

Sistemas Operativos en Red

# UD 11 - Compartir ficheros e impresoras en Linux con Samba

---



Consejería d'Educació,  
Investigació, Cultura i Esport

I.E.S. SERRA PERENXISA



Fons Social Europeu

UNIÓ EUROPEA

L'FSE inverteix en el teu futur

Autor: Sergi García

Actualizado Enero 2024



## Licencia



**Reconocimiento - No comercial - CompartirIgual (BY-NC-SA):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se ha de hacer con una licencia igual a la que regula la obra original.

## Nomenclatura

A lo largo de este tema se utilizarán diferentes símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

**Importante**

**Atención**

**Interesante**

## ÍNDICE

<b>1. Introducción</b>	<b>3</b>
<b>2. Compartir ficheros e impresoras en Linux</b>	<b>3</b>
<b>3. Servidor Samba: Windows y Linux coexistiendo en una misma red</b>	<b>4</b>
3.1 Paso 0: Cambiar nombre de equipo en Linux	4
3.2 Paso 1: Instalación de Samba	4
3.3 Paso 2: Compartiendo recursos en Samba	5
3.4 Paso 4: Registro de usuarios	6
3.5 Paso 5: Inicio y parada del servidor Samba	7
3.6 Paso 6: Testeando configuraciones con "testparm"	7
<b>4. Accediendo a una carpeta compartida</b>	<b>8</b>
4.1 Acceso a carpetas compartidas en sistemas Linux	8
4.1.1 Desde navegador de ficheros	8
4.1.2 Desde consola de comandos	8
4.2 Acceso a carpetas compartidas en sistemas Windows	9
4.3 Montaje de carpetas compartidas en sistemas Linux	9
4.4 Montaje de carpetas compartidas de forma automática al iniciar un sistema Linux	9
<b>5. Escritorio móvil con Samba [home]</b>	<b>10</b>
5.1 Paso 1: Configuración en el servidor	10
5.2 Paso 2: Configuración en los clientes Linux	10
<b>6. Cuotas de disco en Linux</b>	<b>11</b>
6.1 Ejemplos de comandos para establecer cuotas	11
<b>7. Servidor de impresión CUPS</b>	<b>12</b>
<b>8. Instalar impresora PDF en CUPS</b>	<b>13</b>
<b>9. Compartiendo impresoras instaladas en CUPS con Samba</b>	<b>13</b>
<b>10. Montar impresora compartida por Samba en el cliente</b>	<b>14</b>
<b>11. Herramientas gráficas para gestionar Samba</b>	<b>14</b>
<b>12. Bibliografía</b>	<b>14</b>

## UNIDAD 11. COMPARTIR FICHEROS E IMPRESORAS EN LINUX CON SAMBA

### 1. INTRODUCCIÓN

Cuando nos encontramos en la posición de administradores de sistemas a cargo de servidores de red con múltiples usuarios, la gestión de la información a la que pueden acceder y los recursos que pueden utilizar se vuelve una tarea crucial. Una de las acciones más comunes y relevantes en este contexto es la compartición de archivos, carpetas e impresoras, así como la gestión de permisos de acceso de los mismos. En esta unidad, vamos a ver como realizar esta tarea desde Linux con Samba (Implementación libre del protocolo de archivos compartidos de Microsoft Windows).

### 2. COMPARTIR FICHEROS E IMPRESORAS EN LINUX

La compartición de archivos en entornos Linux se simplifica considerablemente gracias a la presencia de Samba. Esta herramienta, que implementa de forma libre el protocolo de archivos de Microsoft Windows (anteriormente conocido como SMB y actualmente denominado CIFS), facilita la interoperabilidad entre sistemas Linux/Unix y Windows. Con Samba, la conectividad entre ordenadores con Linux, MAC OS X o Unix y sistemas operativos Windows en una misma red se vuelve posible, permitiendo el intercambio de archivos y la utilización de impresoras sin importar el sistema operativo de los dispositivos en la red.

Inicialmente diseñado para compartir archivos, impresoras y otros servicios en redes con sistemas operativos Windows, Samba ha evolucionado para posibilitar la integración de sistemas Windows con entornos GNU/Linux. Esta integración permite el intercambio de archivos y el uso de diferentes impresoras de manera fluida, independientemente del sistema operativo presente en los dispositivos de la red.

