

## Unidad 3

---



# Servidor de impresión y de archivos en Linux

## Administración de sistemas operativos





# Índice

## **3.1. Compartir archivos e impresoras**

## **3.2. Servidor de archivos. Samba**

- 3.2.1. Instalar
- 3.2.2. Configurar
- 3.2.3. Añadir usuarios para acceder a los recursos
- 3.2.4. Asignar permisos a los recursos compartidos
- 3.2.5. Iniciar y detener Samba
- 3.2.6. Conectar clientes a samba

## **3.3. CUPS. Servidor de Impresión**

## **3.4. Samba con soporte para la impresión con CUPS**

## **3.5. Herramientas de línea de comandos para el sistema de impresión CUPS**

- 3.5.1. Imprimir
- 3.5.2. Seleccionar impresora
- 3.5.3. Definir la impresora predeterminada
- 3.5.4. Redirigir la salida estándar de un programa a CUPS
- 3.5.5. Especificar opciones de impresión
- 3.5.6. Cancelar un trabajo
- 3.5.7. Habilitar o deshabilitar una impresora

## **3.6. Administración de discos. Cuotas de espacio**

- 3.6.1. Activar cuotas
- 3.6.2. Asignar cuotas de disco



# Introducción

En esta asignatura estamos viendo las principales funciones que tiene un administrador de sistemas. Dentro de estas, otra de las que debemos de destacar es la de administrar la información a la que los usuarios tienen acceso además de que cantidad de recursos se pueden almacenar.

Compartir archivos en una red es de las acciones más llevadas a cabo en un entorno empresarial y puede que con dos sistemas Windows sea sencillo, pero si tenemos sistemas operativos libres y propietarios juntos, deberemos de realizar ciertas configuraciones que vamos a ver en esta unidad.

## Al finalizar esta unidad

- + Sabremos como administrar un servidor de impresión
- + Conoceremos como configurar e instalar un servidor de impresión mediante herramientas integradas en Linux
- + Seremos capaces de añadir y compartir impresoras tanto en entornos Linux como Windows
- + Podremos gestionar impresoras y colas de impresión de trabajos mediante comandos
- + Conoceremos como administrar un servidor de archivos
- + Seremos capaces de compartir carpetas y asignar permisos
- + Podremos asignar cuotas de almacenamiento en discos
- + Sabremos como integrar sistemas operativos libres y propietarios para el uso compartido de recursos



# 3.1.

## Compartir archivos e impresoras

Para poder compartir recursos en Linux, podemos recurrir a Samba, que es una utilidad que permite el acceso a los recursos por parte de sistemas tanto Windows como Linux.

Samba es una herramienta de *software* libre que se basa en el protocolo de archivos compartidos de *Microsoft Windows*, que actualmente se llama *CIFS* pero antes se llamaba *SMB*, de ahí su nombre. Su funcionamiento es en sistemas UNIX y como hemos dicho antes, permite que se conecte a los recursos compartidos desde todo tipo de sistemas.

Además de la compartición de ficheros, colas de impresión y permitir la autenticación propia, *Samba* nos da la oportunidad de establecer en sistemas Linux un servidor con funciones de *AD DC*, *Active Directory & Domain Controller* para poder conectar clientes Windows a servidores Linux.

Esta utilidad se encuentra formada por distintos módulos que nos facilitan la configuración de los protocolos, pero, además, su núcleo interno se compone de tres demonios que funcionan en segundo plano:

- > **smbd**: el demonio que permite compartir ficheros e impresión a clientes Windows. Este demonio también permite la autenticación de los usuarios, bloqueo de recursos y compartición de datos a través del protocolo CIFS. Este tráfico lo recoge por los puertos TCP 139 y 445 de manera predeterminada. Se puede configurar a través del fichero *smbd.conf*.
- > **nmbd**: encargado de responder a las peticiones de servicio de nombres *NetBIOS*. Su puerto predeterminado para gestionar el tráfico *NMB* es el 137 UDP. A este demonio lo controla el servicio **smb**, por lo que se configura mediante el fichero *smb.conf*.
- > **winbindd**: se encarga de resolver información acerca de grupos y usuarios en servidores Windows para luego trasladarla de forma entendible a los sistemas Linux. Con esto se permite que los usuarios de los dominios de Windows puedan operar como usuarios UNIX en estos sistemas. Es controlado por el servicio **winbind** y no es necesario que se arranque el servicio **smb** para su funcionamiento.





# 3.2.

## Servidor de archivos. Samba

Debemos de llevar a cabo 5 pasos para establecer Samba como servidor de archivos:

1. Instalar el servidor Samba.
2. Configurar el servidor acorde a nuestras necesidades.
3. Añadir a los usuarios que necesitemos que accedan a los recursos compartidos.
4. Dar los permisos a dichos usuarios sobre las carpetas que se necesite.
5. Arrancar el servidor.

### 3.2.1. Instalar

Lo primero que debemos hacer es instalar el servidor Samba en nuestro sistema. Para esto debemos de ser superusuario e instalar el paquete *samba*.

