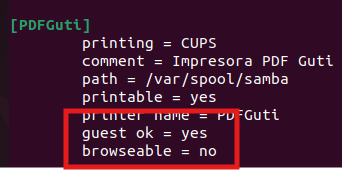
**Unidad Didáctica 6. Compartición de recursos en red mediante el protocolo CIFS/SAMBA**

**Ejercicio 1. Configura una impresora con el acceso anónimo habilitado y con la opción navegable a no. Comprueba estos cambios en Ubuntu y Windows, dejando constancia de ello.**

Comenzamos editando el archivo de configuración /etc/samba/smb.conf y agrega o modifica la siguiente sección:

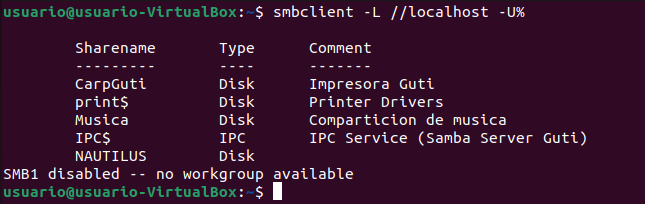


* guest ok = yes → Permite el acceso anónimo
* browseable = no → Evita que la impresora aparezca en la lista de recursos compartidos.

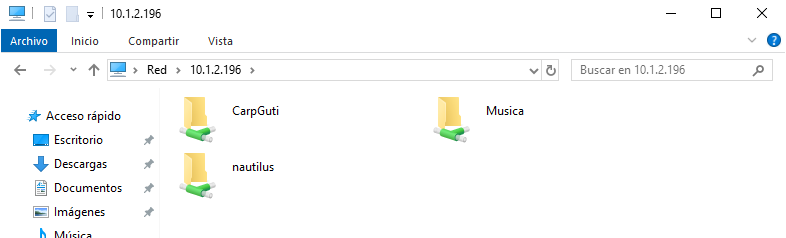
Una vez añadas las líneas, reiniciamos el servicio smbd y, a continuación, asegúrate de que la impresora esté configurada y habilitada para compartir



Para verificar usaremos el siguiente comando. Si la configuración es correcta, la impresora no debería aparecer en la lista de recursos compartidos.



Para hacer la comprobación desde Windows buscaremos la impresora a través de la ruta de red en el explorador de archivos

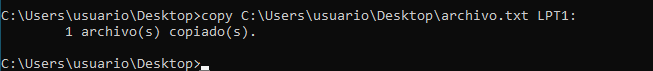


Como podemos ver, a través de la ruta especificada no se encuentra la impresora PDFGuti, lo cual significa que hemos realizado correctamente este apartado, no obstante queda comprobar que podemos imprimir archivos, para ello accedemos CMD de Windows y aplicamos el siguiente comando:

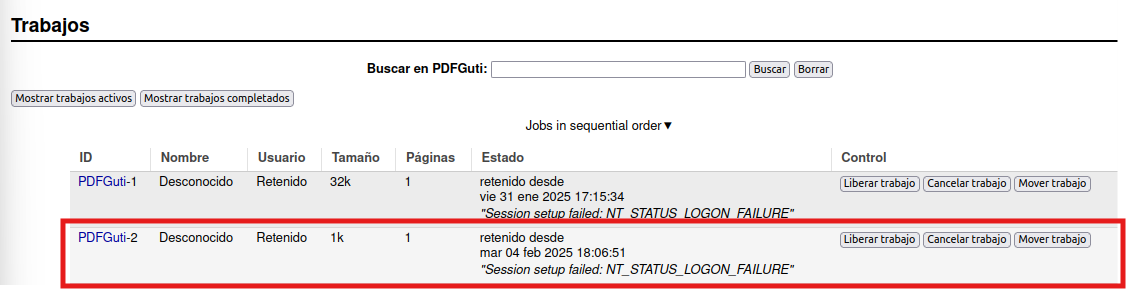


Continuaremos creando un archivo txt y compartiendolo con la impresora:



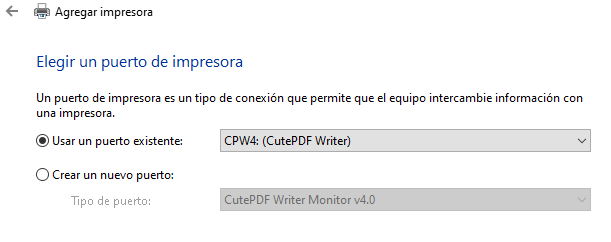


Con esto comprobamos que la impresora permite impresión sin autenticación y para su comprobación, podemos ver desde CUPS cómo ha recibido el trabajo correspondiente

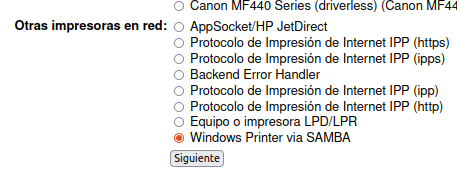


**Ejercicio 2. Tras crear una impresora en Windows, configurarla para que pueda ser compartida. Finalizada la configuración, comparte la impresora desde CUPS utilizando una conexión de tipo SMB.**

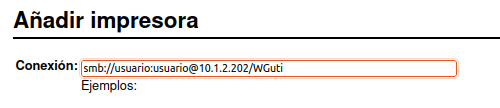
A esta impresora la llamaremos WGuti.

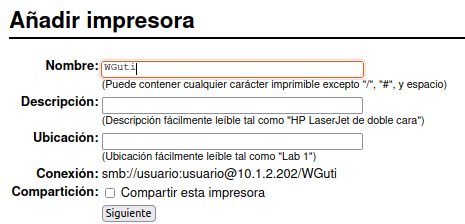


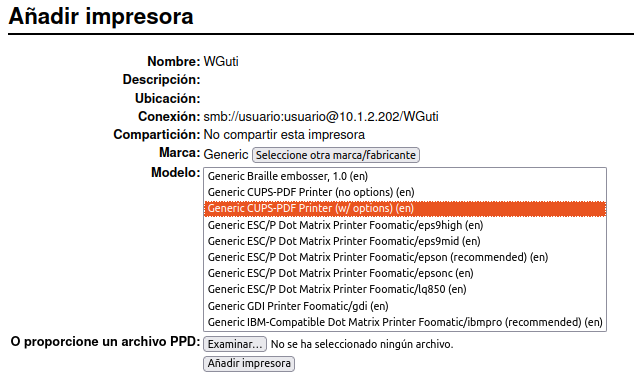
Una vez creada la impresora, entramos en CUPS para agregarla via SMB siguiendo los siguientes pasos:

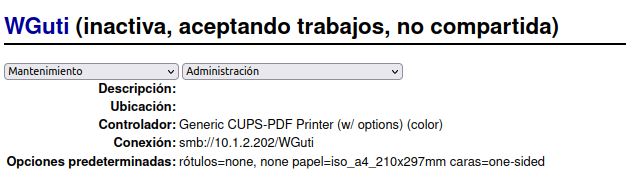


Agregamos la ruta correspondiente a la impresora creada en Windows





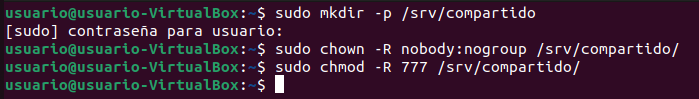




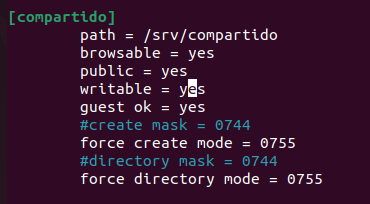
Con esto, la impresora de Windows estará disponible en Linux a través de CUPS utilizando SMB

**Ejercicio 3. Configura un directorio compartido con los parámetros de seguridad “create mask”, “force create mode”, “directory mask” y “force directory mode”. Comprueba el comportamiento del recurso desde los sistemas GNU/Linux y Microsoft Windows.**

Para el siguiente ejercicio comenzamos creando un nuevo directorio y aplicando los permisos necesarios

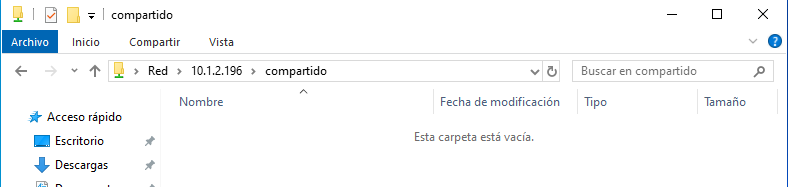


Una vez creado el directorio, editamos el archivo de configuración de SAMBA para agregar las siguientes líneas:



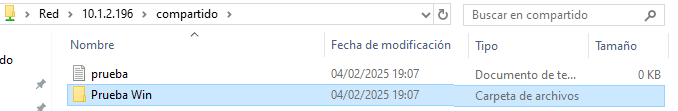
Por ultimo, guardamos el archivo y reiniciamos el servicio smbd

Para acceder al recurso desde Windows, abriremos el explorador de archivos y aplicaremos la ruta correspondiente a la carpeta “compartido”



Para comprobar el funcionamiento en Windows crearemos un archivo en la carpeta compartida y comprobaremos que otros usuarios puedan verlo con los permisos esperados.

Además crearemos una carpeta para verificar que los permisos sean 0755

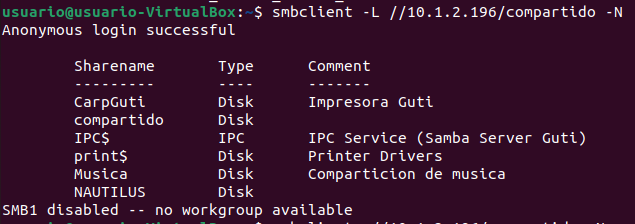


Para comprobar los permisos aplicados, entraremos en la ruta de la carpeta y haremos ls -l



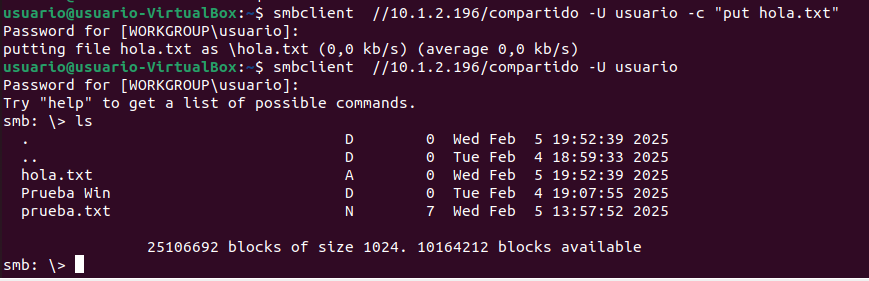
**DESDE OTRA MÁQUINA LINUX (cliente)**

Para acceder al recurso compartido aplicaremos el siguiente comando en la máquina cliente



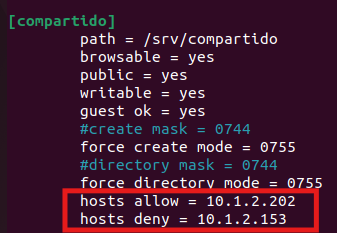
Con este comando listamos los recursos posibles

Para verificar el comportamiento desde Linux comenzaremos creando un archivo de prueba y verificando los permisos aplicados directamente

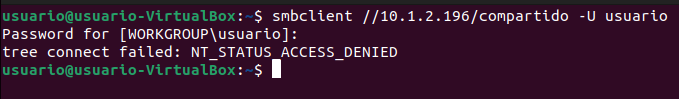


**Ejercicio 4. Limita el acceso a un directorio utilizando los parámetros “hosts allow” y “hosts deny”. Chequea el resultado desde Ubuntu y Windows. Repite lo anterior, aunque esta vez utilízalo con el elemento IPC$.**

A continuación añadiremos las siguientes líneas de configuración en el fichero de configuración de samba (smb.conf)

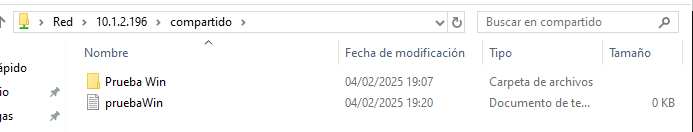


Estas líneas permiten el acceso a la máquina Windows (10.1.2.202) y denegará el acceso a la máquina Linux (10.1.2.153). Una vez hecho esto, reiniciamos el servicio smbd y entramos en el cliente Linux para probar la conexión con el siguiente comando:



Como podemos ver, nos encontramos con el mensaje de error de acceso

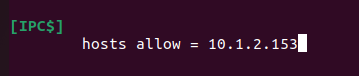
Sin embargo, en Windows, como hemos aplicado el host de acceso podemos acceder a la carpeta “compartido” sin ningún problema



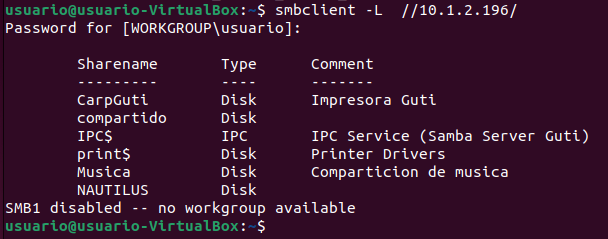
**IPC$**

Para realizar este ejercicio con IPC$ seguiremos los siguientes pasos:

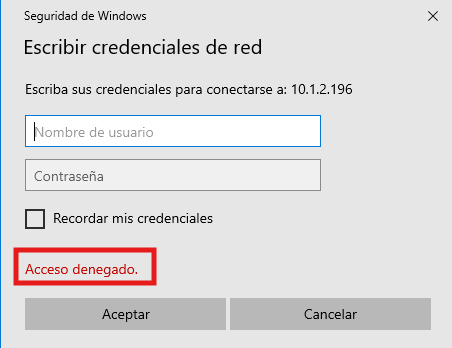
Editaremos el archivo de configuración smb.conf añadiendo la siguientes líneas:



Como hemos permitido la dirección de Linux, podemos ver que accediendo desde esta máquina si tenemos acceso.



En caso de denegar el acceso en Windows nos encontraríamos con:

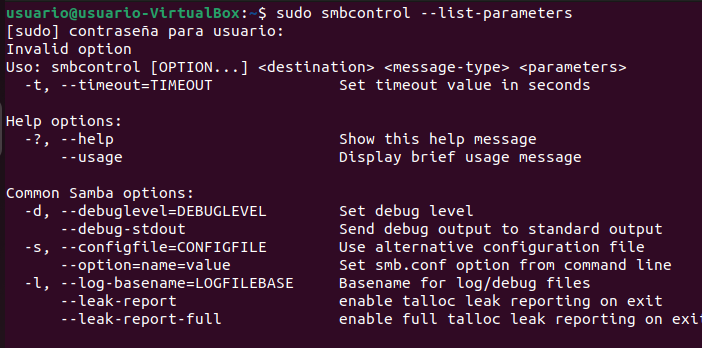


**Ejercicio 5. Investiga sobre smbcontrol y smbstatus. Práctica lo ampliado.**

El comando smbcontrol permite enviar señales y mensajes a procesos de Samba en ejecución. Es útil para administrar y depurar conexiones sin necesidad de reiniciar los servicios.

Su sintaxis general consiste en: **smbcontrol <destino> <comando> [parámetro]**

1. *Listar procesos disponibles*



1. *Enviar una señal a todos los procesos smbd*



1. *Cerrar todas las conexiones activas*



1. *Cambiar el nivel de depuración*



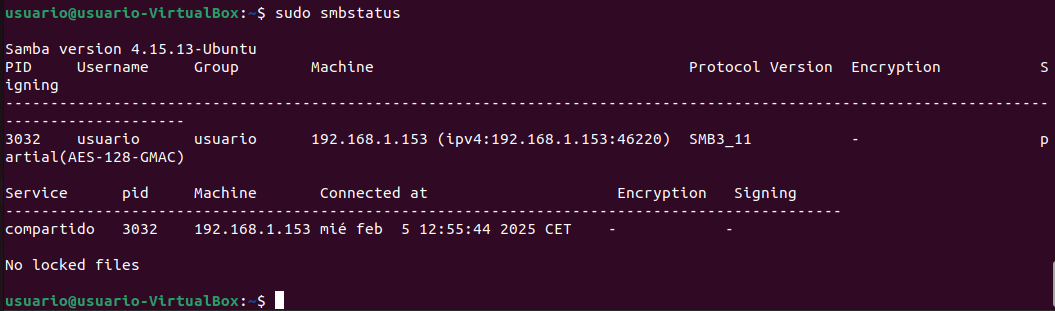
1. *Recargar la configuración de Samba sin reiniciar*



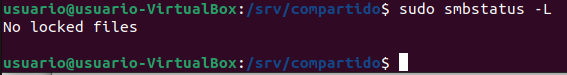
El comando smbstatus permite ver el estado actual de las conexiones Samba, incluyendo sesiones activas, archivos abiertos y bloqueos.

