
CONFIGURACIÓN DE SISTEMAS OPERATIVOS
CONFIGURACIÓN DE SISTEMAS OPERATIVOS
user@dam-server:~$

Índice General

Carlos Guaraca Nagua **IES Infanta Elena**

Práctica: Administración y Configuración de Sistemas Operativos

Curso: Grado Superior DAM
Asignatura: Sistemas Informáticos
Tema: Administración de Sistemas Operativos (Fedora, Linux Mint, OpenSUSE)

Introducción

En esta práctica, los alumnos investigarán y aplicarán de manera práctica los conocimientos necesarios para administrar y configurar sistemas operativos. Utilizarán Fedora, Linux Mint y OpenSUSE como entornos para realizar las tareas propuestas.

Cada apartado describe una tarea a realizar con una explicación clara de los objetivos, y se incluye la solución explicada paso a paso para los tres sistemas operativos.

---

1. Configuración de Usuarios y Grupos Locales

Tarea: Crear dos usuarios, uno con privilegios administrativos y otro sin ellos. Además, deben crear un grupo y asignar ambos usuarios a dicho grupo.

Explicación: Los usuarios son fundamentales para administrar el acceso al sistema. Un usuario con privilegios administrativos tiene permisos elevados para realizar configuraciones críticas, mientras que un usuario estándar tiene permisos limitados. Los grupos permiten gestionar permisos colectivos para varios usuarios. El objetivo es garantizar una estructura básica de administración y organización de cuentas.

---

2. Seguridad de Cuentas de Usuario

Tarea: Configurar una política de bloqueo tras 3 intentos fallidos de inicio de sesión.

Explicación: Configurar políticas de seguridad evita accesos no autorizados al sistema. Esta tarea requiere establecer un límite en los intentos fallidos de inicio de sesión, lo que protege contra ataques de fuerza bruta. Al bloquear temporalmente la cuenta después de varios intentos fallidos, se refuerza la seguridad del sistema.

---

3. Seguridad de Contraseñas

Tarea: Configurar una política para que las contraseñas de los usuarios caduquen

cada 30 días.

Explicación: La renovación periódica de contraseñas minimiza riesgos de seguridad en caso de que estas sean comprometidas. Esta tarea implica forzar a los usuarios a cambiar sus contraseñas regularmente, lo que mejora la seguridad general del sistema.

---

4. Gestión del Entorno de Trabajo del Usuario

Tarea: Personalizar el shell de un usuario añadiendo un alias para comandos frecuentes y configurando una variable de entorno adicional.

Explicación: La personalización del shell permite optimizar el entorno de trabajo de los usuarios, facilitando el acceso rápido a comandos habituales y a rutas específicas del sistema.

---

5. Acceso a Recursos y Permisos Locales

Tarea: Configurar permisos de lectura y escritura para un archivo accesible únicamente por un grupo específico.

Explicación: Gestionar permisos asegura que solo los usuarios autorizados puedan acceder o modificar ciertos archivos. Asignar permisos por grupo permite gestionar eficientemente el acceso para varios usuarios.

Tarea 2: Configurar dos usuarios nuevos, uno solo podrá acceder a su carpeta de usuario mientras que el otro podrá acceder a la suya y a la del otro usuario.

Explicación: Gestionar el encapsulamiento de un usuario es muy útil cuando el mismo equipo puede ser utilizado por varios usuarios y no queremos que alguno de ellos tenga los permisos necesarios para acceder a los datos de todo el sistema.

---

6. Configuración de la Impresión

Tarea: Configurar un servidor de impresión utilizando CUPS y añadir una impresora.

Explicación: CUPS es una herramienta estándar para gestionar impresoras en sistemas operativos Linux. Esta tarea implica instalar, habilitar y usar este sistema para gestionar impresoras locales o de red.

---

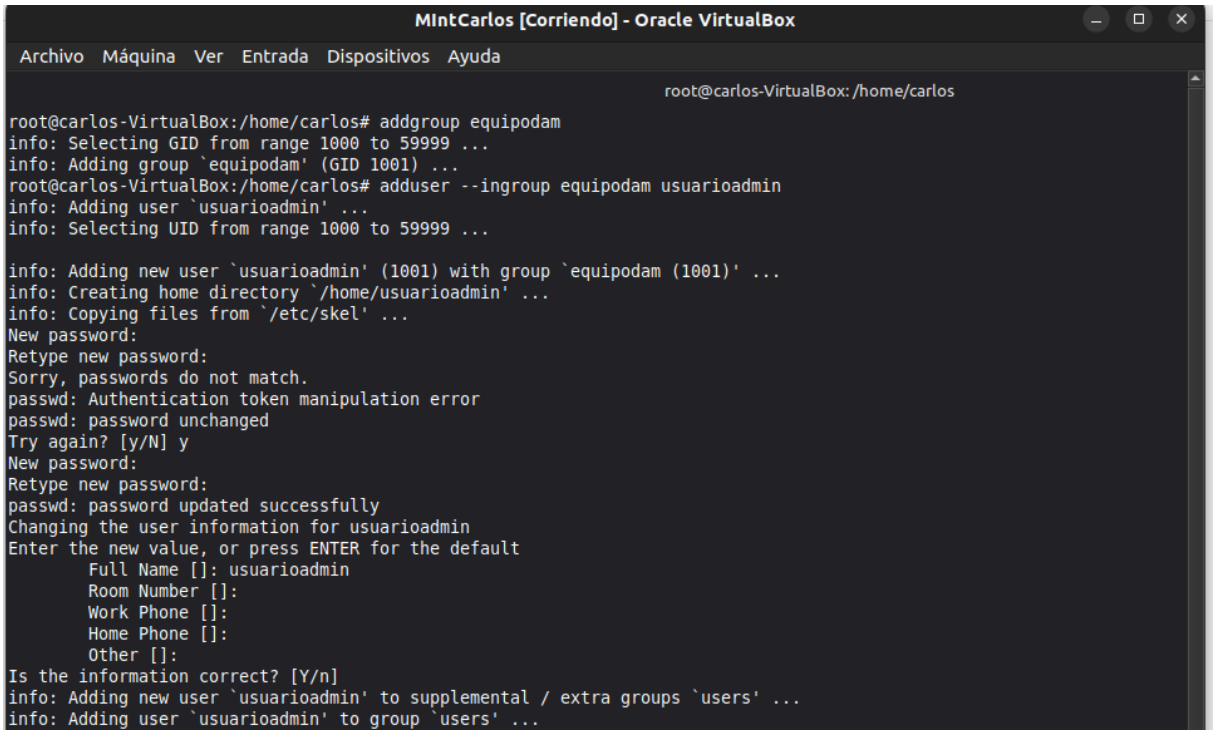
7. Programación Básica de Shell Script

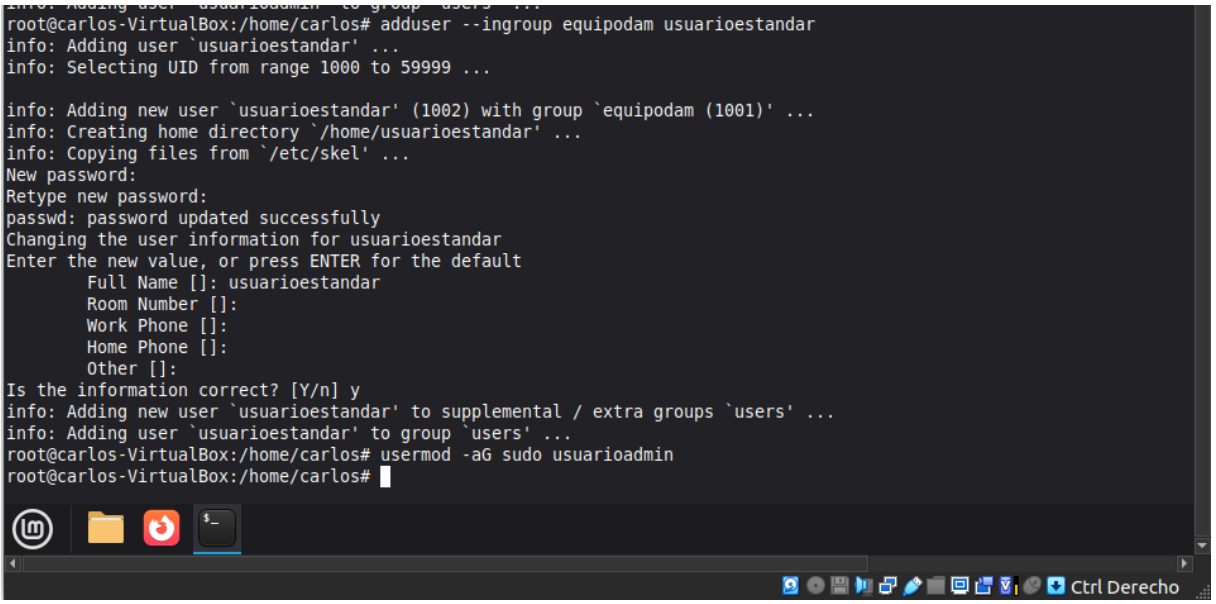
Tarea: Crear un script que realice una copia de seguridad de un directorio especificado.

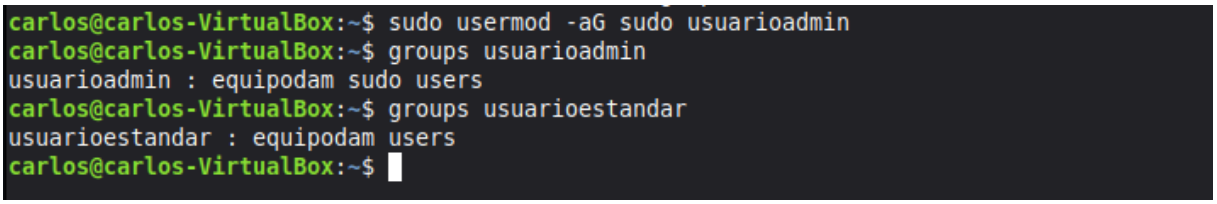
Explicación: Los scripts de shell permiten automatizar tareas administrativas repetitivas. Este ejercicio ayuda a comprender cómo escribir scripts básicos para manejar archivos y realizar copias de seguridad.