Una vez que se ha instalado, queda definida una pequeña configuración previa que se refleja en el fichero */etc/samba/smb.conf*. Dicho fichero es el que vamos a ir cambiando durante estos apartados para adaptar a nosotros la configuración.

```

GNU nano 6.2                                smb.conf
#
# Sample configuration file for the Samba suite for Debian GNU/Linux.
#
#
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options most of which
# are not shown in this example
#
# Some options that are often worth tuning have been included as
# commented-out examples in this file.
# - When such options are commented with ";", the proposed setting
#   differs from the default Samba behaviour
# - When commented with "#", the proposed setting is the default
#   behaviour of Samba but the option is considered important
#   enough to be mentioned here
#
# NOTE: Whenever you modify this file you should run the command
# "testparm" to check that you have not made any basic syntactic
# errors.

##### Global Settings #####

[global]

## Browsing/Identification ##

# Change this to the workgroup/NT-domain name your Samba server will part of
workgroup = WORKGROUP

# server string is the equivalent of the NT Description field
server string = %h server (Samba, Ubuntu)

```

Imagen 1. Fichero */etc/samba/smb.conf*



### 3.2.2. Configurar

El segundo paso para Samba es la configuración de este. Para esto vamos a configurar el fichero `/etc/samba/smb.conf`, fichero que se divide en diferentes secciones, con cada sección identificada con un título entre corchetes. En cada sección tenemos distintos parámetros que podemos activar o desactivar.

En el fichero de configuración tenemos la opción de añadir al principio de la línea una almohadilla, `'#'` para que esta figure como comentario y así no está activada.

Tenemos tres secciones predefinidas y principales, que son:

- > **[global]**: sirve para definir los parámetros globales del servidor.
- > **[homes]**: sirve para definir las carpetas de trabajo de los usuarios como recursos compartidos.
- > **[printer]**: sirve para definir las impresoras compartidas.

Vamos a ver ahora por pasos como realizar una configuración inicial del servidor:

1. Todos los pasos se deben de realizar como superusuario.
2. Creamos un directorio que actúe como recurso principal para samba:

`mkdir directorio`

```
profesor@universae:~$ sudo mkdir /samba
profesor@universae:~$ ls /
bin  dev  home  lib32  libx32  media  opt  root  samba  snap  swap.img  tmp  var
boot  etc  lib  lib64  lost+found  mnt  proc  run  sbin  srv  sys  usr
```

Imagen 2. Configuración de samba 1

3. Hacemos una copia de seguridad del fichero de configuración `/etc/samba/smb.conf` con el siguiente comando:

`sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.orig`

```
profesor@universae:/etc/samba$ sudo cp smb.conf smb.conf.orig
[sudo] password for profesor:
profesor@universae:/etc/samba$ ls
gdbcommands  smb.conf  smb.conf.orig  tmp
profesor@universae:/etc/samba$ _
```

Imagen 3. Configuración de samba 2

4. Entramos en el fichero de configuración de Samba y añadimos una nueva sección con el siguiente formato:

```
[nombre-sección]
comment = Descripción
path = directorio de trabajo
read only = no
browseable = yes
```

```
[samba-compartido]
comment = Samba
path = /samba
read only = no
browseable = yes
```

Imagen 4. Configuración de samba 3



Cada uno de estos parámetros indica lo siguiente:

- > **[nombre-sección]**: nombre de la sección.
- > **comment**: descripción del contenido del directorio.
- > **path**: ruta completa al directorio de trabajo que hemos creado previamente.
- > **read only = no**: indica que podamos leer y escribir en la carpeta.
- > **browseable = yes**: indica que podemos navegar por los directorios que creemos dentro de este.

### 3.2.3. Añadir usuarios para acceder a los recursos

Los usuarios que van a acceder a los recursos de Samba deben de tener una cuenta creada en el servidor y posteriormente añadirlos a la base de datos de samba.

Para añadir a los usuarios realizamos los siguientes pasos:

1. Lo primero es añadir al usuario al sistema Linux.
2. Una vez creado, lo añadimos a la base de datos de samba con el siguiente comando:

**smbpasswd -a usuario**

```
profesor@universae:/etc/samba$ sudo useradd samba
[sudo] password for profesor:
profesor@universae:/etc/samba$ sudo smbpasswd -a samba
New SMB password:
Retype new SMB password:
Added user samba.
profesor@universae:/etc/samba$ _
```

Imagen 5. Crear usuarios en Samba

Cuando añadamos el usuario a Samba, se nos pedirá una contraseña para poder conectarnos a los recursos compartidos.

### 3.2.4. Asignar permisos a los recursos compartidos

Dentro de la carpeta que hemos creado en una primera instancia crearemos otra que podemos ver, por defecto tendrá los permisos de leer y ejecutar solamente, con *root* como propietario y grupo propietario.

```
profesor@universae:/etc/samba$ sudo mkdir /samba/compartido
profesor@universae:/etc/samba$ ls -l /samba
total 4
drwxr-xr-x 2 root root 4096 ago  4 13:39 compartido
profesor@universae:/etc/samba$
```

Imagen 6. Asignación de permisos en samba 1

#### IMPORTANTE

Si queremos que el usuario pueda ejecutar ciertos servicios como SSH o telnet, necesitaremos especificar una contraseña, pero si por el contrario no es necesario que se acceda a ningún servicio, es recomendable que el usuario no tenga contraseña y que además su Shell sea o bien */sbin/nologin* o bien */bin/false*, para que no pueda iniciar sesión. En nuestro caso lo hemos dejado con el Shell por defecto, pero sin contraseña.