Su sintaxis general consiste en: **smbstatus [opciones]**

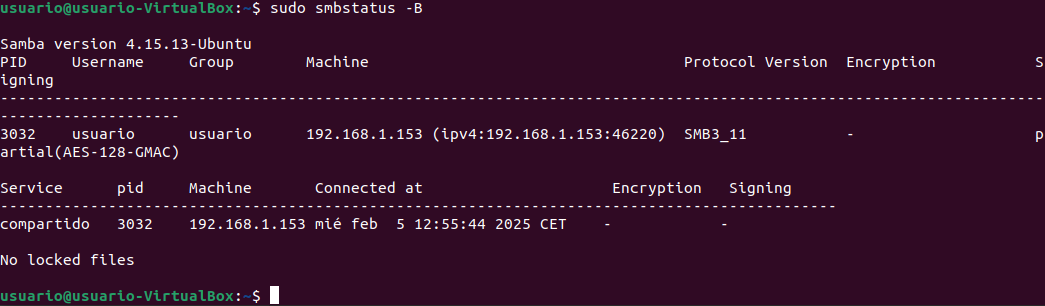
1. *Ver todas las conexiones activas*



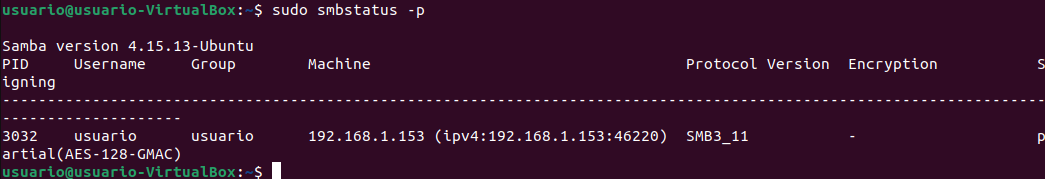
1. *Ver los archivos abiertos por los usuarios*



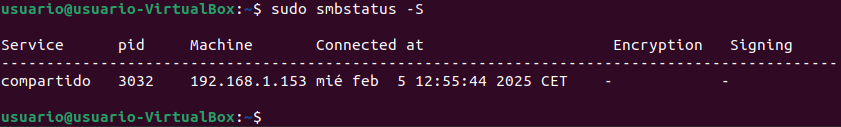
1. *Ver los bloqueos en los archivos compartidos*



1. *Ver las conexiones con detalles de PID*



1. *Ver información sobre los bloqueos en ficheros*

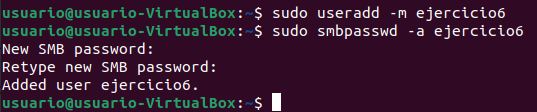


**Ejercicio 6. Busca información sobre smbpasswd y pdbedit. Práctica lo estudiado.**

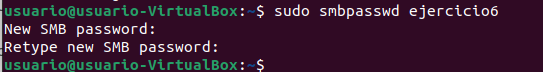
Smbd es una herramienta para gestionar contraseñas de usuarios en Samba. Se usa para:

* Añadir, modificar y eliminar contraseñas de usuarios Samba.
* Habilitar o deshabilitar usuarios.
* Sincronizar usuarios de Linux con Samba

1. *Añadir un usuario a Samba*



1. *Modificar la contraseña de un usuario*



1. *Eliminar un usuario de Samba*



1. *Deshabilitar un usuario (sin eliminarlo)*



1. *Se puede reactivarla después con el siguiente comando:*

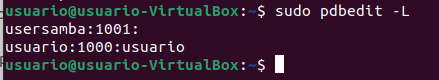


**PDBEDIT**

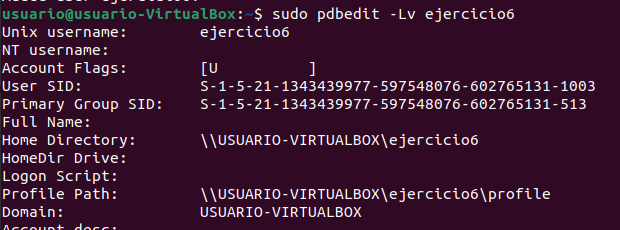
Pdbedit es otra herramienta para administrar la base de datos de usuarios de Samba que permite:

* Listar, modificar y eliminar usuarios Samba.
* Ver detalles de los usuarios.
* Cambiar información como el perfil o directorio de inicio.