Linux Mint

Paso 1. Configuración de Usuarios y Grupos Locales







Carlos Guaraca Nagua **IES Infanta Elena**

Como usuarios **root** primeramente creamos el grupo y dos usuarios (**usuarioadmin** y **usuarioestandar**) los dos los metemos en el grupo "**equipodam**", al primero le asignamos privilegios y verificamos que pertenecen a ese grupo.

Comandos (si somos usuarios root no hace falta usar "sudo"):

 # 1. Crear el grupo compartido (como root)

sudo addgroup equipo_dam

 # 2. Crea usuarios y los asigna a un grupo (equipodam)

sudo adduser --ingroup "equipo_dam" "usuarioestandar/usuarioadmin"

 # 3. Permite dar privilegios a un usuario añadiéndolo al grupo "sudo"

sudo usermod -aG sudo usuarioadmin

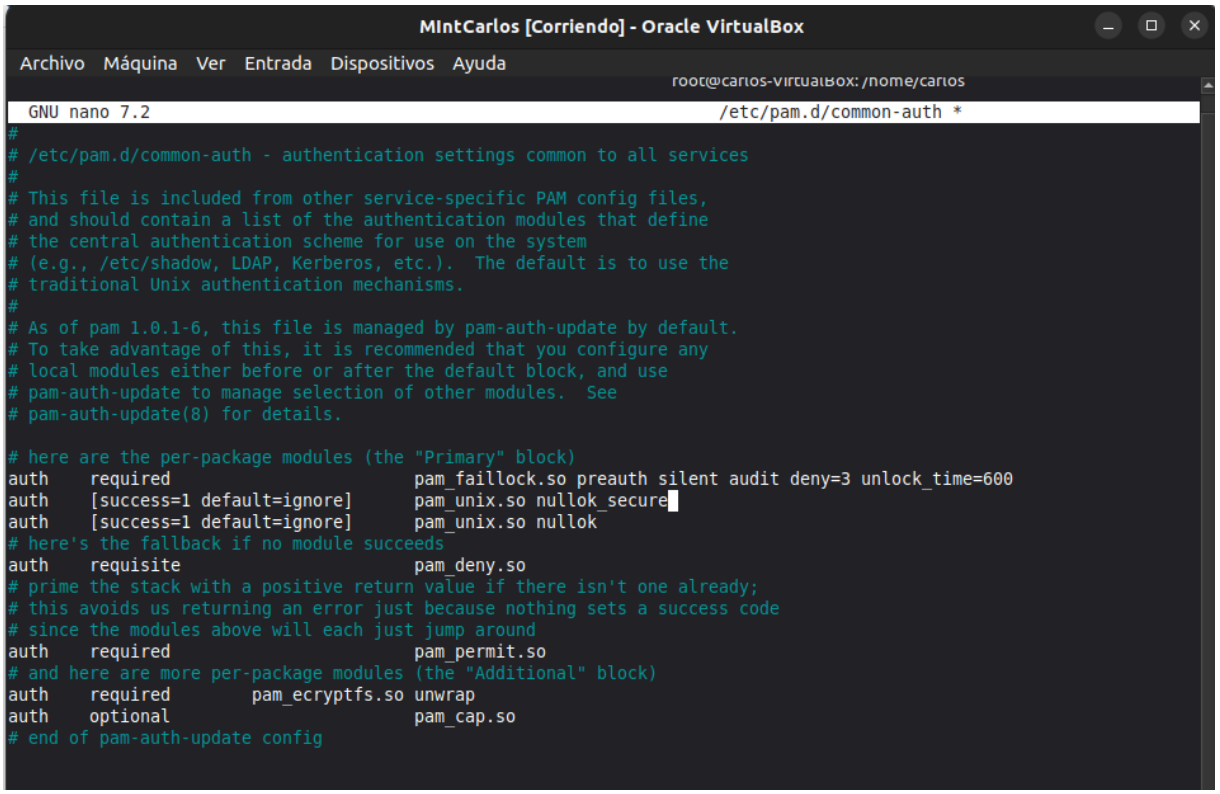
 # 4. Verifica que pertenece a un grupo

groups "usuarioestandar/usuarioadmin"

Paso 2. Seguridad de Cuentas de Usuario (utilizaremos el comando "nano" para acceder a estos archivos)

Para poder modificar la política de bloque debemos acceder a los archivo PAM.

Primeramente modificaremos el archivo **/etc/pam.d/common-auth**:



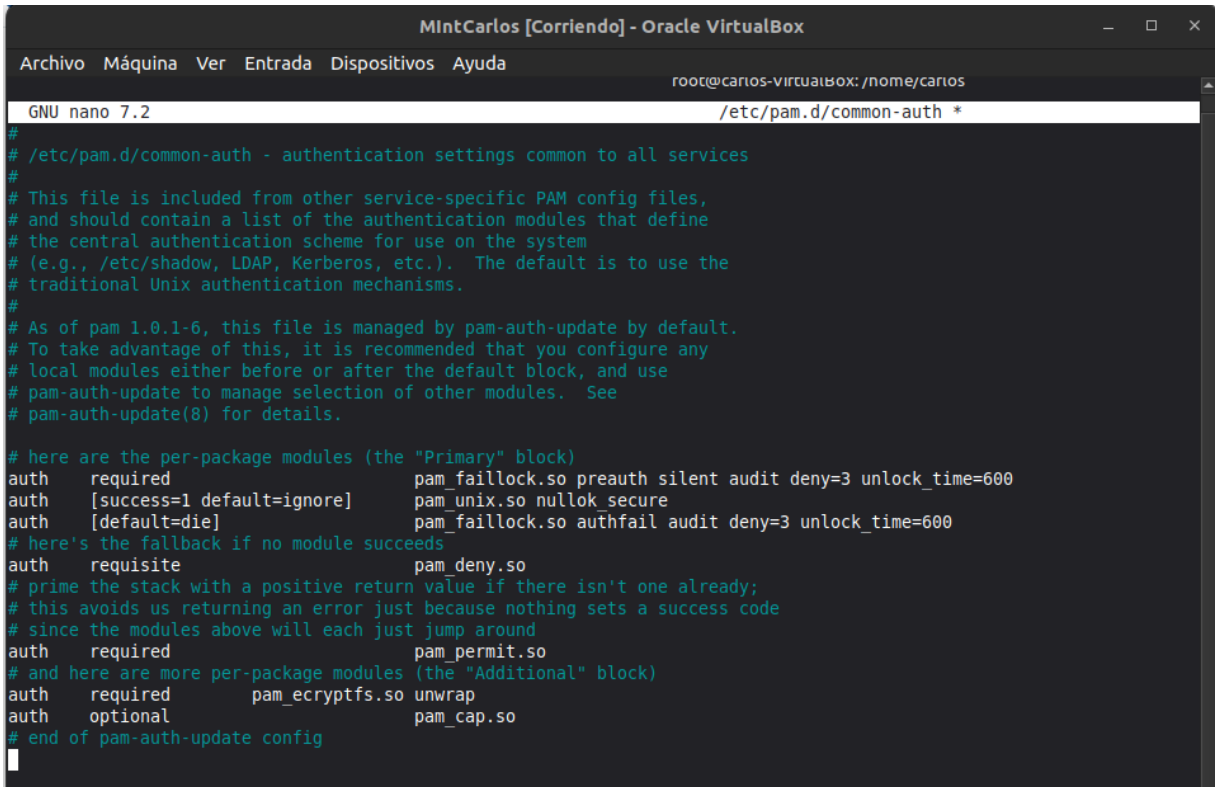
Debemos añadir las líneas de **pam_faillock.so** que bloquea y cuenta los intentos:

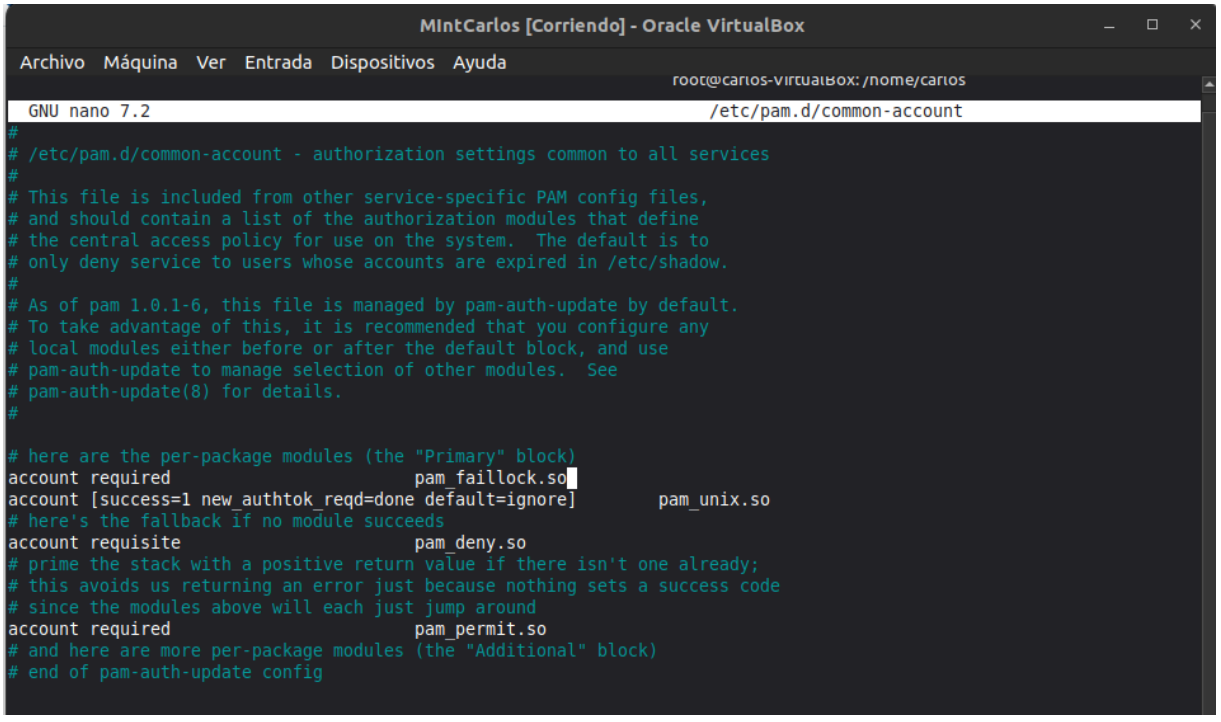
auth required pam_faillock.so preauth silent audit deny=3 unlock_time=600 auth [success=1 default=ignore]

pam_unix.so nullok_secure

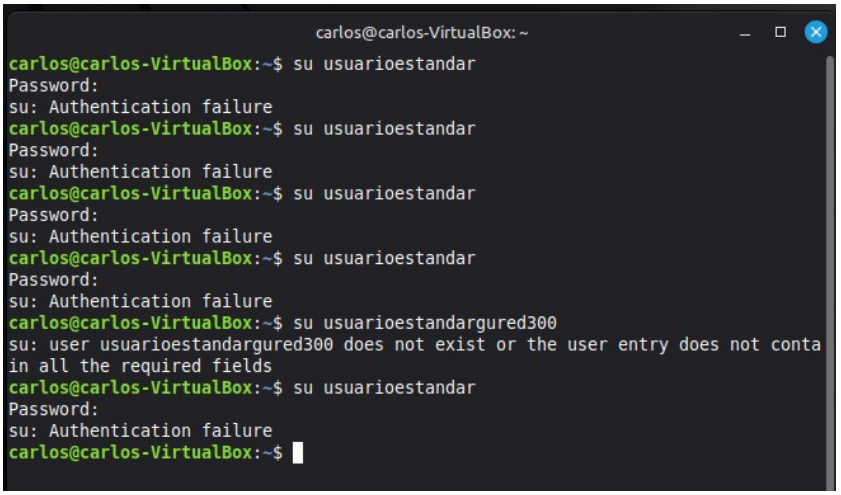
Estas líneas deben ponerse al principio de la sección **auth** (Sin embargo esto no era lo único que hacer, ya que ahora hay un conflicto con la línea **pam_unix.so nullok**). Para poder proseguir con la tarea y que funcione tuve que eliminar esa línea e insertar la línea **auth [default=die] pam_faillock.so authfail audit deny=3 unlock_time=600**.

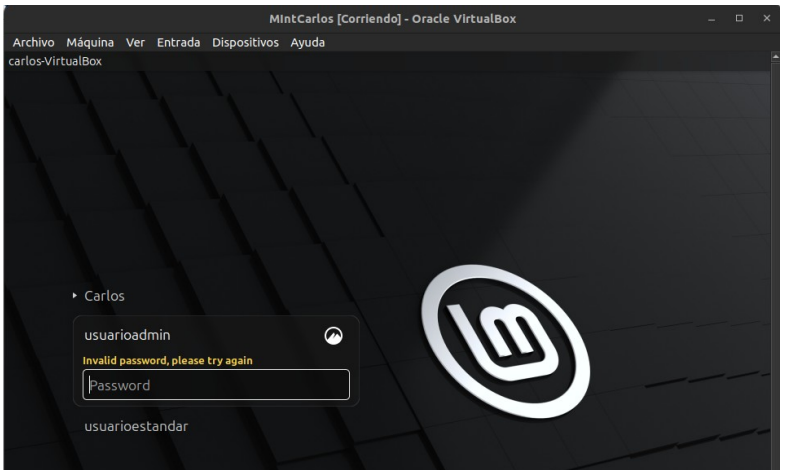
Finalmente, hay que modificar otro archivo PAM, en **/etc/pam.d/common-account** hay que poner al principio la línea **account required pam_faillock.so** (Esta línea le indica al sistema: "Si la autenticación ha sido exitosa, borra cualquier registro de fallo de este usuario".)



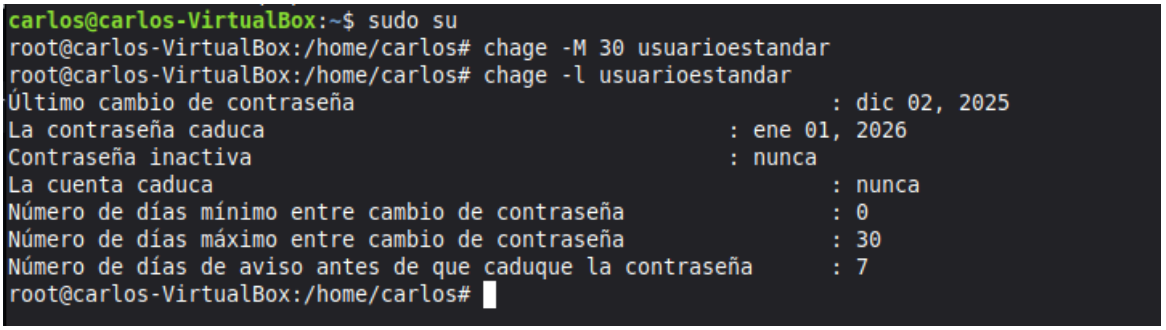


Por último, comprobamos que funcione (Como vemos, luego del 4º Intento, aunque ponga la contraseña correcta, no me deja iniciar):





Paso 3. Seguridad de Contraseñas



Para que la contraseña del usuario "**usuarioestandar**" caduque en 30 días utilizamos un comando en concreto y luego lo verificamos (como vemos, la contraseña caduca 30 días después de la modificación).

Comandos (si somos usuarios root no hace falta usar "sudo"):

 # 1. Establece la caducidad máxima (30 días):

sudo chage -M 30 "usuarioestandar"

 # 2. Verifica la configuración:

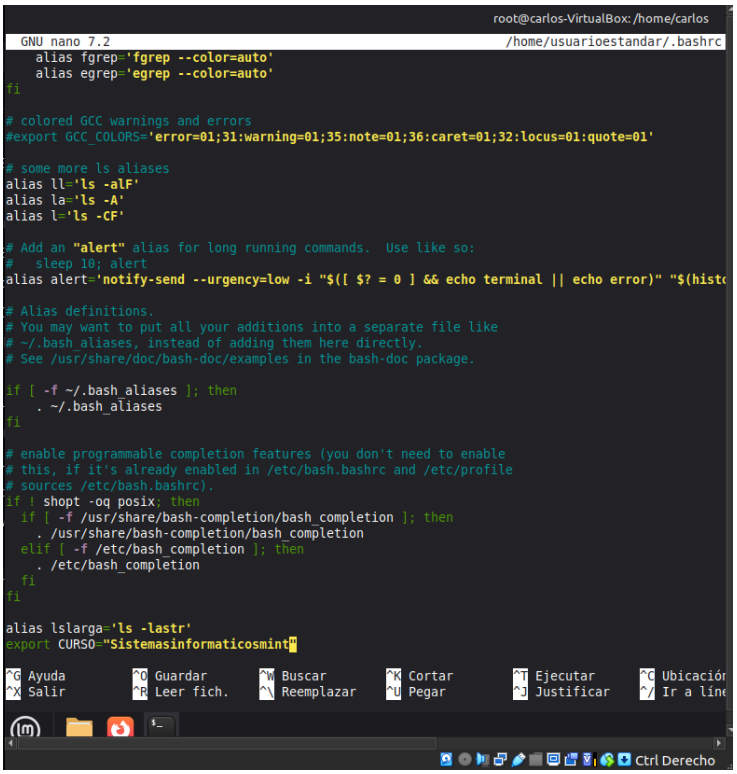
sudo chage -l "usuarioestandar"

Paso 4. Gestión del Entorno de Trabajo del Usuario

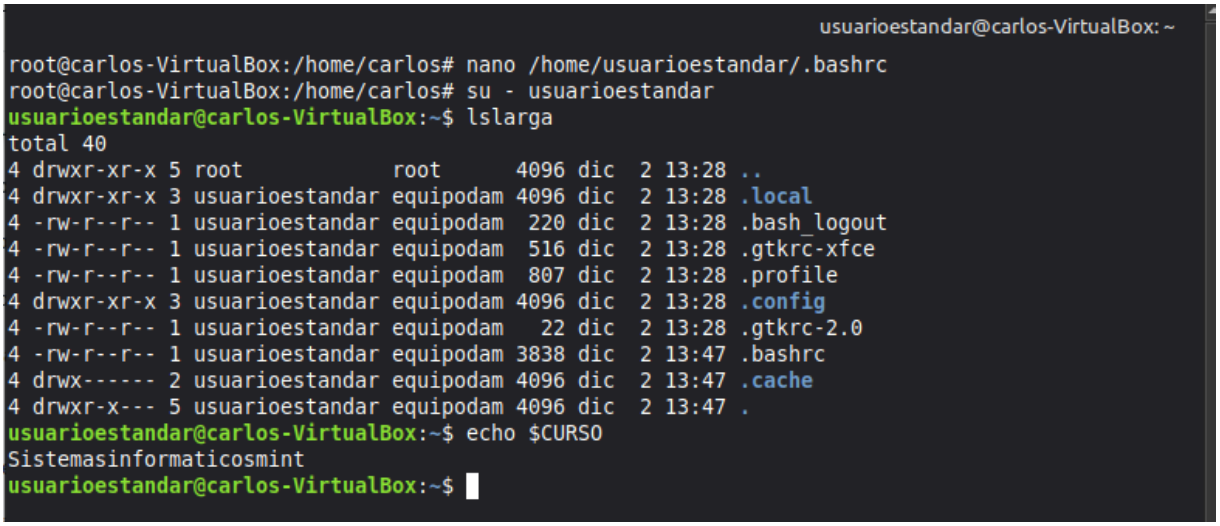
Crearemos un alias "**lslarga**" y una variable "**CURSO**" para el usuario "**usuarioestandar**". Para ello modificaremos el archivo personal del usuario **/home/usuarioestandar/.bashrc** e insertaremos estas 2 líneas al final del archivo:

alias lslarga='ls -latr'

export CURSO="Sistemasinformaticosmint"



Y ya por último, lo comprobamos:



Comandos (si somos usuarios root no hace falta usar “sudo”):

 # 1. Accedomos al archivo .bashrc:

sudo nano /home/usuarioestandar/.bashrc

 # 2. Cambiamos al usuarioestandar para verificar que funcione:

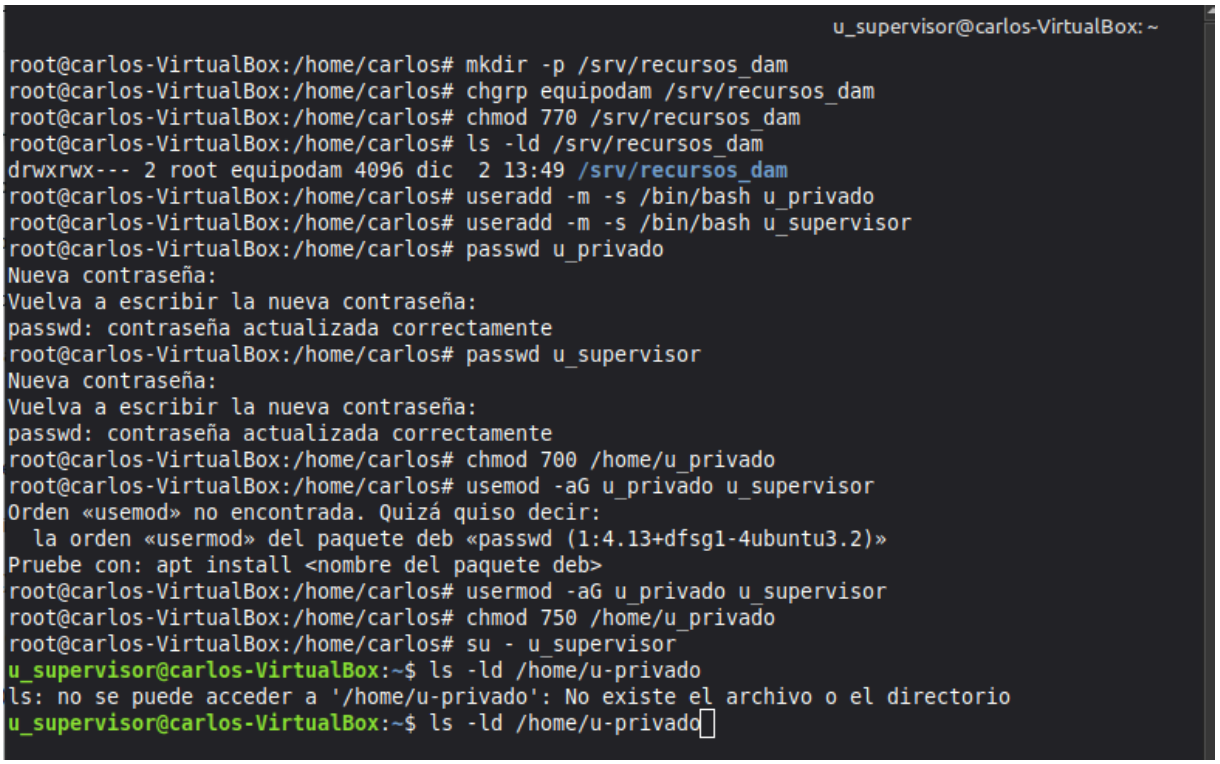
su – “usuarioestandar”

 # 3. Comprobamos que funcionen el alias y la variable:

lslarga

echo $CURSO

Paso 5. Acceso a Recursos y Permisos Locales



Parte 1:

Primeramente creamos un directorio (carpeta) donde unicamente el grupo **equipodam** (asignando al grupo propietario) pueda modificar. Luego, configuramos los permisos (el número **770** hace referencia a: 7(Dueño/root)rwx, 7(Grupo asignado)rwx, 0(Otros)) y verificamos los permisos (como podemos observar, tanto **root** como el grupo "**equipodam**" tiene permisos).

Comandos (Parte 1) (si somos usuarios root no hace falta usar "sudo"):

 # 1. Creamos el directorio que queremos modificar los permisos:

sudo mkdir -p /srv/recursos_dam

 # 2. Asignamos al grupo propietario:

sudo chgrp equipodam /srv/recursos_dam

 # 3. Asignamos privilegios al directorio/carpeta:

sudo chmod 770 /srv/recursos_dam

 # 4. Verificamos los permisos:

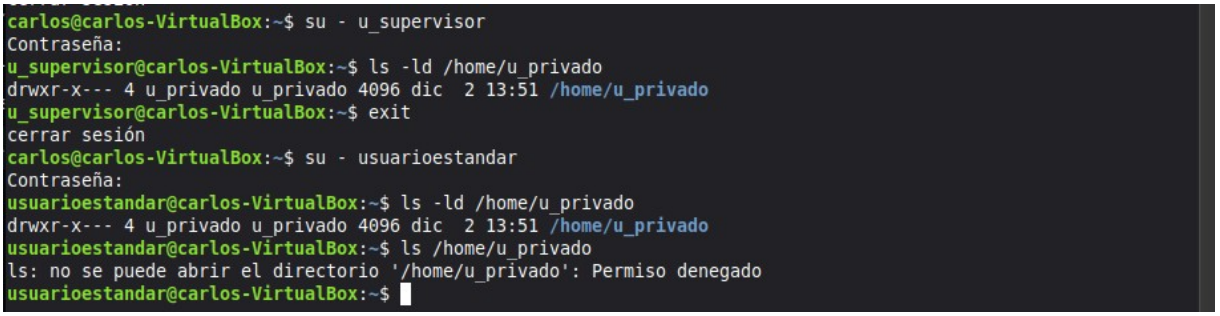
ls -ld /srv/recursos_dam

Parte 2:

Creamos los usuarios **u_supervisor**(puede entrar a la carpeta de privado) y **u_privado**. Dato: En linux Mint, nosotros al crear un usuario, el sistema crea un grupo con el mismo nombre, lo que nos servirá para

más adelante.

Por lo que pasaremos a crear los usuarios con sus contraseñas y pasamos a configurar los permisos con **700** (Bloqueamos el acceso a los datos). Luego, pasaremos a añadir al "**usuariosupervisor**" al grupo de "**usuarioprivado**" (que ya existe). Y ahora sí permitimos los permisos para el grupo con **750** (permitiendo que pueda acceder y leer).



Por último, verificamos que este configurado correctamente.

Reflexión: vemos como a los dos usuarios les deja acceder con el comando **ls -ld /home/u_privado** (¿Eso quiere decir que no esta funcionando correctamente?). Ese comando lo único que hace es comprobar cuales son las propiedades de esa carpeta. Para realmente comprobar el acceso utilizaremos el comando ls /home/u_privado (sin **-ld**) o simplemente intentando entrar en la carpeta.

Comandos (Parte 2) (si somos usuarios root no hace falta usar "sudo"):

 # 1. Creamos los usuarios y le asignamos una contraseña:

sudo useradd -m -s /bin/bash "u_privado/u_supervisor"

sudo passwd "u_privado/u_supervisor"

 # 2. Aseguramos la privacidad inicial del grupo "u_privado":

sudo chmod 700 /home/u_privado

 # 3. Permitimos el acceso al grupo "usuario_privado" al usuario "usuario_supervisor":

sudo usermod -aG u_privado u_supervisor

 # 4. Habilitamos los permisos:

sudo chmod 750 /home/u_privado

 # 5. Realizamos la comprobación necesaria:

su - "u_supervisor/usuarioestandar"

ls -ld /home/u_privado

Paso 6. Configuración de la Impresión

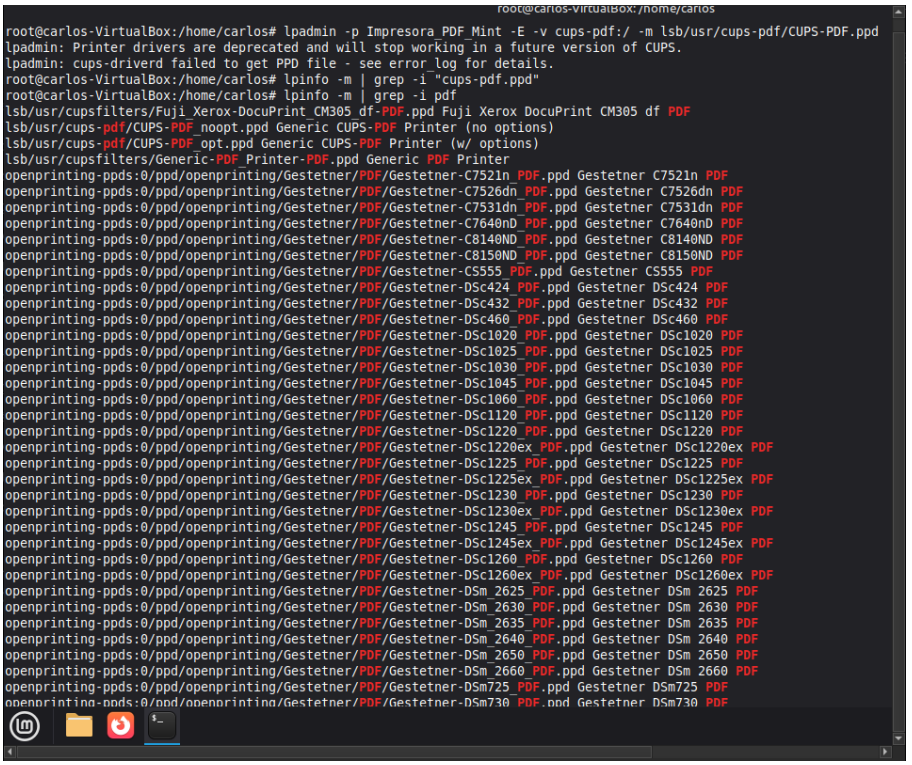
Primeramente, actualizamos la lista de paquetes con sudo su e instalamos el servidor CUPS junto con el paquete que permite imprimir en pdf con: **sudo apt install cups cups-pdf**.

```
root@carlos-VirtualBox:/home/carlos# usermod -aG lpadmin usuarioadmin
root@carlos-VirtualBox:/home/carlos# systemctl enable --now cups
Synchronizing state of cups.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable cups
root@carlos-VirtualBox:/home/carlos#
```

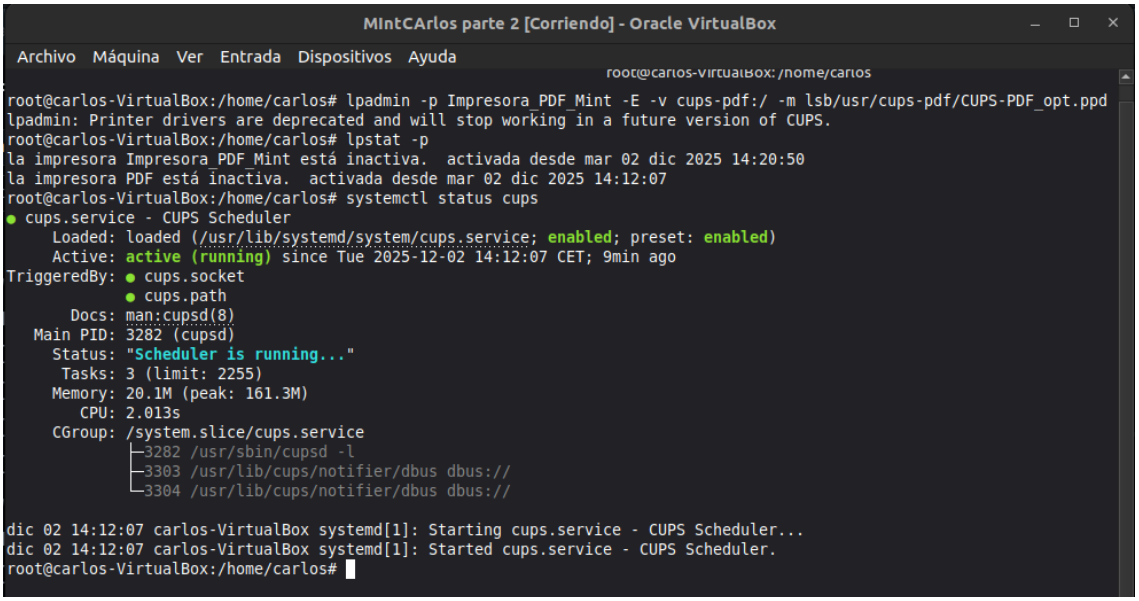
Ahora, como seguimos con los anteriores usuarios, un aporte extra fue añadir al usuario "**usuarioadmin**" al grupo **lpadmin** (grupo que gestiona las impresora en Mint). Para finalizar, habilitamos el servicio de impresión para que se ejecute en futuros arranques y ahora.

```
root@carlos-VirtualBox:/home/carlos
root@carlos-VirtualBox:/home/carlos# lpadmin -p Impresora_PDF_Mint -E -v cups-pdf:/ -m lsb/usr/cups-pdf/CUPS-PDF.ppd
lpadmin: Printer drivers are deprecated and will stop working in a future version of CUPS.
lpadmin: cups-driverd failed to get PPD file - see error_log for details.
root@carlos-VirtualBox:/home/carlos#
```

Seguimos con añadir la impresora PDF a **lpadmin** mediante **cups-pdf:/** (contiene un driver genérico). Sin embargo, en este caso el nombre genérico no funciono y tuve que encontrar una solución para ello.



Para poder encontrar el nombre exacto del driver preguntamos al sistema el driver del PDF que esta instalado. Se puede utilizar el comando **lpinfo -m | grep -i "cups-pdf.ppd"** pero si no existe no te aparecerá nada (como a mi), por lo que utilizaremos el comando **lpinfo -m | grep -i pdf** y buscamos la línea que diga **CUPS-PDF.ppd** (En mi caso, el driver se llama **CUPS-PDF_opt.ppd**). Ahora si, utilizamos el primer comando para de **lpadmin** con este nuevo nombre y verificamos que no da error y que aparece en la lista.



Reflexión: como observamos en la consola, finalmente funciona, pero nos advierte de que el driver es muy obsoleto y es mejor actualizarlo (sin embargo funciona y que esta active).

Comandos (si somos usuarios root no hace falta usar “sudo”):

 # 1. Instalamos el servidor CUPS y el paquete cups-pdf:

sudo apt install cups cups-pdf

 # 2. Configuramos permisos para acceder al grupo “lpadmin”:

sudo usermod -aG lpadmin usuario_admin

 # 3. Habilitamos el servicio de impresión para futuros arranques:

sudo systemctl enable --now cups

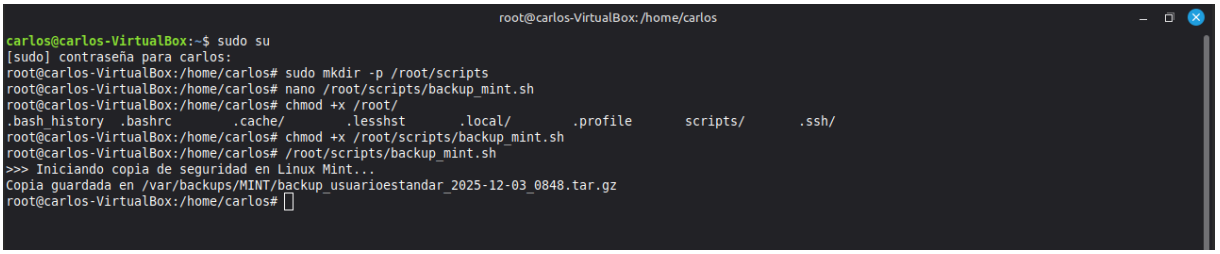
 # 4. Añadimos la impresora PDF a lpadmin (forma correcta):

sudo lpadmin -p Impresora_PDF_Mint -E -v cups-pdf:/ -m lsb/usr/cups-pdf/CUPS-PDF_opt.ppd

 # 5. Verificamos su estado:

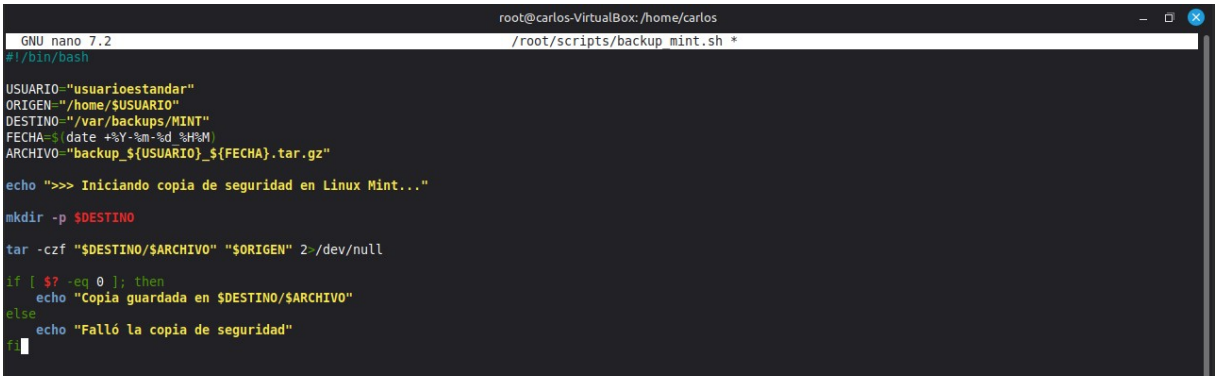
lpstat -p

Paso 7. Programación Básica de Shell Script



Primeramente creamos una carpeta donde guardaremos el script dentro de **root**.

Seguidamente, crearemos el archivo con nano dentro de esa carpeta, ese archivo se llamara **backup_mint.sh** (con el nombre que quieras pero que termine por **.sh)**.



Realizaremos un pequeño código que realizara una copia de seguridad al ejecutarlo, en caso se que algo falle, mandara un mensaje de advertencia.

Para finalizar, le otorgaremos permisos al archivo y lo ejecutaremos para comprobar que funcione y finalmente terminamos la práctica de Linux Mint.

<u>Comandos (si somos usuarios root no hace falta usar “sudo”):</u>

 # 1. Crearemos el directorio donde guardar el script:

sudo mkdir -p /root/scripts

 # 2. Crearemos el archivo que realizara la copia de seguridad:

sudo nano /root/scripts/backup_mint.sh

 # 3. Otorgaremos permisos al archivo:

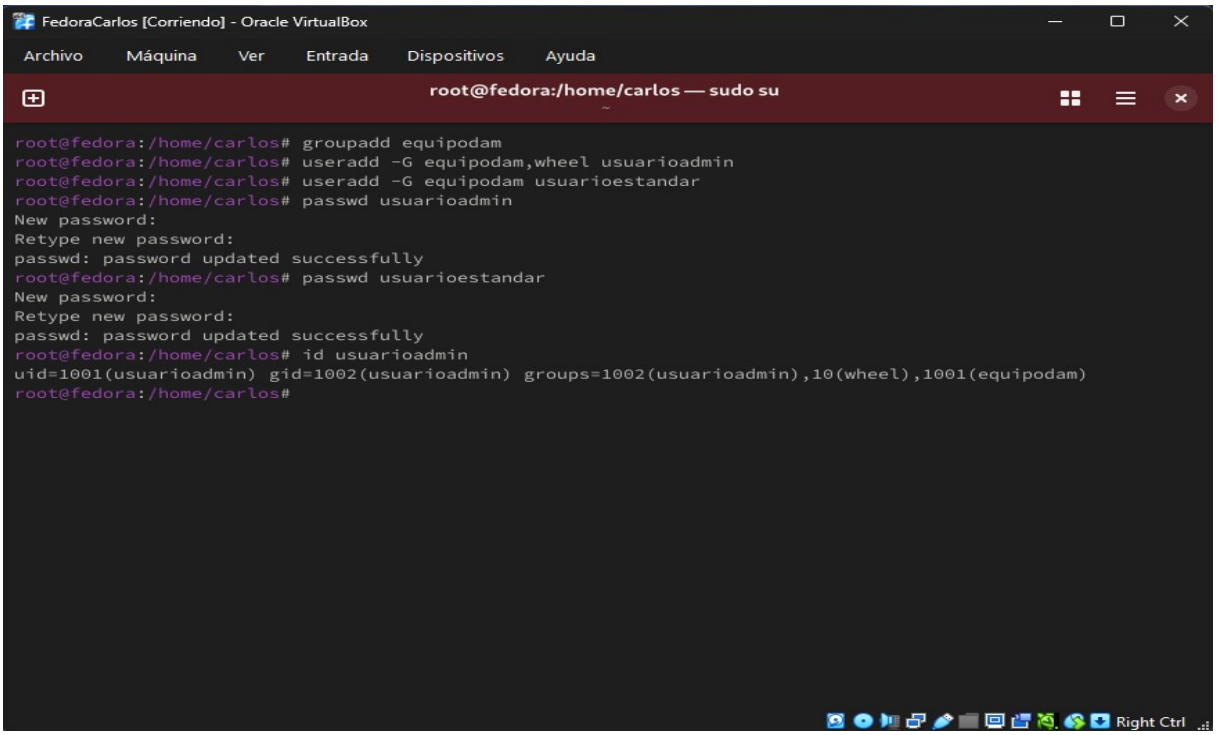
sudo chmod +x /root/scripts/backup_mint.sh

 # 4. Verificamos que se haya realizado correctamente el backup:

ls -lh /var/backups/MINT

Fedora

Paso 1. Configuración de Usuarios y Grupos Locales



Como usuarios **root** primeramente creamos el grupo y dos usuarios (**usuarioadmin** y **usuarioestandar**) y los metemos en el grupo "**equipodam**", al primero le asignamos privilegios (**con wheel**) y verificamos que pertenecen a ese grupo.

Comandos (si somos usuarios root no hace falta usar "sudo"):

 # 1. Crear el grupo compartido (como root)

sudo groupadd equipodam

 # 2. Crea usuarios y los asigna a un grupo (equipodam)

sudo useradd -G "equipodam",wheel "usuarioadmin" (Privilegios)

sudo useradd -G "equipodam" "usuarioestandar"

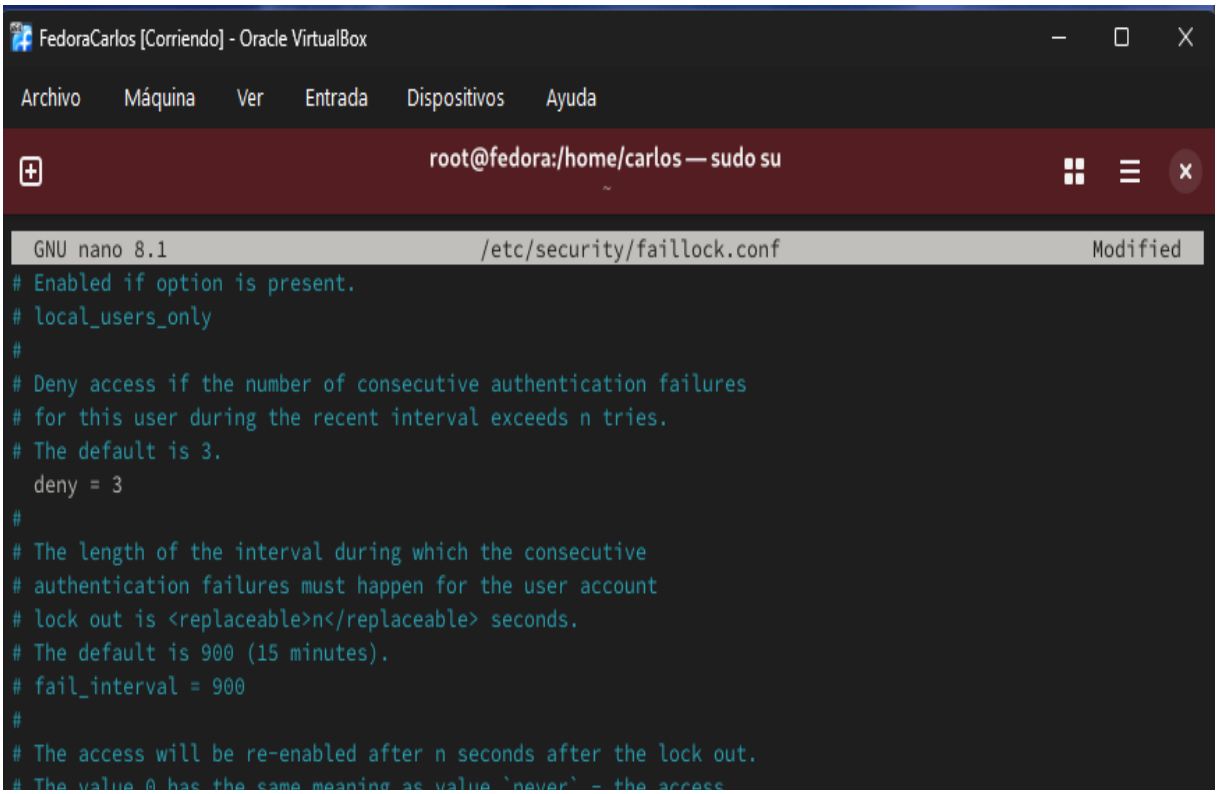
 # 4. Verifica que pertenece a un grupo

sudo id "usuarioadmin/usuarioestandar"

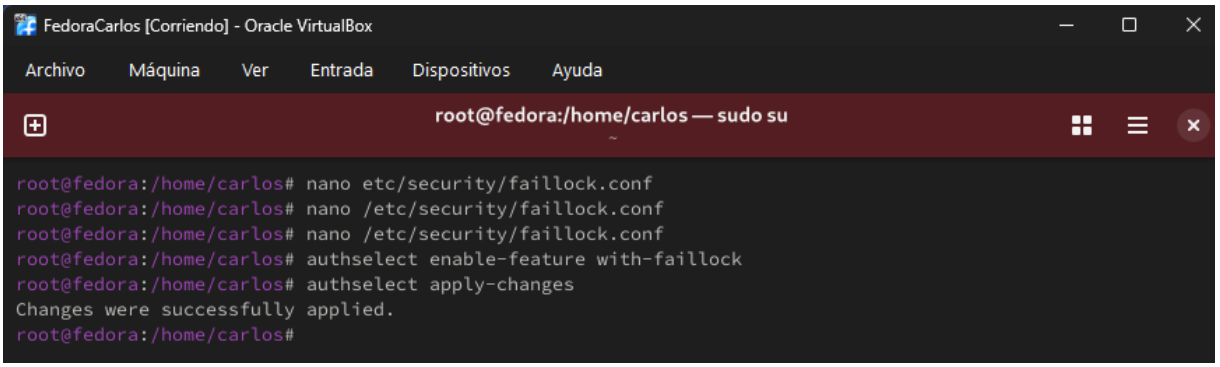
Paso 2. Seguridad de Cuentas de Usuario (utilizaremos el comando "nano" para acceder a estos archivos)

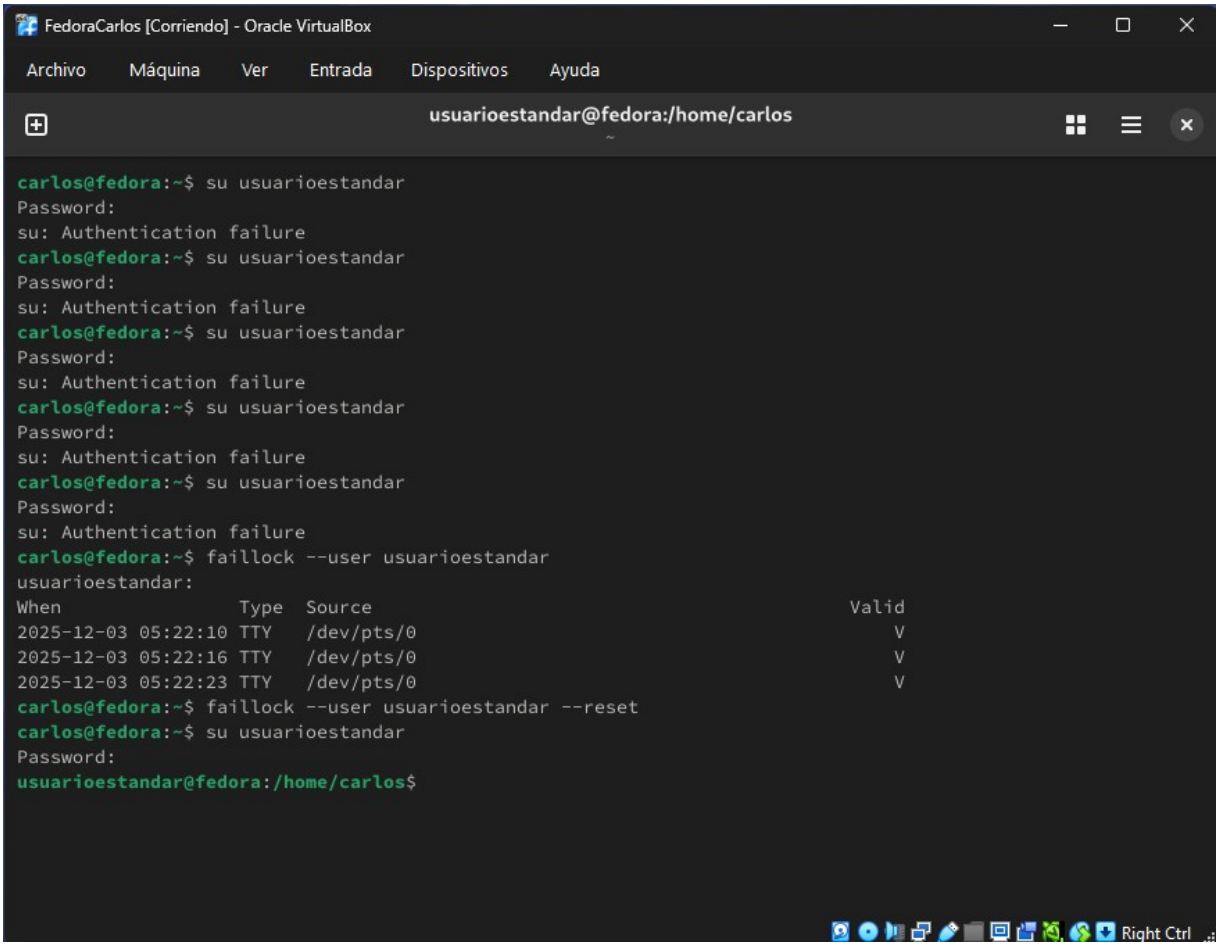
Para poder modificar la política de bloque debemos acceder a los archivo PAM.