Debemos ahora de cambiar los permisos, pero siguiendo un orden:

1. Primero cambiamos el usuario propietario del fichero por el usuario *nobody*.
2. Segundo, otorgamos los permisos al completo para este directorio.

```
profesor@universae:/etc/samba$ sudo chown -R nobody /samba/compartido/
profesor@universae:/etc/samba$ sudo chmod -R 0777 /samba/compartido/
profesor@universae:/etc/samba$ _
```

Imagen 7. Asignación de permisos en samba 2

3. Cambiamos el grupo propietario por el predeterminado de samba, que es *smbashare*.

```
profesor@universae:/etc/samba$ sudo chgrp smbashare /samba/compartido/
profesor@universae:/etc/samba$
```

Imagen 8. Asignación de permisos en samba 3

4. Comprobamos que los cambios han surtido efecto.

```
profesor@universae:/etc/samba$ ls -l /samba/
total 4
drwxrwxrwx 2 nobody smbashare 4096 ago  4 13:39 compartido
profesor@universae:/etc/samba$ _
```

Imagen 9. Asignación de permisos en samba 4

### 3.2.5. Iniciar y detener Samba

Para que todos los cambios que hemos realizado tomen efecto, tendremos que reiniciar el servicio de samba, que se llama *smbd.service* y que podemos arrancar, parar, reiniciar o comprobar con el comando **systemctl**.

```
profesor@universae:/etc/samba$ sudo systemctl start smbd
profesor@universae:/etc/samba$ sudo systemctl stop smbd
profesor@universae:/etc/samba$ sudo systemctl restart smbd
profesor@universae:/etc/samba$ sudo systemctl status smbd
* smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-08-04 14:03:38 CEST; 6s ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
  Process: 4055 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (code=exited, status=0)
 Main PID: 4064 (smbd)
   Status: "smbd: ready to serve connections..."
    Tasks: 4 (limit: 2241)
   Memory: 8.7M
      CPU: 79ms
   CGroup: /system.slice/smbd.service
           └─4064 /usr/sbin/smbd --foreground --no-process-group
             └─4066 /usr/sbin/smbd --foreground --no-process-group
               └─4067 /usr/sbin/smbd --foreground --no-process-group
                 └─4068 /usr/lib/x86_64-linux-gnu/samba/samba-bgqd --ready-signal-fd=45 --parent-watch=5

ago 04 14:03:38 universae.lan systemd[1]: Starting Samba SMB Daemon...
ago 04 14:03:38 universae.lan systemd[1]: Started Samba SMB Daemon.
lines 1-20/20 (END)
```

Imagen 10. Opciones de systemctl con smbd



### 3.2.6. Conectar clientes a samba

Ya hemos visto como configurar el servidor samba para que funcione como un servidor de ficheros, pero ahora tenemos que ver cómo debemos conectarnos a las carpetas compartidas.

#### Cliente Linux

Si queremos conectarnos desde Linux, debemos de seguir los siguientes pasos:

1. Ir al explorador de archivos del sistema, en este caso *Ubuntu Desktop*.
2. En la barra de la izquierda, seleccionar la última de las secciones, '*Otras ubicaciones*'.
3. Una vez aquí al final de la ventana tenemos la opción '*Conectar al servidor*', donde tendremos que escribir:

`smb://IP_servidor/`



Imagen 11. Conexión a Samba desde cliente Linux 1

Indicamos *smb* porque como hemos dicho antes, es el protocolo usado por samba.

4. Ahora vemos que nos aparecen las secciones predefinidas, accedemos a la nuestra.

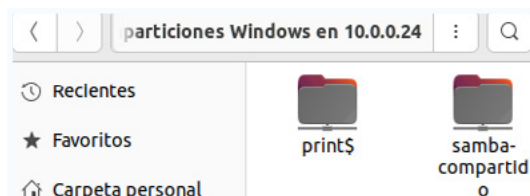


Imagen 12. Conexión a Samba desde cliente Linux 2

5. Se nos pedirá usuario y contraseña, que es el que hemos creado anteriormente.

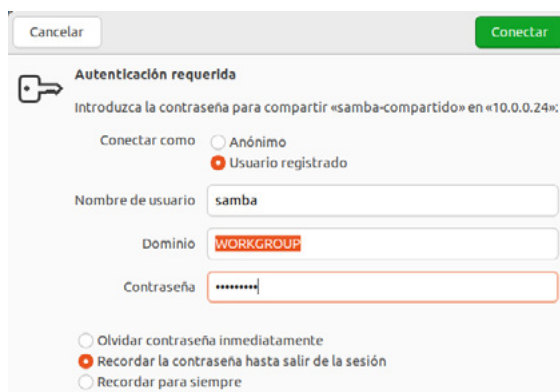


Imagen 13. Conexión a samba desde cliente Linux 3



6. Ya visualizamos el contenido de la carpeta que creamos en una primera instancia.



Imagen 14. Conexión a Samba desde cliente Linux 4

7. Si dentro de esta carpeta probamos a crear otra, nos dirá que no podemos, porque no es a la que le hemos asignado los permisos.

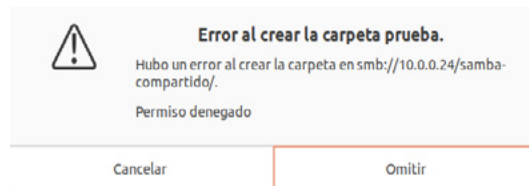


Imagen 15. Conexión a Samba desde cliente Linux 5

8. Entramos en la carpeta con los permisos asignados y vemos que aquí sí que nos deja crear otra carpeta.

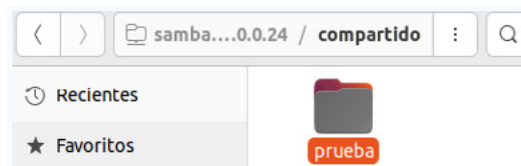


Imagen 16. Conexión a Samba desde clientes Linux 6

9. Vemos que dicha carpeta se puede visualizar en nuestro servidor si nos vamos a la ruta de compartición, con el usuario que ha accedido como propietario.

```
profesor@universae:/samba/compartido$ ls -l
total 4
drwxr-xr-x 2 samba samba 4096 ago  4 14:24 prueba
profesor@universae:/samba/compartido$ _
```

Imagen 17. Conexión a Samba desde clientes Linux 7



## Ciente Windows

Para conectarnos a Samba desde un cliente Windows, que era una de las mejores ventajas de Samba, debemos de hacer lo siguiente:

1. Conectar el cliente a la misma red y permitir que los dispositivos de la red nos detecten.



Imagen 18. Conexión a Samba desde clientes Windows 1

2. En el inicio de Windows, buscamos 'Ejecutar' y abrimos la aplicación.

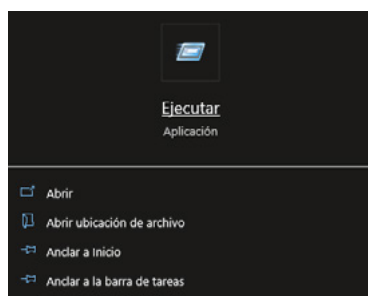


Imagen 19. Conexión a Samba desde clientes Windows 2

3. Una vez en la aplicación, escribimos lo siguiente:

[\\VP\\_servidor\recurso\\_compartido](#)

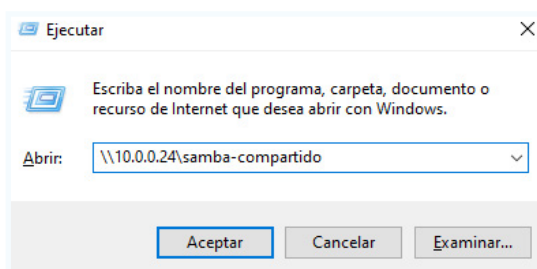


Imagen 20. Conexión a Samba desde clientes Windows 3

4. El sistema nos pedirá las credenciales de red del usuario con acceso:

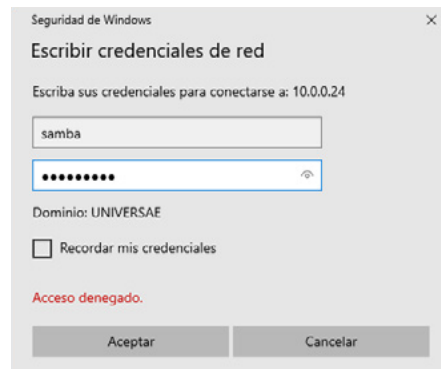


Imagen 21. Conexión a Samba desde clientes Windows 4

Podemos ver que nos aparece el nombre del dominio porque lo detecta.

5. En el explorador de archivos ya nos aparece disponible la carpeta, pero de nuevo no nos deja crear nuevos elementos.

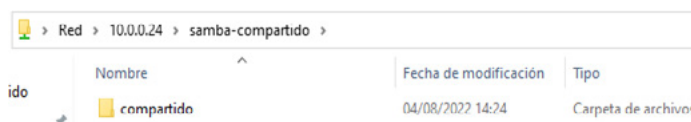


Imagen 22. Conexión a Samba desde clientes Windows 5

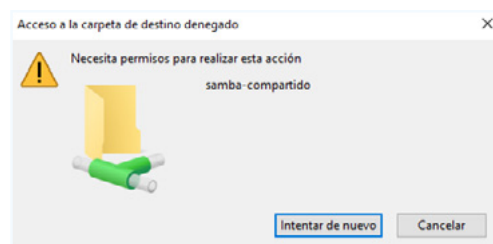


Imagen 23. Conexión a Samba desde clientes Windows 6

6. Si entramos en el recurso con los permisos asociados que creamos al principio, podemos comprobar que si nos permite crear objetos del explorador y...