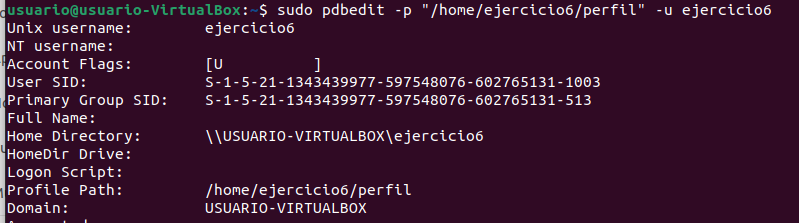
1. *Listar todos los usuarios de Samba*



1. *Ver información detallada de un usuario*



1. *Modificar el perfil de un usuario*



1. *Eliminar un usuario de Samba*

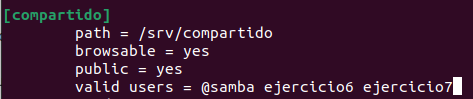


**Ejercicio 7. Investiga sobre los parámetros de SAMBA “valid users” y “read list”. Tras indagar en su uso, realiza pruebas para comprobar su comportamiento asociado.**

Samba usa un archivo de configuración para definir permisos y accesos a los recursos compartidos. Dos parámetros importantes en este archivo son valid users y read list, los cuales controlan quién puede acceder a las carpetas compartidas y con qué permisos.

1. **Parámetro valid users**

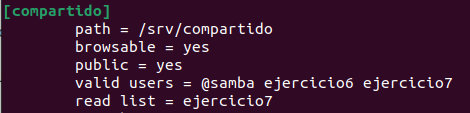
Este parámetro restringe el acceso a una carpeta compartida solo a ciertos usuarios o grupos.



Con esta linea solo los usuarios en el grupo samba y ejercicio6 pueden acceder a la carpeta

1. **Parámetro read list**

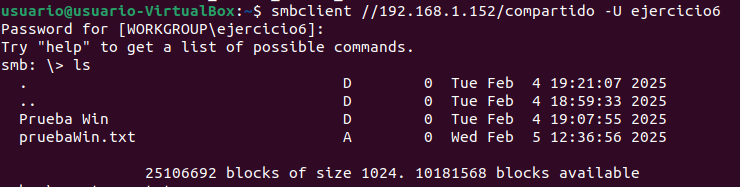
Este parámetro permite que ciertos usuarios o grupos solo tengan permisos de lectura, incluso si read only = no.



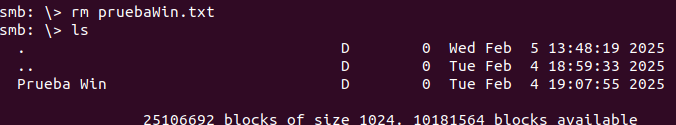
Con esta linea ejercicio6 puede leer y escribir, pero ejercicio7 solo puede leer.

Para comprobar este ejercicio accedemos desde el cliente smb en otra máquina cliente

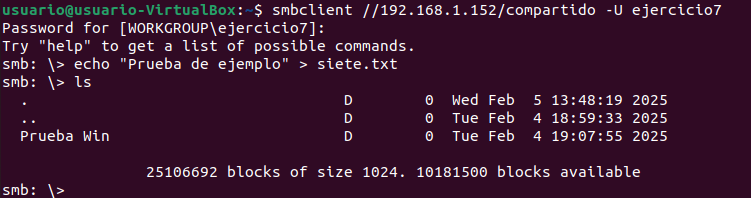
Comenzamos entrando en ejercicio6:



Con este usuario podemos eliminar un archivo de prueba, como por ejemplo pruebaWin.txt



A continuación, cambiamos de usuario a ejercicio7 para comprobar los permisos



Como podemos ver, aun creando un nuevo archivo, no podemos crearlo y si intentamos borrar el archivo prueba.txt nos encontramos con:

