

Puesta en marcha de un servidor FTP



Orlando Alemán Ortiz
Samuel Díaz Cabrera

4º Ing. Informática
Curso 2005/06

Licencia



Esta obra ha sido publicada bajo licencia "Reconocimiento-NoComercial-CompartirIgual 2.5 Spain" de Creative Commons, la cual implica que:

Usted es libre de:

- copiar, distribuir y comunicar públicamente la obra
- hacer obras derivadas

Bajo las condiciones siguientes:

Reconocimiento. Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador.

No comercial. No puede utilizar esta obra para fines comerciales.



Compartir bajo la misma licencia. Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.



Y además:

- Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.
- Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor
- Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.

Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-sa/2.5/es/> o envíe una carta a Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Índice de contenidos

1. Previos.....	2
Introducción.....	2
Equipamiento empleado.....	2
2. Desarrollo.....	3
Instalación del software.....	3
Configuración de Xinetd.....	4
Acceso de usuarios anónimos.....	7
Opciones.....	8
Posibilidad de acceso en modo de usuario del sistema.....	11
Opciones.....	11
Posibilidad de acceso como usuarios virtuales.....	12
3. Referencias.....	16

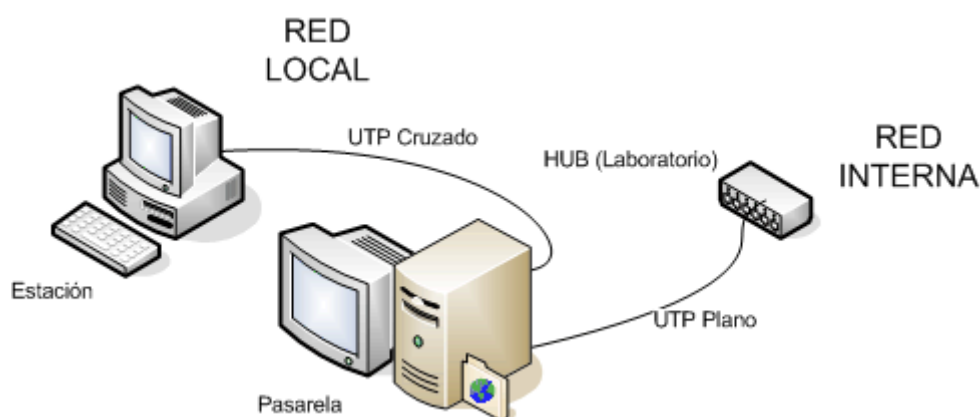
1. Previos

Introducción

En esta tercera práctica montaremos un servidor FTP de forma rápida y sencilla. Lo haremos teniendo en cuenta distintos perfiles de funcionamiento que pueden darse. Esto implicará el establecimiento de una serie de parámetros en tiempo de compilación y en la configuración posterior. Como ejemplo de estos ajustes tendríamos fijar una cantidad máxima de usuarios conectados simultáneamente o establecer cuentas de usuario.

Como viene siendo habitual, implementaremos la solución sobre *GNU Linux*. La aplicación escogida para esta ocasión es *PureFTP*, de libre acceso y distribución, y conocida por su potencialidad y seguridad.

Llevaremos a cabo la instalación en la pasarela, si bien realizarla en la estación no difiere en ningún aspecto el procedimiento a seguir. El esquema de nuestra red local es, como siempre, el de la figura siguiente.



Equipamiento empleado

Hardware:

- 1 x PC con 2 tarjetas de red, que actuará como pasarela.
- 1 x PC con 1 tarjeta de red, que actuará de estación.
- 1 x cable de red UTP Cat.5 plano con conectores RJ45.
- 1 x cable de red UTP Cat.5 cruzado y con conectores RJ45.

Software:

- Por cada PC, una instalación de *Fedora Linux*.
- En las instalaciones de *Fedora*, el paquete *PureFTP*.

2. Desarrollo

A continuación expondremos la resolución de la problemática de montar un servidor FTP, atendiendo eso sí, a los distintos puntos propuestos en el guión.

Lo primero que debemos hacer es obtener la última versión estable desde la página oficial del proyecto e instalarlo. Luego integraremos la aplicación en el sistema, configurando el superdemonio xinetd.

“xinetd” es el demonio para los servicios de Internet que emplea Fedora Linux. En lugar de tener muchos servidores distintos corriendo a la vez, sólo se carga xinetd. En el momento en que se hace una petición de servicio, xinetd lanza el servidor correspondiente.

Instalación del software

Una novedad importante con respecto a prácticas anteriores, es que en esta ocasión vamos a instalar el software compilando los fuentes, no instalando un binario. Esto nos permitirá tener un servidor menos pesado, que disponga sólo de los módulos que necesitemos.

Procedemos a descargar las fuentes desde www.pureftp.org. En el momento de realizar esta práctica, la versión actual es la 1.0.21. Así que obtendremos el correspondiente paquete desde el repositorio oficial:

```
$ wget http://download.pureftpd.org/pub/pure-ftpd/releases/pure-ftpd-1.0.21.tar.gz
```

Situados en el directorio donde se ha descargado el paquete, descomprimos y extraemos el tarball:

```
$ tar -xvzf pure-ftpd-1.0.21.tar.gz
```

Como resultado se habrá generado un directorio con las fuentes y la documentación de apoyo, todo listo para realizar la instalación. Accedemos a él:

```
$ cd pure-ftpd-1.0.21/
```

Teniendo claro nuestro objetivo, nos fijaremos únicamente en los ficheros que nos interesan.

El script 'configure'. Permite establecer las opciones de compilación que deseemos incluir, de acuerdo a las posibilidades explicadas en el fichero README y construir un “Makefile” de acuerdo a las características del sistema: librerías, rutas, máquina, etc.

El fichero 'Makefile'. Es generado por 'configure' y permite compilar adecuadamente la aplicación y llevar a cabo la instalación y desinstalación de la misma, de acuerdo a las especificaciones recogidas en README.

El fichero 'README', citado ya varias veces, contiene toda la documentación que el fabricante considera oportuna para poder utilizar sin problemas la aplicación.

Práctica 3: Puesta en marcha de un servidor FTP

Teniendo en cuenta estos detalles, procedemos a la instalación. Compilaremos *PureFTP* con los siguientes parámetros:

- Generación de un fichero de logging adicional (ademas de syslog): `--with-altlog`
- Posibilidad de mostrar un mensaje de fortuna o un banner personalizado cuando los usuarios se loguean: `--with-cookie`
- Lenguaje español: `--with-language=spanish`
- Soporte de usuarios virtuales: `--with-puredb`
- Posibilidad de limitar el ancho de banda: `--with-throttling`
- Límite de usuarios que pueden existir en el servidor: `--with-peruserlimits`
- Si solo se desea actuar en modo Super-Server: `--with-standalone`

Es decir,

```
$ ./configure --with-language=spanish --with-altlog --with-cookie --with-peruserlimits \  
--with-throttling --with-puredb --with-standalone
```

Para compilar e instalar el servidor en nuestro sistema:

```
$ make install-strip
```

Una vez finalizado el proceso de instalación podemos lanzar el ejecutable creado en el directorio `"/usr/local/sbin"` con:

```
$ /usr/local/sbin/pure-ftpd &
```

La aplicación puede ser desinstalada mediante la ejecución del comando:

```
$ make uninstall
```

A continuación configuraremos el superdemonio `xinetd` para configurar la ejecución de *"PureFTP"* bajo demanda y se ajuste a nuestras necesidades.

Configuración de Xinetd

El superdemonio `xinetd` utiliza el fichero de configuración `"xinetd.conf"` localizado en el directorio `"/etc/"`, el cual contiene la configuración básica con las opciones generales con las que funcionará `xinetd`. En él se incluye la directiva `"includedir /etc/xinetd.d"` que especifica un directorio donde se encuentran el conjunto de ficheros específicos para los servicios que gestiona `xinetd`. En ellos se detalla la configuración utilizada para el arranque de los mismos.