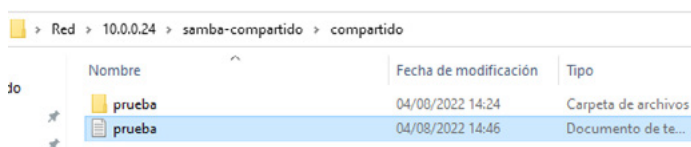


Imagen 24. Conexión a Samba desde clientes Windows 7

7. En el servidor comprobamos que todo ha funcionado correctamente.

```
profesor@universae:/samba/compartido$ ls -l
total 8
drwxr-xr-x 2 samba samba 4096 ago  4 14:24 prueba
-rwxr--r-- 1 samba samba 19 ago  4 14:46 prueba.txt
profesor@universae:/samba/compartido$ cat prueba.txt
¡Samba Funciona!
profesor@universae:/samba/compartido$
```

Imagen 25. Conexión a Samba desde clientes Windows 8

## PARA TENER EN CUENTA

Debemos saber que los usuarios crean objetos con permisos solo de lectura (y ejecución en caso de directorios) por defecto, por lo que, si en las carpetas que estos han creado, otro usuario quisiera crear más objetos, no podría porque no tendría permisos.



# 3.3.

## CUPS. Servidor de Impresión

Para imprimir en UNIX, el principal mecanismo usado es el sistema de impresión común de UNIX, llamado *CUPS*. Esta utilidad se basa en impresión libre y portable además de ser gratuita, lo que la ha convertido en el principal medio de impresión en los sistemas Linux. El funcionamiento de CUPS se basa en gestionar los trabajos y colas de impresión basándose en el estándar IPP, *Internet Printing Protocol* o protocolo de impresión en internet. CUPS es además compatible con PPD, *PostScript Printer Description* y es capaz de detectar automáticamente impresoras en red. Gracias a todas estas ventajas, CUPS también permite soportar una amplia gama de impresoras independientemente de su marca, función o funcionamiento.

CUPS viene ya preinstalado en todas las versiones de Ubuntu y funciona de manera automática, pero por si no, lo que debemos de instalar son los siguientes paquetes: *cups* & *cups-client*.

Los ficheros de configuración de CUPS se alojan en el directorio */etc/cups*.

```
profesor@universae:~$ cd /etc/cups/
profesor@universae:/etc/cups$ ls
cups-browsed.conf  cups-files.conf  ppd          raw.types  ssl
cupsd.conf         interfaces       raw.convs    snmp.conf  subscriptions.conf
profesor@universae:/etc/cups$
```

Imagen 26. Directorio */etc/cups*

Y, el principal fichero de configuración de CUPS es */etc/cups/cupsd.conf*.

```
GNU nano 6.2 cupsd.conf
# Configuration file for the CUPS scheduler. See "man cupsd.conf" for a
# complete description of this file.
#
# Log general information in error_log - change "warn" to "debug"
# for troubleshooting...
LogLevel warn
PageLogFormat
# Specifies the maximum size of the log files before they are rotated. The value "0" disables log
MaxLogSize 0
# Default error policy for printers
ErrorPolicy retry-job
# Only listen for connections from the local machine.
Listen localhost:631
# Listen /run/cups/cups.sock
# Show shared printers on the local network.
Browsing No
BrowseLocalProtocols dnssd
# Default authentication type, when authentication is required...
DefaultAuthType Basic
# Web interface setting...
WebInterface Yes
# Timeout after cupsd exits if idle (applied only if cupsd runs on-demand - with -l)
IdleExitTimeout 60
```

Imagen 27. Fichero */etc/cups/cupsd.conf*

Pero, realmente este fichero de configuración no hay que tocarlo apenas, porque se basa en su mayoría en la gestión web. Para esto, CUPS automáticamente genera una aplicación web de gestión del servicio al que se puede acceder introduciendo en un navegador la siguiente dirección:

<http://localhost:631>



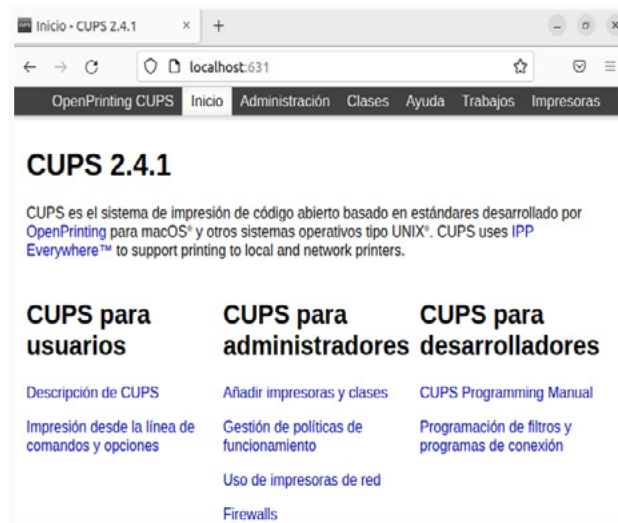


Imagen 28. Página web de CUPS

Cabe destacar que, en nuestro caso, como *Ubuntu Server* no tiene interfaz gráfica, nos hemos conectado mediante SSH al servidor de manera gráfica para poder ejecutar el servicio en la web.

En esta aplicación, nos debemos de fijar en la página '*Administración*', donde se nos da la opción de agregar una impresora nueva, a lo que nos pedirá el usuario y contraseña del sistema para poder realizar dicha acción.

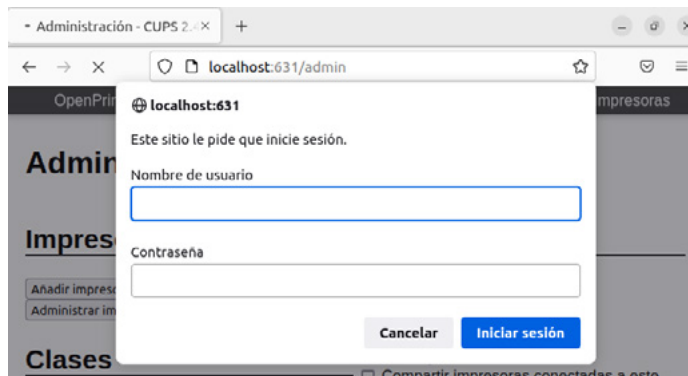


Imagen 29. Página web de administración de CUPS

Cuando estamos en la página de añadir impresoras, tendremos diversas opciones para añadir impresoras como podemos en la siguiente imagen:



Imagen 30. Página web para añadir impresoras en CUPS

Otra opción de configuración interesante es en la página principal, activar la casilla '*Compartir impresoras públicas conectadas a este sistema*' para que CUPS pueda compartir la impresora basándose en el protocolo *http*.

#### Configuración del servidor:

##### Avanzada ►

- ☒ **Compartir impresoras conectadas a este sistema**
  - ☐ Permitir la impresión desde Internet
  - ☐ Permitir administración remota
  - ☐ Usar autenticación Kerberos (FAQ)
- ☐ Permitir a los usuarios cancelar cualquier trabajo (no sólo los suyos propios)
  - ☐ Guardar información de depuración para búsqueda de problemas

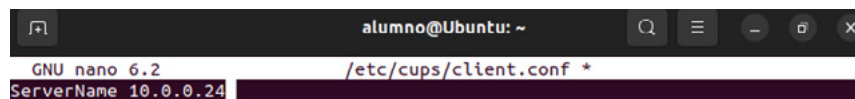
[Cambiar configuración](#)

Imagen 31. Opciones avanzadas de CUPS

Por último, para conectar clientes al servidor de impresión, debemos de instalar el paquete *cups-client*.

La configuración de este paquete debe de hacerse a través del fichero */etc/cups/client.conf*, donde tenemos que añadir la siguiente línea:

**ServerName** IP\_Servidor

Imagen 32. Fichero */etc/cups/client.conf*

Muchas veces, dicho fichero no existe, por lo que tendremos que crearlo nosotros con esta entrada anteriormente comentada.



# 3.4.

## Samba con soporte para la impresión con CUPS

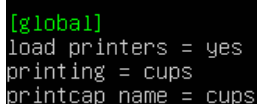
Como se ha dicho en puntos anteriores, *Samba* nos permite que se haga una conexión del servidor de impresión con impresoras compartidas con sistemas Windows.

Es verdad que hay diversos sistemas de impresión, pero el más apropiado es CUPS, debido a que se integra a la perfección con *Samba*.

Vamos a ver ahora los parámetros que debemos de definir para su configuración en el fichero `/etc/samba/smb.conf`.

1. Lo primero que debemos hacer es en el fichero es añadir en la sección `[global]`, las siguientes líneas:

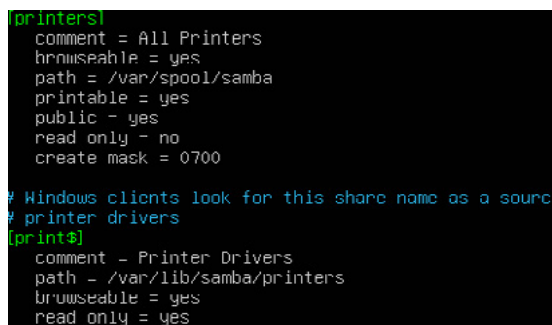
```
load printers = yes
printing = cups
printcap name = cups
```



```
[global]
load printers = yes
printing = cups
printcap name = cups
```

Imagen 33. Configuración de samba para CUPS 1

2. Un ejemplo de configuración básica del fichero para el soporte CUPS puede ser el que vemos en la siguiente imagen:



```
[printers]
comment = All Printers
browseable = yes
path = /var/spool/samba
printable = yes
public = yes
read only = no
create mask = 0700

# Windows clients look for this share name as a source
# printer drivers
[print$]
comment = Printer Drivers
path = /var/lib/samba/printers
browseable = yes
read only = yes
```

Imagen 34. Configuración de samba para CUPS 2

Podríamos establecer configuraciones que sean más difíciles, pero no es necesario que hagamos hincapié en eso ahora mismo, porque realmente cada sistema o cada impresora dictaminará como tenemos que hacer las configuraciones. Lo que sí es importante aquí es saber que el recurso último que hemos especificado, `[print$]` recoge los controladores de la impresora para que cuando intentemos conectar con un cliente, se puedan obtener de aquí y no localmente. Lógicamente este recurso es opcional, porque siempre podemos obtenerlos en el cliente directamente. Además, al indicar la opción `browseable` como verdadera, estamos diciendo que en el entorno de red se pueda ver y detectar la impresora, siempre que nuestro servidor esté correctamente configurado.

Para que todos los cambios surjan efecto debemos de reiniciar el servidor de *Samba*.



# 3.5.

## Herramientas de línea de comandos para el sistema de impresión CUPS

La impresión mediante CUPS soporta el uso de varios tipos de archivos, como pueden ser imágenes o PDF. Gracias a esto, tenemos en CUPS la opción de imprimir directamente desde aplicaciones o usando la línea de comandos. Los distintos comandos de impresión que vamos a indicar a continuación los encontramos para el uso de *root* o de usuarios normales, pero la potencia distinguirá entre quien lo use. El mayor ejemplo de esto está en la cola de impresión. Ya que un usuario normal solo puede cancelar sus tareas pendientes, mientras que *root* las puede cancelar todas.

### 3.5.1. Imprimir

Si queremos imprimir desde la ruta de comandos debemos de hacerlo con el comando `lp`, y se hará por la impresora predeterminada, con la siguiente estructura:

#### `lp fichero`

Si, por otro lado, lo que queremos es imprimir por otra impresora distinta, se hará con el mismo comando, pero indicando al final el nombre de la impresora con la opción `-d`.

Pero para poder controlar nuestros trabajos en marcha necesitamos poder ver la cola de impresión, y esto lo realizamos con el comando `lpq`.

```
root@universae:~# lpq
no entries
root@universae:~# _
```

Imagen 35. Comando `lpq`

### 3.5.2. Seleccionar impresora

Para en el comando `lp`, poder seleccionar una impresora en caso de que tengamos varias, debemos de saber primero el nombre de las impresoras que tenemos conectadas. Para ver esta lista usamos el comando `lpstat` con las siguientes opciones:

- > `-p`: para indicar que queremos ver una lista de las impresoras actuales.
- > `-d`: nos indica cual es la impresora actual predeterminada.

```
root@universae:~# lpstat -p -d
lpstat: No se han añadido destinos.
no hay un destino predeterminado del sistema
root@universae:~# _
```

Imagen 36. Comando `lpstat`

Podemos ver en la imagen que no tenemos ninguna impresora configurada.



### 3.5.3. Definir la impresora predeterminada

Si queremos que una impresora en específico sea la predeterminada, podemos usar el comando **lpoptions**, a continuación, tenemos una imagen con sus principales opciones:

```
root@universae:~# lpoptions -d
Usage: lpoptions [options] [-d destination] [-l]
lpoptions [options] [-p destination] -o option[=value]
lpoptions [options] -x destination

Opciones:
-d destination      Set default destination
-E                  Encrypt the connection to the server
-h server[:port]    Connect to the named server and port
-l                  Show supported options and values
-n name[=value]     Set default option and value
-p destination      Specify a destination
-U username         Specify the username to use for authentication
-x destination      Remove default options for destination
root@universae:~#
```

Imagen 37. Opciones de lpoptions

### 3.5.4. Redirigir la salida estándar de un programa a CUPS

Para indicar que la salida de un programa en concreto se imprima desde una impresora usaríamos el comando siguiente:

comando | lp opciones impresora

### 3.5.5. Especificar opciones de impresión

Aunque es verdad que en casi todos los casos usaremos las opciones de impresión predeterminadas, puede que haya otros aspectos que debamos de editar para que se impriman de un modo en concreto y no genérico.

La opción que incluye para esto **lp** es **-o** para cualquier configuración y **-n** para indicar el número de copias que deseamos.

### 3.5.6. Cancelar un trabajo

Cuando lanzamos impresiones, habrá ocasiones en las que queramos cancelar trabajos por distintas cuestiones, ya sea por error o por mantenimiento. Para realizar dicha acción usamos el comando **lprm** al que, si no indicamos nada, borra todos los trabajos (si no es *root*, solo los del usuario que lo ejecute) y si le añadimos un número de trabajo, solo elimina dicho trabajo.

Puede que no siempre queramos mover un trabajo, sino moverlo a otra impresora o incluso servidor por distintos motivos. Para realizar esta acción existe **lpmove**, que cuenta con distintas opciones. Su sintaxis y opciones, las vemos en la imagen de más abajo:

```
root@universae:~# lpmove
Usage: lpmove [options] job destination
lpmove [options] source-destination destination

Opciones:
-E                  Encrypt the connection to the server
-h server[:port]    Connect to the named server and port
-U username         Specify the username to use for authentication
root@universae:~#
```

Imagen 38. Opciones de lpmove

### 3.5.7. Habilitar o deshabilitar una impresora

Las impresoras que se pueden o no usar se pueden ir cambiando, según las activemos o desactivemos con, respectivamente, los siguientes comandos:

- > cupsenable impresora
- > cupsdisable impresora





# 3.6.

## Administración de discos. Cuotas de espacio

Cuando trabajamos en un mismo sistema con varios usuarios, hay que llevar cuidado con la cantidad de datos que se pueden almacenar, pues pueden saturarnos el sistema. Para frenar esto existen las cuotas de disco y podemos fijarlas para los usuarios individuales o para grupos.