Primeramente modificaremos el archivo **/etc/security/faillock.conf**:



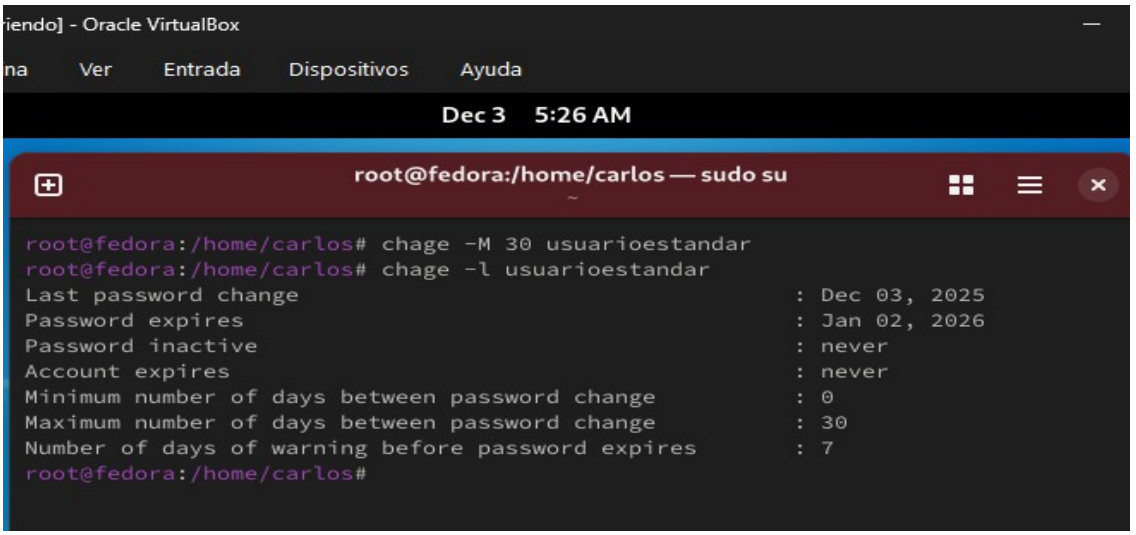
Una vez dentro del archivo, lo único que tenemos que hacer es descomentar la líneas "**#deny**" y asignarle 3 (cantidad de intentos acertados). También puedes descomentar o crear la línea "**unlock_time**", y asignarle el tiempo que la pantalla se bloquea. Por último, debemos habilitar los comandos (características) por si por algún caso, no vinieran activadas (**authselect enable-feature with-faillock**) y guardar el archivo y aplicarlo los cambios con el comando **authselect apply-changes**.





Mediante la verificación (intentamos acceder al usuarioestandar) comprobamos que luego del tercer intento, al insertar la contraseña correcta, de igual manera no la da por errónea. Podemos verificar que esta funcionando por el comando **faillock --user usuario_estandar**, luego de los tres intentos nos tendría que salir en cada intento un **Valid: V**. ¿Pero la V de valid no significa que la contraseña es correcto?, en este caso no, puesto que lo único que está validando con esa V es que el error fue válido, válido para el conteo de bloqueo(Lo que significa que a contado los errores límites). Para poder resetear o poder volver a poner la contraseña, utilizamos **faillock --user usuario_estandar –reset**, que resetea el contador de fallos a 0 (como podemos observar en la imagen, al resetear y poner la contraseña, ahora nos deja entrar).

Paso 3. Seguridad de Contraseñas



Para que la contraseña del usuario "**usuarioestandar**" caduque en 30 días utilizamos un comando en concreto y luego lo verificamos (como vemos, la contraseña caduca 30 días después de la modificación).

Comandos (si somos usuarios root no hace falta usar "sudo"):

 # 1. Establece la caducidad máxima (30 días):

sudo chage -M 30 "usuarioestandar"

(-M = cantidad máxima de días de validez)

 # 2. Verifica la configuración:

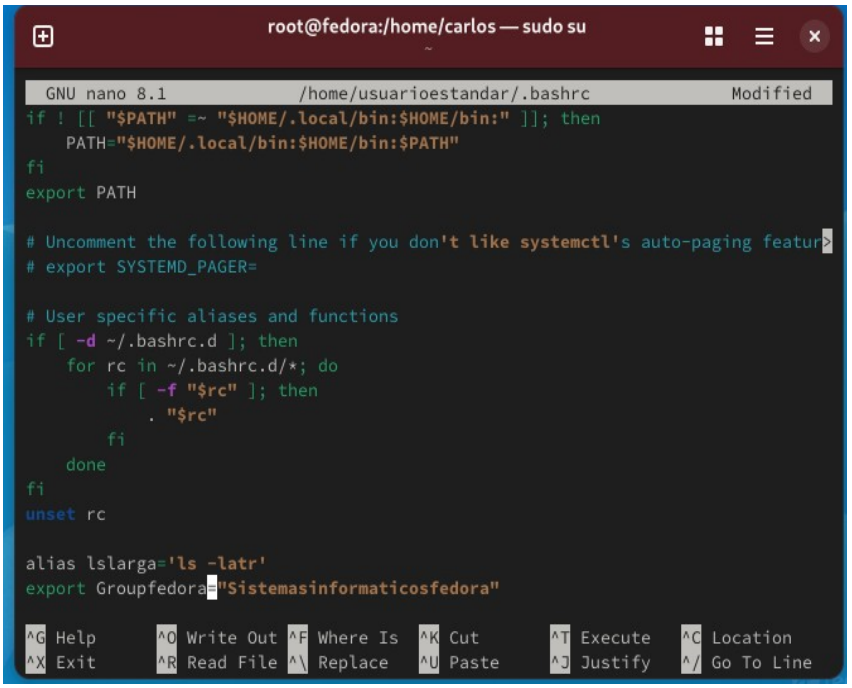
sudo chage -l "usuarioestandar"

Paso 4. Gestión del Entorno de Trabajo del Usuario

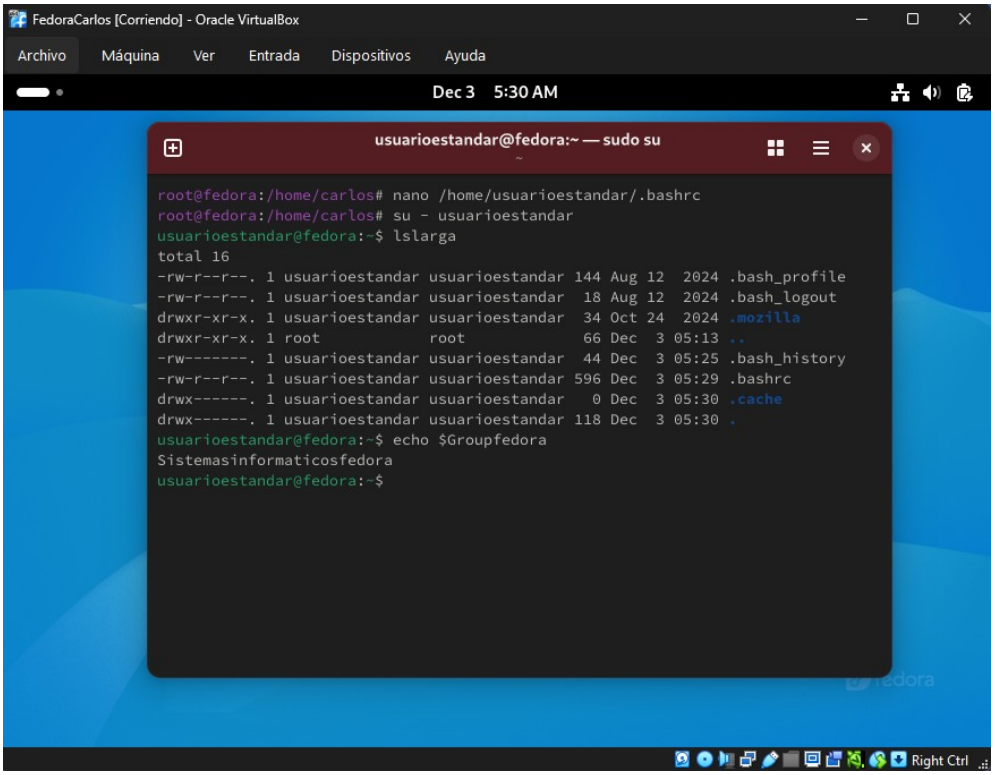
Crearemos un alias "**lslarga**" y una variable "**Groupfedora**" para el usuario "**usuarioestandar**". Para ello modificaremos el archivo personal del usuario **/home/usuarioestandar/.bashrc** e insertaremos estas 2 líneas al final del archivo:

alias lslarga='ls -latr'

export Groupfedora="Sistemasinformaticosfedora"



Y ya por último, lo comprobamos:



Comandos (si somos usuarios root no hace falta usar “sudo”):

 # 1. Accedomos al archivo .bashrc:

sudo nano /home/usuarioestandar/.bashrc

 # 2. Cambiamos al usuarioestandar para verificar que funcione:

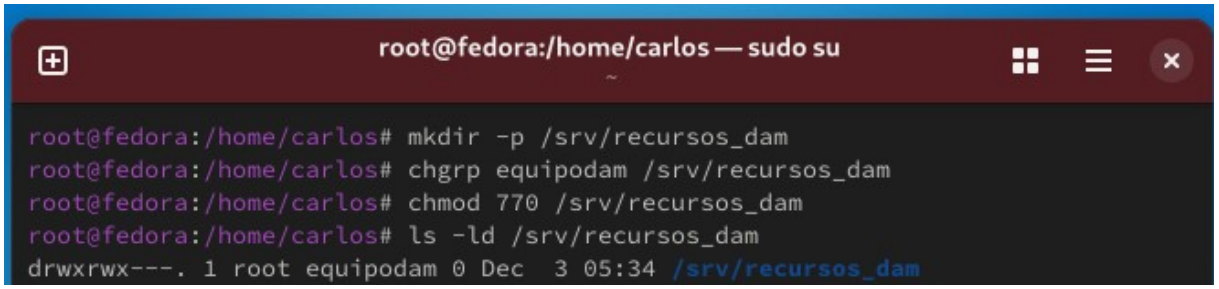
su – “usuarioestandar”

 # 3. Comprobamos que funcionen el alias y la variable:

lslarga

echo $Groupfedora

Paso 5. Acceso a Recursos y Permisos Locales



Parte 1:

Primeramente creamos un directorio (carpeta) donde unicamente el grupo **equipodam** (asignando al grupo propietario) pueda modificar. Luego, configuramos los permisos (el número **770** hace referencia a: 7(Dueño/root)rwx, 7(Grupo asignado)rwx, 0(Otros)) y verificamos los permisos (como podemos observar, tanto **root** como el grupo "**equipodam**" tiene permisos).