Práctica 3: Puesta en marcha de un servidor FTP

```
#
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/

defaults
{
    instances                = 60
    log_type                  = SYSLOG authpriv
    log_on_success            = HOST PID
    log_on_failure            = HOST
    cps                       = 25 30
}

includedir /etc/xinetd.d
```

Para la gestión del servicio ftp se ha creado el archivo “*pure-ftpd*” con la siguiente configuración:

```
service ftp
{
    protocol      = tcp
    flags         = REUSE
    socket_type   = stream
    wait         = no
    user         = root
    server        = /usr/local/sbin/pure-ftpd
    server_args   = -j -l puredb:/etc/pureftpd.pdb
    log_on_failure
    # acceso remoto unicamente
    # no_access   = localhost
    disable      = no
}
```

Estos archivos usan las mismas convenciones, están estructurados de forma que se definan parámetros para globales y parámetros para los distintos servicios. Encapsulados todos por { } y encabezados por una directiva que especifica si los parámetros contenidos son globales (“*default*”) o destinados a los servicios (“*service ftp*”).

En la figura anterior, se utilizan las siguientes cláusulas:

- `protocol` .- Determina el protocolo que es empleado por el servicio. El protocolo debe existir en “*/etc/protocols*”.
- `flags` .- Configura cualquier número de atributos para la conexión.
- `socket_type` .- Configura el socket de red para escribir a stream.
- `wait` .- Define si el servicio es de un sólo hilo (yes) o de múltiples hilos (no).
- `user` .- Define bajo qué ID de usuario el proceso se ejecutará.
- `server` .- Define el binario ejecutable a lanzar

Práctica 3: Puesta en marcha de un servidor FTP

- `server_args` .- Define el conjunto de argumentos que se le suministrarán al servidor.
- `log_on failure` .- Define los parámetros de registro para `log_on_failure` además de aquellos ya definidos en `xinetd.conf`
- `no_access` .- Limita quien tiene acceso al servidor.
- `disable` .- Define si el servicio está activo o no.

Con la línea “`no_access = 127.0.0.0`” especificamos que solamente se pueda acceder remotamente al ftp.

Para que se carguen este conjunto de ficheros el servicio `xinetd` debe ser reiniciado:

```
$ service xinetd restart
```

o usando:

```
$ killall -USR2 xinetd
```

Para ver que un demonio está activo, es decir, que escucha peticiones, ejecutamos:

```
$ netstat -atu | grep '*:*' | more
```

Comprobamos que el demonio `pure-ftpd` funciona correctamente bajo demanda:

```
[Terminal:1] .- En el terminal 1 ejecutamos:
$ ftp localhost
[Terminal:2] .- En el terminal 2 observamos que se ha activado un servicio
$ ps auxw | grep pure-ftpd
root      3551  0.0  0.1 4196  860 ?        Ss   12:33   0:00 pure-ftpd
(IDLE)
```

Otra manera de cargar las distintas configuraciones es especificarlas en un fichero de configuración como “`/etc/pure-ftpd.conf`”. Primero debemos copiar la plantilla de fichero `.conf` que se nos proporciona con `pure-ftpd`

```
$ cp configuration-files/pure-ftpd.conf /etc
```

Esta plantilla la editamos a nuestras necesidades. Copiamos el fichero “`pure-config.pl`” a nuestro directorio de binarios y le aplicamos permisos de ejecución.

```
$ chmod 755 pure-config.pl
```

Para ejecutar el servidor, pondríamos en el fichero `/etc/xinetd.d/pure-ftpd`

```
server = [dir_binarios]/pure-config.pl /etc/pure-ftpd.conf
```

Por último, la posibilidad de añadir la opción en la línea:

```
server_args =
```

A continuación explicaremos tanto la configuración de los ficheros anteriores como los

Práctica 3: Puesta en marcha de un servidor FTP

posibles argumentos que se le pueden suministrar a ftpd para cumplir los puntos solicitados en la práctica.

Acceso de usuarios anónimos

En general cuando un usuario se conecta al servicio FTP se sitúa en su directorio personal. Para los usuarios anónimos será necesario crear la carpeta que les corresponde, estos usuarios entrarán al sistema mediante el servicio ftp, para obtener la ruta por defecto del directorio al que pertenece este “usuario” utilizamos el siguiente comando:

```
$ cat /etc/passwd | grep ftp
```

```
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
```

Obtendremos como respuesta la línea donde se nos especifica que el directorio personal será “/var/ftp”, el cual no existe y debemos crear para permitir el acceso de los usuarios anónimos. Si el usuario 'ftp' existe y su directorio es alcanzable (se ha creado), Pure-FTPd aceptará conexión anónima, usando como usuario 'ftp' o 'anonymous'. Si no existiese, habría que crearlo. Para ello se dispone del comando 'useradd'. La sintaxis sería similar a la siguiente:

```
$ useradd -d /var/ftp -n ftp
```

Además para que estos usuarios puedan subir y descargar archivos debemos asignarles permisos a la carpeta:

```
$ mkdir /var/ftp  
$ chown -R ftp /var/ftp  
$ chown -R 700 /var/ftp
```

Para permitir exclusivamente el acceso de los usuarios anónimos debemos ejecutar PureFTPd con el argumento “-e”.

Comprobamos que ya podemos conectarnos al servicio y enviar archivos a la ruta permitida correspondiente como usuario anónimo:

```
$ ftp localhost
```

Práctica 3: Puesta en marcha de un servidor FTP

```
Name (localhost:root): ftp
 230 Usuario Anónimo dentro del sistema
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> send /root/ejemplo ejemplo
local: /root/ejemplo remote: ejemplo
227 Entering Passive Mode (127,0,0,1,189,69)
150 Conexin de datos aceptada
26-Archivo transmitido con éxito
26 0.032 segundos (medidos aquí), 8.66 Mbytes por segundo
287744 bytes sent in 0.031 seconds (9e+03 Kbytes/s)
ftp> quit
```

Opciones

Presentación de un mensaje de bienvenida cuando un usuario acceda al ftp.

Una de las opciones habituales que suelen incorporar los servidores ftp es la posibilidad de mostrar un mensaje personal a la entrada del servidor que, por ejemplo, lo identifique o identifique a la empresa que lo mantiene.

Se disponen de varias maneras para mostrar un mensaje de bienvenida a los usuarios anónimos cuando se conectan.

Forma 1: Personal

Crear un fichero de texto '.banner' en el directorio principal del usuario 'ftp' (límite 4000 caracteres ASCII).

Forma 2: Global

Compilar el servidor con la opción '--with-cookie' y pasarle al demonio el argumento '-F' seguida del nombre del fichero del mensaje. En la próxima conexión ya podremos visualizarlo. Ejemplo:

```
230-----
230-- Bienvenido al FTP Pasarela --
230-- Frase del día: Que bonitas son las practicas de redes --
230-----
```

Los usuarios no pueden salir de los directorios de los cuales son propietarios (chroot)

Para especificar que los usuarios no puedan acceder a los directorios de los que no sean propietarios usaremos el argumento “-A” cuando ejecutemos ftpd, o incluyendola en las opciones de “/etc/xinetd.d/pure-ftpd”.

Presentación de mensajes de especificación de contenidos de los distintos directorios.

Práctica 3: Puesta en marcha de un servidor FTP

En los directorios se puede especificar un mensaje que se muestre cuando se acceda a ellos. Este contenido del mensaje se especificará en un archivo “.message“ dentro del directorio correspondiente.

