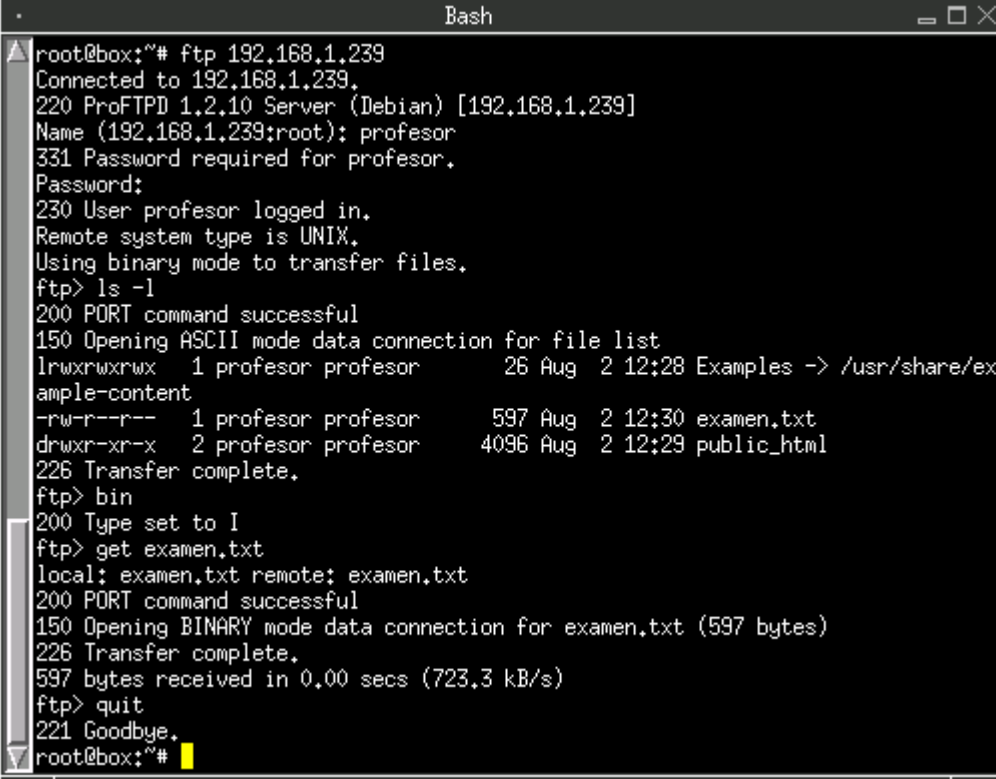
LDAPDoAuth on "ou=users,dc=ieslapaloma,dc=com"

# Permisos recomendados para UPG (grupos privados de usuario)

Umask                                002  002
```

Práctica - Probar el servidor ftp

Para probar que está funcionando el servidor ftp intentaremos entrar con el usuario profesor y una vez dentro descargaremos un archivo desde el servidor ftp a nuestro PC:



```
Bash
root@box:~# ftp 192.168.1.239
Connected to 192.168.1.239.
220 ProFTPD 1.2.10 Server (Debian) [192.168.1.239]
Name (192.168.1.239:root): profesor
331 Password required for profesor.
Password:
230 User profesor logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -l
200 PORT command successful
150 Opening ASCII mode data connection for file list
lrwxrwxrwx  1 profesor profesor    26 Aug  2 12:28 Examples -> /usr/share/ex
ample-content
-rw-r--r--  1 profesor profesor   597 Aug  2 12:30 examen.txt
drwxr-xr-x  2 profesor profesor  4096 Aug  2 12:29 public_html
226 Transfer complete.
ftp> bin
200 Type set to I
ftp> get examen.txt
local: examen.txt remote: examen.txt
200 PORT command successful
150 Opening BINARY mode data connection for examen.txt (597 bytes)
226 Transfer complete.
597 bytes received in 0.00 secs (723.3 kB/s)
ftp> quit
221 Goodbye.
root@box:~#
```