Además de sus capacidades fundamentales, Samba ofrece la posibilidad de actuar como controlador de dominio, permitiendo el uso del servicio como Active Directory y controlado de dominio. Esto resulta especialmente útil para conectar clientes Windows a un servidor Linux.

La arquitectura de Samba se compone de varios módulos destinados a simplificar la configuración del protocolo SMB/CIFS. Su núcleo consta de tres servicios que operan en segundo plano de manera constante:

- **smbd**: responsable de compartir archivos e impresiones a clientes Windows, así como de autenticar usuarios, bloquear recursos y compartir datos mediante el protocolo SMB. Los puertos predeterminados para el tráfico SMB son TCP 139 y 445, siendo configurables desde el archivo *"smb.conf"*.
- **nmbd**: encargado de comprender y responder a las solicitudes de servicios de nombres NetBIOS. Funciona en el puerto UDP 137 por defecto y es controlado por el servicio *"smbd"*, configurándose a través del archivo *"smb.conf"*.
- **winbindd**: resuelve la información de grupos y usuarios en un servidor Windows, convirtiéndola en comprensible para sistemas Unix/Linux. Esto permite que los usuarios del dominio Windows aparezcan y operen como usuarios Unix en una máquina Unix. El servicio winbind controla el demonio winbindd, y a diferencia de smb, no necesita que este último se inicie para funcionar.

### 3. SERVIDOR SAMBA: WINDOWS Y LINUX COEXISTIENDO EN UNA MISMA RED

Comúnmente, nos enfrentamos a desafíos al intentar compartir impresoras y archivos en un entorno con una diversidad de sistemas operativos. La solución a esta complejidad se encuentra en la implementación de Samba, un software versátil que despliega sus funciones tanto como cliente como servidor, actuando como un intermediario entre dispositivos individuales.

Esta suite de software de código abierto implementa el protocolo de red SMB/CIFS en entornos Unix y Linux. Visualízalo como un lenguaje universal que facilita la comunicación de recursos compartidos en redes Windows. Gracias a esto, Samba se integra perfectamente en la red de distribución de Windows, ya sea desempeñando el papel de cliente o asumiendo la función de servidor.

Cuando nos referimos a un servidor Samba en un entorno Linux, pensamos en un regulador maestro del intercambio de archivos y servicios en la red, equiparándolo a la funcionalidad de un servidor de Windows. A partir de la versión 4, este servidor Samba Linux incluso asume la responsabilidad de controlar dominios a través de Microsoft Active Directory. La flexibilidad es una de sus características distintivas, permitiéndonos gestionar y configurar el servidor de forma remota desde cualquier ordenador cliente. Este proceso puede realizarse tanto desde la línea de comandos, para aquellos más orientados a la eficiencia técnica, como instalando una interfaz gráfica adicional, lo cual simplifica la administración y configuración.

Para poner en marcha Samba como servidor de archivos, es esencial seguir una serie de pasos.

#### 3.1 Paso 0: Cambiar nombre de equipo en Linux

Antes de empezar, dedicamos unas líneas a recordar como cambiar el nombre de equipo en un sistema Linux correctamente. Podemos consultar el nombre del equipo con el comando:

```
hostnamectl
```

Para cambiar el nombre del equipo, debemos usar:

```
sudo hostnamectl set-hostname nuevonombre
```

Donde “nuevonombre” es el nombre deseado del equipo. Para que el cambio sea efectivo, es necesario tras esto reiniciar el equipo. Podemos hacerlo con:

```
sudo reboot
```

#### 3.2 Paso 1: Instalación de Samba

En primer lugar, debemos instalar el servidor Samba en el sistema. Esta operación se ejecuta desde la terminal con privilegios de superusuario. Como práctica recomendada, se sugiere realizar una actualización del sistema antes de instalar paquetes, garantizando así la incorporación de las últimas versiones de cada componente.

```
sudo apt update  
sudo apt install samba -y
```

El paquete de Samba establece una configuración mínima durante la instalación inicial. Sin embargo, se aconseja encarecidamente personalizar la configuración según las necesidades específicas de la organización. Durante la instalación, se generará un archivo en “/etc/samba” llamado “smb.conf”, que contiene todas las configuraciones esenciales del servidor.

Nota: Antes de realizar cualquier modificación en el archivo de configuración, es prudente crear una copia de seguridad por si se produce algún error.

! **Atención:** antes de realizar cualquier modificación en el archivo de configuración, es prudente crear una copia de seguridad por si se produce algún error.

```
cp -p /etc/samba/smb.conf /etc/samba/smb_backup.conf
```

### 3.3 Paso 2: Compartiendo recursos en Samba

Para permitir el acceso a los recursos compartidos, es necesario añadir usuarios al sistema. Esto garantiza que solo las personas autorizadas puedan acceder a los archivos compartidos. Una vez añadidos estos usuarios, debemos otorgar los permisos adecuados a las carpetas compartidas, ya que es esencial para controlar el acceso de los usuarios. Esto se realiza mediante configuraciones específicas en el archivo “*smb.conf*” (**recuerda hacer copia de seguridad antes de manipularlo**) y asegura que la seguridad de los recursos compartidos se ajuste a los requisitos de la organización.

En el archivo “*smb.conf*”, las distintas secciones, identificadas por títulos entre corchetes, contienen parámetros que pueden activarse o desactivarse. Es importante señalar que el carácter “#” al principio de cada línea se utiliza para comentar la configuración.

El fichero define 3 secciones clave con nombre reservados. Estas son las siguientes:

- **[global]:** Define los parámetros del servidor.
  - Ahí se definen elementos como el Workgroup (nombre del grupo de trabajo) o qué interfaces de red permiten el uso de recursos compartidos.
- **[homes]:** Define las carpetas (/home) de los usuarios como recursos compartidos.
- **[printer]:** Define las impresoras compartidas.

**También podemos crear nuestras propias secciones de un recurso compartido.** Para ilustrar como crear una sección propia, veamos un ejemplo. Si deseamos compartir una carpeta llamada “nominas” de manera privada, que requiere autenticación con usuario y contraseña, la sección a agregar en “*smb.conf*” sería algo así:

```
[nominas]
comment = carpeta con contraseña
path = ruta completa del directorio (carpeta)
browseable = yes
read only = no
create mask = 0775
directory mask = 0775
```

Donde:

- **[nominas]:** es el nombre de la sección.
- **comment:** ofrece una descripción del contenido de la carpeta.
- **path:** indica la ruta completa del directorio compartido.
- **browseable:** permite la navegación por subcarpetas.
- **read only:** indica si la carpeta es solo lectura. En este caso indica que no, por lo cual, si existen suficientes permisos locales, permite leer y escribir en la carpeta.
- **create mask:** establece la máscara de permisos (permisos al crear los archivos).
- **directory mask:** establece la máscara de permisos (permisos al crear las carpetas).

! **Atención:** además de estar compartida la carpeta, los usuarios deben tener permisos locales para que funcione la carpeta compartida. Es un error muy común compartir algo y que luego no tenga acceso por no tener concedidos permisos locales.

Complementamos el ejemplo anterior, con una configuración avanzada donde que deseamos compartir una carpeta llamada "documentos" en el directorio "/home/usuario/documentos" en modo solo lectura a un grupo de usuarios concreto. Aquí tienes un ejemplo de cómo hacerlo en el archivo "smb.conf":

```
[documentos]
comment = Carpeta de documentos compartidos
path = /home/usuario/documentos
browseable = yes
read only = no
create mask = 0700
directory mask = 0700
valid users = @grupo_usuarios
```

Donde:

- **[documentos]:** Este es el nombre del recurso compartido que se utilizará para acceder a la carpeta compartida.
- **comment:** Una descripción opcional del recurso compartido.
- **path:** La ruta completa del directorio que se compartirá.
- **browseable:** Indica si el recurso compartido es visible al navegar a través de la red. En este caso, se establece en "yes".
- **read only:** indica si la carpeta es solo lectura. En este caso indica que no, por lo cual, si existen suficientes permisos locales, permite leer y escribir en la carpeta.
- **create mask:** Establece los permisos para los nuevos archivos creados en la carpeta.
- **directory mask:** Establece los permisos para los nuevos directorios creados en la carpeta.
- **valid users:** Permite definir qué usuarios o grupos de usuarios tienen acceso al recurso compartido. Puedes especificar un usuario individual o un grupo de usuarios.

### 3.4 Paso 4: Registro de usuarios

Para que un usuario exista en Samba, debe existir como usuario local en Linux y además, debe ser registrado en el entorno de Samba para poder acceder a los recursos compartidos de forma segura. Para llevar a cabo este registro, se utiliza el comando:

```
sudo smbpasswd -a usuario_samba
```

Aquí un ejemplo de código donde creamos un usuario local y luego lo añadimos a Samba:

```
sudo adduser usuario_samba
sudo smbpasswd -a usuario_samba
```

**En ocasiones, queremos que las cuentas de usuario utilizadas para acceder a recursos compartidos en Samba, por seguridad, no tengan acceso al intérprete de comandos localmente.**

Si se desea esto, no es necesario asignar una contraseña utilizando el comando *passwd* y si se quiere que no puedan loguearse localmente, se debe definir *"/sbin/nologin"* o *"/bin/false"* como el intérprete de comandos para la cuenta de usuario específica.

Un ejemplo donde creamos un usuario que no se pueda logear localmente en el sistema, pero si exista en Samba es el siguiente:

```
sudo useradd -s /sbin/nologin sambauser1
sudo smbpasswd -a sambauser1
```

**En el caso contrario, si se necesita que las cuentas puedan loguearse localmente o acceder a otros servicios como SSH**, entre otros, es decir, que tengan acceso al intérprete de comandos, es necesario especificar `"/bin/bash"` como el intérprete de comandos. Además, se debe asignar una contraseña en el sistema utilizando el comando `passwd`. Esto se realiza mediante los siguientes comandos:

```
sudo useradd -s /bin/bash sambauser2
sudo passwd sambauser2
sudo smbpasswd -a sambauser2
```

Estos pasos garantizan una correcta gestión de cuentas de usuario en el entorno Samba.

### 3.5 Paso 5: Inicio y parada del servidor Samba

Para que Samba funcione, debe estar en marcha el servicio `"smbd"`. A continuación proporcionamos comandos para gestionar el servicio Samba, permitiendo iniciar, detener, reiniciar o recargar la configuración según las necesidades específicas del entorno.

Podemos comprobar el estado del servicio con el comando:

```
sudo service smbd status
```

Para iniciar el servidor Samba, se utiliza el siguiente comando:

```
sudo service smbd start
```

Para detenerlo, se emplea:

```
sudo service smbd stop
```

Cada modificación que se realice en el servidor `"smb.conf"`, requerirá que se reinicie el servicio `"smbd"` para que esta modificación sea efectiva.

En caso de necesitar reiniciarlo, la instrucción es la siguiente:

```
sudo service smbd restart
```

Cabe destacar un comando adicional que resulta bastante interesante y práctico: el reinicio condicional. La opción `condrestart` inicia el servicio `"smbd"` solo si está en ejecución en ese momento, evitando iniciar el servicio si no está en ejecución. La sintaxis de este comando sería:

```
sudo service smbd condrestart
```

Si por alguna razón se experimenta un problema con la recarga automática del servicio `smb` y se requiere recargar manualmente el archivo `"smb.conf"`, se puede recurrir al siguiente comando:

```
sudo service smbd reload
```

### 3.6 Paso 6: Testeando configuraciones con `"testparm"`

La utilidad `"testparm"` en Samba desempeña un papel importante al garantizar la integridad y coherencia de la configuración del servidor. Este comando analiza la configuración establecida en el archivo `"smb.conf"`.

A efectos prácticos, `"testparm"` se convierte en una herramienta esencial para verificar la configuración de Samba antes de reiniciar el servicio o implementar cambios significativos en la configuración. Su función es asegurar que el archivo `"smb.conf"` esté libre de errores, garantizando así que el servidor Samba esté preparado para aplicar la configuración de manera efectiva.

Este comando realiza las siguientes funciones:

- **Detección de errores de configuración:** “*testparm*” realiza un análisis exhaustivo del archivo de configuración de Samba en busca de posibles errores de sintaxis, configuraciones incorrectas o cualquier problema que pueda afectar el funcionamiento adecuado del servidor Samba.
- **Visualización de la configuración activa:** después de analizar el archivo de configuración, “*testparm*” presenta en la salida estándar (stdout) la configuración activa que el servidor Samba utilizará. Esto abarca tanto las secciones globales como las configuraciones específicas de los recursos compartidos.
- **Validación de configuración:** la herramienta también contribuye a validar que la configuración sea coherente y siga la sintaxis y estructura previstas. En caso de problemas, “*testparm*” generalmente emite mensajes de error que facilitan la identificación y corrección de los posibles inconvenientes.