Por ejemplo, en el directorio /musica encontramos:

```
ftp> cd musica
250-La SGAE esta 'golisniando' los FTPs. ¿Nos demandara?
```

Fijar un máximo de conexiones anónimas (opción de compilación –with-peruserlimits)

Añadir la opción '-y' aplicando el formato '-y <max user logins>:<max anonymous logins>'. Para establecer, por ejemplo, 2 conexiones anónimas como máximo estableceremos (y cualquier límite para usuarios autenticados) pondremos:

```
'-y 0:2'
[3er terminal de usuario anónimo]
$ ftp localhost
...
Name (localhost:orlando): ftp
421 No puedo aceptar mas de 2 conexiones del mismo usuario
Login failed.
No control connection for command: Success
```

Limitar el número de conexiones máximas

Mediante el argumento “-c número” se pueden especificar el número máximo de conexiones que puede manejar el servicio.

```
/usr/local/sbin/pure-ftpd -c 3 &
```

El comando anterior nos permite realizar sólo 1 conexión.

```
$ ftp localhost
Connected to localhost.localdomain.
220----- Bienvenido a Pure-FTPd -----
220-Eres el usuario numero 1 de 3 permitidos
220-La hora local es ahora 23:59. Puerto del servidor: 21
...
```

Limitar el número de conexiones por dirección IP

Al igual que el apartado anterior, esta posibilidad se especifica mediante el argumento “-C número”. Esta opción requiere que el servidor esté funcionando en modo “StandAlone”. Limitará el número de conexiones que puede realizar un usuario desde una misma ip. Esta opción requiere que el servidor esté funcionando en modo “StandAlone”.

Práctica 3: Puesta en marcha de un servidor FTP

Ejemplo: Una conexión máxima / IP

```
$ /usr/local/sbin/pure-ftpd -A -y 0:2 -C 1 &
```

```
$ ftp localhost
Connected to localhost.localdomain
421 Demasiadas conexiones (1) desde esta IP
```

Desconexión en caso de inactividad

Por defecto el tiempo máximo de inactividad viene fijado a 15 mins. Pero podemos reducirlo (o ampliarlo), de forma que quizás reduzcamos la carga del servidor con la opción '-I <timeout>'. Estableceremos como tiempo máximo 2 mins.

```
'-I 2'
```

Transcurrido ese tiempo si el cliente intenta ejecutar cualquier comando nos informará de que hemos sido rechazados del servidor.

Limitar el tamaño utilizado de la partición

Para limitar el porcentaje de disco que el usuario anónimo puede utilizar usamos la directiva “-k %”. por ejemplo con -k 90 conseguiremos que el disco nunca se llene más del 90 % de su capacidad mediante ftp.

Limitar el ancho de banda (Requiere compilar con --with-throttling)

Para limitar el ancho de subida y de descarga del servidor ftp por cada usuario utilizamos la opción “-T <Kbytes/segundo>”. Con “-t” limitamos el ancho de banda para el usuario anonimo en concreto. Para distinguir 2 velocidades distintas, una de subida y otra de bajada utilizamos “<subida>:<bajada>”

Pongamos, por ejemplo, '-t 4' para recordar aquellos tiempos de modem 56kb.

```
$ ftp localhost
...
Name (localhost:orlando): orlando
331 Usuario orlando OK. Clave requerida
Password:
Tu utilizacion de ancho de banda esta restringida
```

Posibilidad de acceso en modo de usuario del sistema

En esta ocasión habilitaremos el acceso al servidor ftp de usuarios del sistema. Al igual que en el caso de los usuarios anónimos podemos restringir el acceso al servidor exclusivamente a este tipo de usuarios utilizando la opción “-E”.

La creación de los mismos se realiza llevando a cabo los siguientes comandos:

```
$ adduser usuario  
$ passwd usuario
```

A estos usuarios se les habrá creado por defecto el directorio personal “/home/usuario” que será aquel en el que “atterricen” tras conectar con el servidor ftp.

Opciones

Presentación de un mensaje de bienvenida cuando un usuario acceda al ftp

Como en el caso de los usuarios anónimos el mensaje de bienvenida para este tipo de usuarios será aquel que se encuentre en el archivo “.banner” del directorio personal del usuario con el que conectamos.

Los usuarios puedan salir de los directorio de los cuales son propietarios

Por defecto el usuario puede entrar en todos los directorios a excepción del directorio root. Si no queremos que puedan salir de sus directorios podemos restringirlo con la opción “-A” que restringirá el acceso a todos usuarios menos al root.

Tras conectarnos como usuario, si “-A” está activo, el comando 'pwd' nos mostrará:

```
ftp> pwd  
257 "/" es tu localizacion actual
```

Permitir el acceso a los ficheros definidos por el sistema

Si entendemos como ficheros definidos por el sistema a aquellos que comiencen por un punto, es decir los ficheros ocultos del sistema, para poder listarlos debemos añadir el argumento “-D”.

Desconexión en caso de inactividad

Al igual que en el caso del usuario anónimo la desconexión en caso de inactividad viene fijada por el argumento “-I minutos”.

Limitar el ancho de banda

Como en el caso del usuario anónimo.

Posibilidad de acceso como usuarios virtuales.

"Usuarios virtuales" es un mecanismo de Pure-FTPd que permite almacenar una lista de usuarios con su correspondiente información asociada independientemente de la del sistema (léase '/etc/passwd'). Esto significa que los distintos usuarios virtuales no tendrán una cuenta propia en el sistema (pero sí compartirán un mismo usuario del sistema).

Para permitir este tipo de usuarios debemos realizar una compilación de "PureFTP" con la opción "--with-standalone".

Lo primero que debemos hacer es crear un grupo de usuarios asociado a un usuario del sistema al que agregaremos los usuarios virtuales que creemos. Crearemos como bien explica la documentación de PureFTP el usuario ftpuser y el grupo ftpgroup para uso exclusivo del FTP. Debido a esta exclusividad no será necesario el uso por parte de este usuario de ningún directorio ni de ningún shell. Para crear el grupo y el usuario hacemos ejecutamos las siguientes sentencias:

```
$ groupadd ftpgroup
$ useradd -g ftpgroup -d /dev/null -s /dev/null ftpuser
```

Con la opción "-d" especificamos el directorio del usuario, en este caso sera null y con "-s" el shell asociado al mismo, que volverá a ser null.

Una vez creado el grupo y asociado al usuario ftpuser podemos empezar a gestionar los usuarios virtuales que se incorporaran a este grupo.

Los ficheros que almacenan usuarios virtuales (por defecto "/etc/pureftpd.passwd") contienen un usuario por línea. Estas tienen la siguiente estructura:

```
<account>:<password>:<uid>:<gid>:<gecos>:<home directory>:<upload
bandwidth>:<download bandwidth>:<upload ratio>:<download ratio>:<max number of
connections>:<files quota>:<size quota>:<authorized local Ips>:<refused local
IPs>:<authorized client IPs>:<refused client Ips>:<time restrictions>
```

Algunos campos pueden estar vacíos con la excepción de "account, password, uid, gid, home directory". Debido a la complejidad de estas líneas la aplicación PureFTP trae consigo un programa para generar estos ficheros.

Esta gestión se llevará a cabo mediante el comando "pure-pw", que realizará la gestión de los ficheros de forma automática. La sintaxis para agregar usuarios será la siguiente:

Práctica 3: Puesta en marcha de un servidor FTP

```
pure-pw useradd <login> [-f <passwd file>] -u <uid> [-g <gid>]-D/-d <home directory>
    [-c <gecos>] [-t <download bandwidth>] [-T <upload bandwidth>]
    [-n <max number of files>] [-N <max Mbytes>] [-q <upload ratio>]
    [-Q <download ratio>] [-r <allow client host>[/<mask>]
    [,<allow client host>[/<mask>]]...]
    [-R <deny client host>[/<mask>][,<deny client host>[/<mask>]]...]
    [-i <allow local host>[/<mask>][,<allow client host>[/<mask>]]...]
    [-I <deny local host>[/<mask>][,<deny local host>[/<mask>]]...]
    [-y <max number of concurrent sessions>] [-z <hhmm>-<hhmm>] [-m]
```

Como ejemplo práctico veremos la orden

```
$ pure-pw useradd joe -u ftpuser -d /home/ftpusers/joe
```

que añadirá al usuario 'joe' en la lista de usuarios del FTP, asociándolo con el usuario del sistema 'ftpuser' y asignándole como directorio personal “/home/ftpusers/joe”. Con '-d joe' el usuario 'joe' sólo podrá acceder a directorios de los cuales es propietario. Si, por el contrario, lo queremos es darle acceso al resto del sistema de ficheros usaremos -D en vez de -d.

Aunque el directorio no exista en este momento, se creará si añadimos en la sentencia de ejecución de *PureFTP* la opción “-j” o en la propia línea de creación del usuario.

Una vez creado los nuevos usuarios, debemos guardar los cambios, para ello usaremos la utilidad “pure-pw mkdb” cuya sintaxis es:

```
pure-pw mkdb [<archivo_pdb>][-f <archivo_passwd>]
```

Como hemos utilizado la configuración por defecto con “pure-pw mkdb” nos es suficiente.

Podemos ver los datos de los usuarios virtuales usando el comando “pure-pw show usuario”. Obtendremos como respuesta:

```
Login          : joe
Password       : $1$0037e6M0$Fko/RW82gwNTSnn8rcc261
UID            : 501 (ftpuser)
GID            : 501 (ftpgroup)
Directory      : /home/ftpusers/joe/.
Full name      :
Download bandwidth : 0 Kb (unlimited)
Upload  bandwidth : 0 Kb (unlimited)
Max files      : 0 (unlimited)
Max size       : 0 Mb (unlimited)
Ratio          : 0:0 (unlimited:unlimited)
Allowed local  IPs :
Denied local  IPs :
Allowed client IPs :
Denied client IPs :
Time restrictions : 0000-0000 (unlimited)
Max sim sessions : 0 (unlimited)
```

Práctica 3: Puesta en marcha de un servidor FTP

Para PureFTP permita la entrada de usuarios virtuales le añadimos como argumentos, -l indicándole el archivo de la base de datos.

```
$ /usr/local/sbin/pure-ftpd -j -l puredb:/etc/pureftpd.pdb &
```

Para comprobar el buen funcionamiento del servidor con los usuarios virtuales, entraremos al servidor logueados con el usuario joe.

```
$ ftp localhost
```

```
Connected to localhost.localdomain.
220----- Bienvenido a Pure-FTPd -----
220-Eres el usuario número 1 de 50 permitidos
220-La hora local es ahora 18:34. Puerto del servidor: 21.
220-Las conexiones IPv6 también son bienvenidas en este servidor
220 Serás desconectado después de 15 minutos de inactividad.
500 Extensiones de seguridad no implementadas
500 Extensiones de seguridad no implementadas
KERBEROS_V4 rejected as an authentication type
Name (localhost:root): joe
331 Usuario joe OK. Clave requerida
Password:
230-Usuario joe tiene acceso de grupo a: 501
230 OK. El directorio actual es /
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (127,0,0,1,209,118)
150 Conexión de datos aceptada
226-Options: -l
226 0 ocurrencias en total
```

3. Referencias

- <http://www.pureftpd.org/project/pure-ftpd/doc>
- <http://fferrer.dsic.upv.es/cursos/Linux/Avanzado/HTML/ch09s02.html>
- <http://www.eldemonio.org/documentos/26060513214.html>
- “Man pages” Apartados: pure-ftpd
URL: <http://www.die.net/doc/linux/man/>