### 3.6.1. Activar cuotas

Lo primero que debemos hacer es instalar los paquetes de Ubuntu que nos van a ayudar a crear las cuotas, aunque en su mayoría estos vienen instalados por defecto, pero por si acaso, los paquetes que debemos instalar son *quota & quotatool*.

Una vez que tenemos todo instalado, debemos de tener el sistema montado correctamente y de acuerdo con las directrices que vamos a seguir, para lo que editamos el fichero */etc/fstab*.

En dicho fichero, aunque lo recomendable es editar el directorio */home*, nosotros lo vamos a hacer sobre el directorio raíz del sistema y en la línea en la que este aparezca, debe de quedar como en la siguiente imagen, con los parámetros *usrquota,grpquota* añadidos.

```

GNU nano 6.2 /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust
# that works even if disks are added and removed. See
#
# <file system> <mount point> <type> <options>
# / was on /dev/sda3 during installation
<| ext4 usrquota,grpquota 0 1
# /boot/efi was on /dev/sda2 during installation
UUID=151D-A258 /boot/efi vfat umask=0077
/swapfile none
  
```

Imagen 39. Fichero */etc/fstab*

Una vez que tenemos el fichero editado, lanzamos el siguiente comando para volver a montar la partición:

```
mount -o remount /
```

Además, podemos comprobar que esto ha tenido efecto consultando al fichero */proc/mounts* filtrando por el directorio raíz.

```

root@Ubuntu: ~
root@Ubuntu:~# mount -o remount /
root@Ubuntu:~# cat /proc/mounts | grep ' / '
/dev/sda3 / ext4 rw,relatime,quota,usrquota,grpquota,errors=remount-ro 0 0
root@Ubuntu:~#
  
```

Imagen 40. Remontaje y comprobación

Podemos fijarnos en que todos los comandos aquí usados se realizan siendo superusuario, lógico, porque estamos editando el sistema.



### 3.6.2. Asignar cuotas de disco

Cuando vayamos a activar las cuotas de disco, lo primero que debemos de hacer es comprobar que todo se encuentra correcto, y para eso tenemos el comando `quotacheck`.

Con `quotacheck` podemos usar diversas opciones como pueden ser las siguientes:

- > `-c`: se crea el fichero de cuotas si no existe.
- > `-u`: cuota de usuario de disco.
- > `-g`: cuota de grupo de disco
- > `-m`: no monta de nuevo los sistemas de archivos.

```
root@Ubuntu:~# quotacheck -cum /
root@Ubuntu:~# quotacheck -cgm /
root@Ubuntu:~# quotacheck -cugm /
root@Ubuntu:~#
```

Imagen 41. Comando quotacheck

Una vez que se ha comprobado todo, podemos activar o desactivar las cuotas de disco con los siguientes comandos respectivamente:

- > `quotaon -v directorio`
- > `quotaoff -v directorio`

En ambos casos se especifica la opción `-v` para que nos muestre el proceso.

```
root@Ubuntu:~# quotaon -v /
quotaon: Your kernel probably supports ext4 quota feature but you are using external quota files. Please switch your filesystem to use ext4 quota feature as external quota files on ext4 are deprecated.
/dev/sda3 [/]: group quotas activadas
/dev/sda3 [/]: user quotas activadas
root@Ubuntu:~#
```

Imagen 42. Activar cuotas

Ahora, podemos pasar a asignar las cuotas, para lo que usamos el siguiente comando:

`edquota -u usuario`

Cuando lanzamos dicho comando, nos aparecerá un editor de texto con los datos del usuario y varias columnas, que son:

- > **Sist.arch.** Nos indica en que partición estamos trabajando.
- > **Bloques.** Nos indica el número de bloques que actualmente ocupa el usuario.
- > **Blando.** El número máximo de bloques que puede usar el usuario hasta que se le muestre un aviso.
- > **Duro.** El número máximo de bloques que el usuario nunca puede sobrepasar.



Las demás columnas son perfectamente deducibles y no necesarias, pues solo vamos a editar la 3ª y la 4ª.

```
GNU nano 6.2 /tmp//EdP.atJsR2l
Cuotas de disco para user alumno2 (uid 1001):
Sist. arch. bloques blando duro inodos blando
/dev/sda3 16 0 0 4
```

Imagen 43. Asignar cuotas 1

En la siguiente imagen podemos ver un ejemplo de asignación de cuotas:

```
GNU nano 6.2 /tmp//EdP.aqNdADz *
Cuotas de disco para user alumno2 (uid 1001):
Sist. arch. bloques blando duro inodos blando
/dev/sda3 16 1G 2G 4
```

Imagen 44. Asignar cuotas 2

Como hemos visto, se le han asignado al usuario 1GB para que nos avise de sobrepasar y dos para que no nos deje crear más.

Por último, podemos comprobar que todo funciona bien y que se ha guardado correctamente con el comando:

#### quota -vs usuario

```
root@Ubuntu:~# quota -vs alumno2
Cuotas de disco para user alumno2 (uid 1001):
Sistema de archivos espacio quota límite graciaarchivos quota límite gracia
/dev/sda3 16K 1024M 2048M 4 0 0
```

Imagen 45. Asignar cuotas 3



 [www.universae.com](http://www.universae.com)