## 4. ACCEDIENDO A UNA CARPETA COMPARTIDA

### 4.1 Acceso a carpetas compartidas en sistemas Linux

Aquí comentamos como acceder y montar carpetas compartidas con Samba. **Cabe aclarar que esto permite acceder tanto a carpetas compartidas con Samba desde Linux como a carpetas compartidas de Windows.**

#### 4.1.1 Desde navegador de ficheros

Generalmente, si accedemos desde el navegador de ficheros de un sistema Linux, con una línea similar a la siguiente, accedemos al recurso compartido:

```
smb://IP_del_Servidor/Nombre_de_Carpeta_Compartida
```

#### 4.1.2 Desde consola de comandos

Asegúrate de que el paquete “*smbclient*” esté instalado en tu sistema. Puedes instalarlo con el siguiente comando, dependiendo de tu distribución:

```
sudo apt-get install smbclient -y
```

Una vez instalado, para poder acceder a una carpeta compartida con el comando “*smbclient*” reemplaza IP\_del\_Servidor (o nombre de host), Nombre\_de\_Carpeta\_Compartida y Usuario con la información correspondiente:

```
smbclient //IP_del_Servidor/Nombre_de_Carpeta_Compartida -U Usuario
```

Por ejemplo:

```
smbclient //192.168.1.1/compartido -U usuario
```

Tras ello, se te solicitará que escribas la contraseña del usuario especificado.

Una vez que estés autenticado, puedes utilizar comandos similares a los de un shell FTP para explorar y operar en la carpeta compartida:

- **ls:** Listar archivos en la carpeta compartida.
- **get:** Descargar un archivo desde la carpeta compartida.
- **put:** Subir un archivo a la carpeta compartida.
- **cd:** Cambiar de directorio en la carpeta compartida.

Por ejemplo, para listar los archivos en la carpeta compartida:

```
smb: \> ls
```



Para descargar un archivo:

```
smb: \> get nombre_archivo
```

Cuando hayas terminado de trabajar con la carpeta compartida, puedes salir de “smbclient” escribiendo “exit” en la consola.

Este método te permitirá acceder y operar en una carpeta compartida Samba directamente desde la consola de comandos en Linux. Asegúrate de tener las credenciales correctas y los permisos adecuados para acceder a la carpeta compartida.

## 4.2 Acceso a carpetas compartidas en sistemas Windows

Se accede como a cualquier otra carpeta compartida de Windows, como vimos en unidades anteriores.

## 4.3 Montaje de carpetas compartidas en sistemas Linux

Para montar una carpeta compartida Samba como una carpeta (es decir, conectar un punto de montaje a esa carpeta) realizaremos los siguientes pasos.

Asegúrate de que el paquete “*cifs-utils*” esté instalado en tu sistema. Puedes instalarlo con el siguiente comando.

```
sudo apt-get install cifs-utils -y
```

Tras ello, crea un directorio que servirá como punto de montaje para la carpeta compartida. Por ejemplo podría ser:

```
sudo mkdir /mnt/samba_share
```

Una vez creado el directorio, utiliza el comando mount para montar la carpeta compartida. Reemplaza IP\_del\_Servidor (o nombre de host), Nombre\_de\_Carpeta\_Compartida, Usuario y Contraseña con la información correspondiente.

```
sudo mount -t cifs //IP_del_Servidor/Nombre_de_Carpeta_Compartida /mnt/samba_share -o username=Usuario,password=Contraseña
```

Por ejemplo:

```
sudo mount -t cifs //192.168.0.1/compartido /mnt/samba_share -o username=usuario,password=contraseña
```

Si prefieres no incluir la contraseña directamente en el comando, puedes omitir la opción password y se te pedirá ingresarla interactivamente cuando ejecutes el comando.

```
mount | grep samba_share
```

Deberías ver una línea que indique que la carpeta compartida está montada en /mnt/samba\_share.

## 4.4 Montaje de carpetas compartidas de forma automática al iniciar un sistema Linux

Si deseas que la carpeta compartida se monte automáticamente al iniciar sesión, puedes agregar una entrada al archivo “/etc/fstab”. Más información sobre “fstab” en <https://es.wikipedia.org/wiki/Fstab>

Abre el archivo con un editor de texto de una forma similar a esta

```
sudo nano /etc/fstab
```

y agrega una línea similar a la siguiente:

```
//IP_del_Servidor/Nombre_de_Carpeta_Compartida /mnt/samba_share cifs
```

```
username=Usuario,password=Contraseña 0 0
```

Guarda y cierra el archivo. Finalmente, puedes probar el montaje automático ejecutando:

```
sudo mount -a
```

## 5. ESCRITORIO MÓVIL CON SAMBA [HOME]

### 5.1 Paso 1: Configuración en el servidor

Utilizar el recurso compartido [homes] en Samba es una forma eficiente de permitir a los usuarios acceder automáticamente a sus carpetas de inicio personales. Este recurso compartido se utiliza para crear dinámicamente un recurso compartido personalizado para cada usuario registrado en el sistema. **Esto es muy útil para implementar con Samba la técnica de “Escritorio móvil” (o perfiles móviles) técnica que sea donde sea que te logees, tendrás el mismo escritorio.**

A continuación, te proporcionaré una explicación sencilla de cómo configurar y utilizar [homes] en el archivo “*smb.conf*”.


Supongamos que tienes un usuario llamado "usuario1" en tu sistema y deseas que tenga acceso a su carpeta de inicio personalizada a través de Samba.

```
[homes]
comment = Carpetas de inicio
browseable = no
read only = no
create mask = 0700
directory mask = 0700
valid users = %S
```

Donde:

- **[homes]:** este es el nombre del recurso compartido especial para las carpetas de inicio de los usuarios.
- **comment:** una descripción opcional del recurso compartido.
- **browseable:** se establece en "no" para evitar que los usuarios vean la lista de carpetas de inicio disponibles al navegar por la red.
- **read only:** establecido en "no" para permitir la escritura en la carpeta de inicio.
- **create mask:** define los permisos para los nuevos archivos creados en la carpeta de inicio.
- **directory mask:** define los permisos para los nuevos directorios creados en la carpeta de inicio.
- **valid users:** permite acceder solo al usuario al que pertenece el recurso compartido (%S representa el nombre del usuario).

Configurando [homes] de esta manera, cuando el “usuario1” inicie sesión y acceda al recurso compartido de red de Samba, automáticamente se le presentará su propia carpeta de inicio. Además, el “usuario1” tendrá los permisos necesarios para leer y escribir en su carpeta de inicio, y otros usuarios no tendrán acceso a ella.

 **Importante:** es importante tener en cuenta que [homes] es un recurso compartido dinámico, y Samba lo gestionará automáticamente en función de los usuarios registrados en el sistema.

### 5.2 Paso 2: Configuración en los clientes Linux

Para configurar el “Escritorio móvil” (o perfiles móviles) simplemente debemos hacer que se monte automáticamente el recurso “homes” en la carpeta “/home”.

## 6. CUOTAS DE DISCO EN LINUX

Las cuotas de disco son una herramienta esencial para gestionar el espacio de almacenamiento en sistemas operativos, desempeñando un papel crucial al permitir a los administradores de sistemas controlar y limitar la cantidad de información que los usuarios pueden almacenar en un volumen específico, generalmente asociado a una partición. Este control es esencial para garantizar un uso equitativo de los recursos y evitar que un usuario o grupo de usuarios monopolice el espacio disponible.

En el contexto de Linux, las cuotas son una característica comúnmente integrada en el núcleo (kernel) del sistema operativo. Este núcleo incorpora la funcionalidad necesaria para gestionar las cuotas de manera eficiente. Sin embargo, en casos donde esta capacidad no esté habilitada por defecto, es posible instalar los paquetes "quota" y "quotatool" para habilitar y administrar estas funcionalidades. Podemos hacerlo con

```
sudo apt install quota quotatool -y
```

Estos paquetes brindan las herramientas necesarias para establecer límites de almacenamiento para usuarios individuales o grupos, asegurando así un control del uso del espacio en disco.

### 6.1 Ejemplos de comandos para establecer cuotas

A continuación mostramos algunos ejemplos prácticos de cómo se pueden utilizar las cuotas de disco en sistemas Linux para gestionar eficientemente el espacio de almacenamiento:

#### Asignar Cuotas a Usuarios Individuales:

Supongamos que en un servidor compartido, se desea limitar el espacio de almacenamiento para un usuario específico llamado "usuario1". Puedes establecer una cuota para este usuario de, por ejemplo, 5 GB. Esto se logra mediante el siguiente comando:

```
sudo setquota -u usuario1 0 5242880 0 0 /ruta/del/volumen
```

En este comando, "5242880" representa el límite en kilobytes, es decir, 5 GB. El último parámetro ("/ruta/del/volumen") indica la ruta al volumen en el que se aplicará la cuota.

#### Asignar Cuotas a Grupos:

Si prefieres gestionar el espacio por grupos, puedes establecer cuotas para un grupo específico. Supongamos que tienes un grupo llamado "grupo1" al que deseas asignar un límite de 10 GB. Puedes hacerlo con el siguiente comando:

```
sudo setquota -g grupo1 0 10485760 0 0 /ruta/del/volumen
```

Aquí, "10485760" representa 10 GB en kilobytes.

#### Consultar cuotas existentes:

Puedes verificar las cuotas existentes utilizando el comando repquota. Por ejemplo:

```
repquota /ruta/del/volumen
```

Este comando proporcionará información detallada sobre las cuotas establecidas en el volumen especificado.

#### Ajustar cuotas existentes:

Supongamos que deseas aumentar la cuota de "usuario1" a 8 GB. Puedes realizar este ajuste con el siguiente comando:

```
sudo setquota -u usuario1 0 8388608 0 0 /ruta/del/volumen
```

Aquí, "8388608" representa 8 GB en kilobytes.

### Eliminar Cuotas:

En caso de que desees eliminar una cuota existente para un usuario o grupo, puedes utilizar el siguiente comando:

```
sudo setquota -u usuario1 0 0 0 0 /ruta/del/volumen
```

Este comando establece todos los valores de cuota a cero, efectivamente eliminando cualquier restricción.

## 7. SERVIDOR DE IMPRESIÓN CUPS

El servidor de impresión CUPS (Common Unix Printing System, en castellano Sistema de Impresión Común para Unix) es un sistema para administrar impresoras en entornos Linux.

CUPS que facilita la vida de los usuarios al permitirles imprimir en impresoras locales o de red de manera sencilla y eficiente. Además, es totalmente gratuito y se ha convertido en el estándar de facto para la impresión en la mayoría de las distribuciones de Linux.

Si estás utilizando una versión actual de Ubuntu, es probable que CUPS ya esté instalado y listo para funcionar, pero si no es así, los pasos para instalarlo son los siguientes:

```
sudo apt update
sudo apt install cups -y
```

Tras ello, para que tu usuario pueda administrar impresoras, haz que este sea parte del grupo "lpadmin" con el comando:

```
sudo usermod -aG lpadmin tu_usuario
```

Una vez hecho esto, debemos iniciar el servicio CUPS:

```
sudo systemctl start cups
```

Si quieres habilitar el servicio para que arranque automáticamente al iniciar el sistema, puedes hacer con el siguiente comando:

```
sudo systemctl enable cups
```

Puedes verificar el estado del servicio:

```
sudo systemctl status cups
```

Para tener un control total sobre CUPS, puedes acceder a su interfaz web, que es como el cuartel general para administrar todo relacionado con la impresión en tu sistema. Abre tu navegador web y visita: <http://localhost:631> y desde aquí, podrás configurar tus impresoras, administrar trabajos de impresión y realizar ajustes adicionales.

Recuerda que todos los archivos de configuración de CUPS están guardados en la carpeta "/etc/cups". Aunque casi nunca necesitarás tocar el archivo "/etc/cups/cupsd.conf", ya que generalmente la configuración se realizará principalmente a través de la interfaz web.

## 8. INSTALAR IMPRESORA PDF EN CUPS

Instalar una impresora PDF en CUPS es bastante sencillo. Aquí tienes los pasos para hacerlo:

```
sudo apt-get install cups-pdf
```

Una vez instalada, puedes configurar la impresora PDF. Abre tu navegador web y accede a la interfaz web de CUPS ingresando la siguiente dirección <http://localhost:631>

Dirígete a la pestaña "Administración" y selecciona "Agregar impresora". Teclea tus credenciales de administrador. En la página siguiente, busca y selecciona la impresora PDF que acabas de instalar. Puede aparecer como "CUPS-PDF" u otro nombre similar.

Sigue las instrucciones para completar la configuración. Puedes optar por establecerla como la impresora predeterminada si lo deseas.

Una vez configurada, puedes imprimir cualquier documento como un archivo PDF. Al seleccionar la impresora PDF como destino al imprimir, el sistema generará un archivo PDF en lugar de enviar el trabajo de impresión a una impresora física. Asegúrate de seleccionar la impresora PDF al imprimir y, en lugar de imprimir físicamente, se generará un archivo PDF en una ubicación específica.

Por defecto, los archivos PDF generados se guardan en el directorio `"/home/usuario/PDF"`.

## 9. COMPARTIENDO IMPRESORAS INSTALADAS EN CUPS CON SAMBA

Para compartir una impresora con Samba, deberemos modificar el fichero `"smb.conf"` con la sección **[printers]**. Podemos modificar el fichero con:

```
sudo nano /etc/samba/smb.conf
```

Supongamos hemos instalado una impresora PDF (valdría con cualquier otro tipo de impresora) y la queremos agregar a la sección **[printers]** en la configuración de Samba.

Para ello realizaremos los siguientes pasos en el archivo `"smb.conf"`, agregando las siguientes líneas al final de la sección **[printers]**:

```
[printers]
    comment = Todas las Impresoras
    browseable = yes
    path = /var/spool/samba
    printable = yes
    guest ok = yes
    read only = yes
    create mask = 0700
```

Asegúrate de que estas líneas están dentro de la sección **[printers]**. Esto permitirá que la impresora sea detectada por los clientes. Una vez hechos los cambios, podemos probar la configuración con:

```
testparm
```

Y para hacerla finalmente efectiva, deberemos reiniciar el servicio `"smb"`:

```
sudo service smbd restart
```

Con estos pasos, la impresora PDF debería estar disponible para ser detectada automáticamente desde clientes de Windows cuando exploran la red. Los usuarios podrán instalar la impresora sin tener que especificar manualmente la ruta.

## 10. MONTAR IMPRESORA COMPARTIDA POR SAMBA EN EL CLIENTE

Asegúrate de tener instalado el cliente Samba en tu sistema. Puedes instalarlo utilizando:

```
sudo apt-get install smbclient
```

Necesitarás conocer la ruta completa de la impresora compartida en el servidor Samba.

Utiliza el comando “*smbclient*” para montar la impresora compartida. Sustituye “*ruta\_impresora*” con la ruta completa de la impresora en el servidor y “*nombre\_impresora*” con el nombre de la impresora. Por ejemplo:

```
smbclient //servidor/ruta_impresora -U usuario_samba
```

Donde:

- **servidor**: la dirección IP o el nombre del servidor Samba.
- **ruta\_impresora**: la ruta completa de la impresora compartida.
- **usuario\_samba**: el nombre de usuario de Samba con los permisos adecuados.

Una vez que hayas montado la impresora, puedes verificar que se ha montado correctamente utilizando el siguiente comando:

```
lpstat -p
```

Esto debería mostrar la impresora que acabas de montar.

## 11. HERRAMIENTAS GRÁFICAS PARA GESTIONAR SAMBA

En esta unidad hemos comentado como gestionar elementos de Samba vía comandos. Pese a ello, existen distintas herramientas gráficas que pueden ayudarnos en su configuración. Algunas de estas herramientas son:

- **Módulo Samba de “Webmin”**: la herramienta “Webmin” es una herramienta para gestionar servicios de Linux que puede ser ampliando mediante módulos. Entre ellos, tiene un módulo para gestionar Samba.
  - El sitio web de “Webmin” es <https://webmin.com/>. Una vez instalado, podéis acceder a “Webmin” vía <https://localhost:10000>.
- **system-config-samba**: sistema para gestionar Samba con interfaz gráfica. Se puede instalar simplemente con “`sudo apt-get install system-config-samba -y`”.
- **Gadmin**: sistema para gestionar Samba con interfaz gráfica. Se puede instalar simplemente con “`sudo apt-get install gadmin-samba -y`”.
- **SWAT**: sistema para gestionar Samba con interfaz gráfica. Se puede instalar simplemente con “`sudo apt-get install swat -y`”.

## 12. BIBLIOGRAFÍA

<https://blog.desdelinux.net/samba-smbclient/>

<https://www.educatica.es/informatica/servicios-en-red/compartiendo-recursos-en-entornos-heterogeneos-smb-cifs-samba/>

<https://somebooks.es/configuracion-avanzada-de-samba-en-ubuntu-18-04-lts/>

<https://somebooks.es/instalar-una-impresora-virtual-en-ubuntu-22-04-lts/>