Comandos (Parte 1) (si somos usuarios root no hace falta usar "sudo"):

 # 1. Creamos el directorio que queremos modificar los permisos:

sudo mkdir -p /srv/recursos_dam

 # 2. Asignamos al grupo propietario:

sudo chgrp equipodam /srv/recursos_dam

 # 3. Asignamos privilegios al directorio/carpeta:

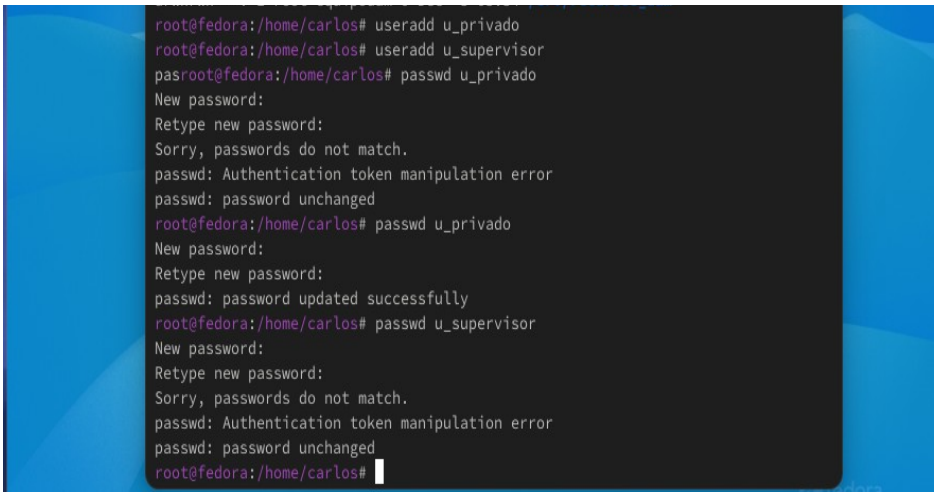
sudo chmod 770 /srv/recursos_dam

 # 4. Verificamos los permisos:

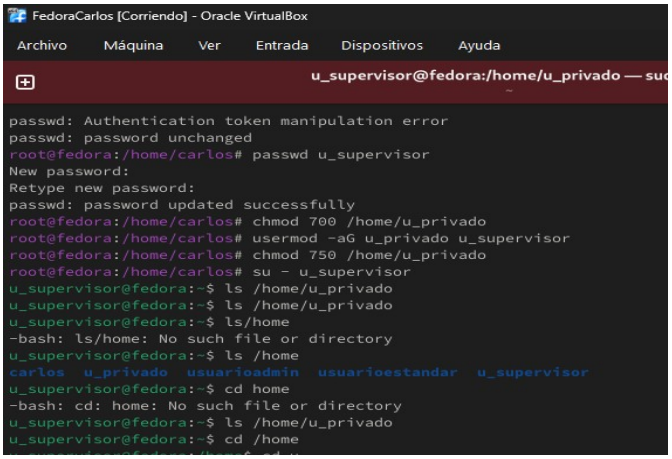
ls -ld /srv/recursos_dam

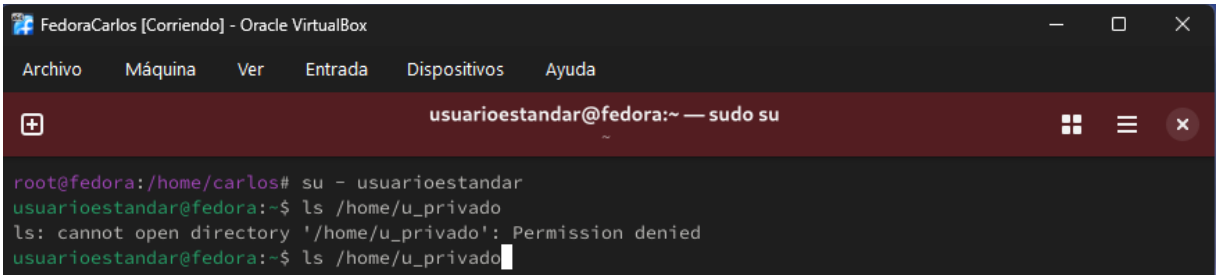
Parte 2:

Creamos los usuarios **u_supervisor**(puede entrar a la carpeta de privado) y **u_privado**. Dato: En Fedora al igual que mint, nosotros al crear un usuario, el sistema crea un grupo con el mismo nombre, lo que nos servirá para más adelante.



Por lo que pasaremos a crear los usuarios con sus contraseñas y pasamos a configurar los permisos con **700** (Bloqueamos el acceso a los datos). Luego, pasaremos a añadir al "**usuariosupervisor**" al grupo de "**usuarioprivado**" (que ya existe). Y ahora sí permitimos los permisos para el grupo con **750** (permitiendo que pueda acceder y leer).





Por último, verificamos que este configurado correctamente.

Reflexión: lo primero que hacemos aquí es acceder al u_supervisor (lo que nos tendría que dejar ver el contenido). Al intentar varias veces poner el comando bien, nos lo acepta y nos permite ver el contenido (como estaba vacío no nos muestra nada). Ahora, realizamos la mismas acciones con el usuarioestandar, denegando el acceso a ver el contenido. Puede que no haya sido necesario para verificar los permisos con el comando **ls -ld /home/u_privado**, esto puede ser por ciertas diferencia que tiene fedora con mint.

Comandos (Parte 2) (si somos usuarios root no hace falta usar "sudo"):

 # 1. Creamos los usuarios y le asignamos una contraseña:

sudo useradd "u_privado/u_supervisor"

sudo passwd “u_privado/u_supervisor”

 # 2. Aseguramos la privacidad inicial del grupo “u_privado”:

sudo chmod 700 /home/u_privado

 # 3. Permitimos el acceso al grupo “usuario_privado” al usuario “usuario_supervisor”:

sudo usermod -aG u_privado u_supervisor

 # 4. Habilitamos los permisos:

sudo chmod 750 /home/u_privado

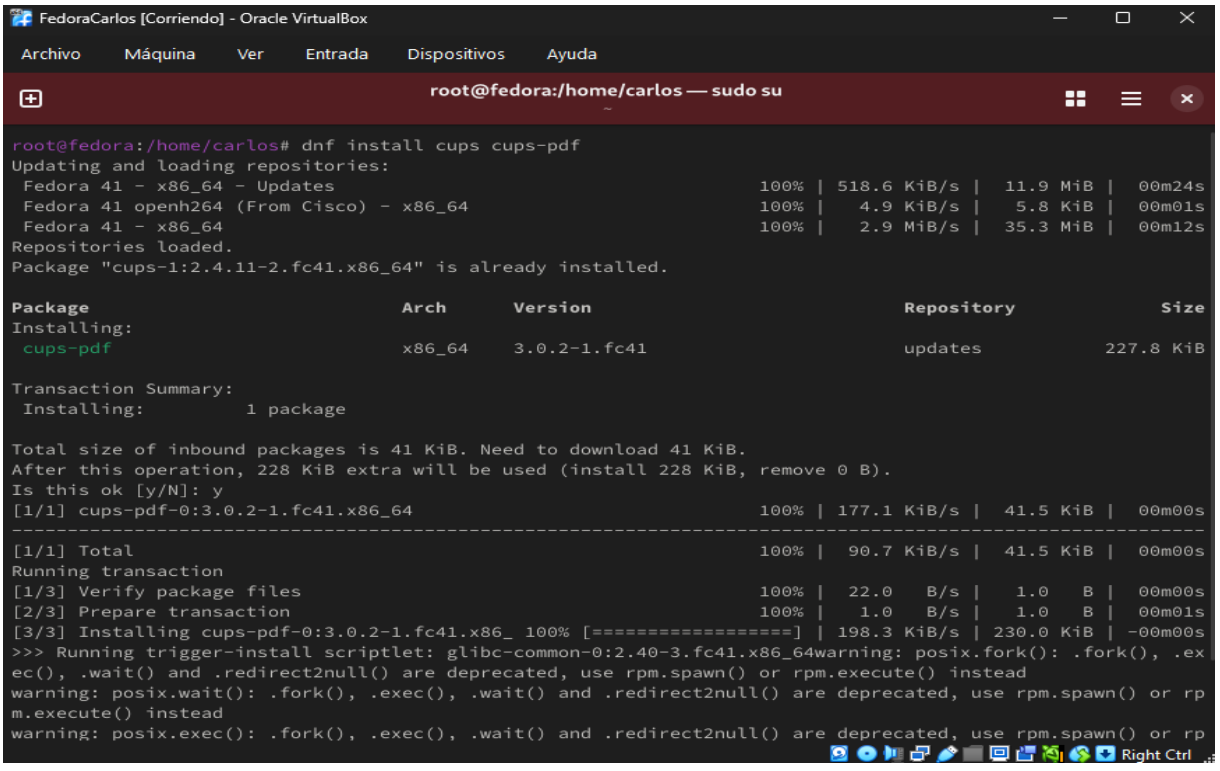
 # 5. Realizamos la comprobación necesaria:

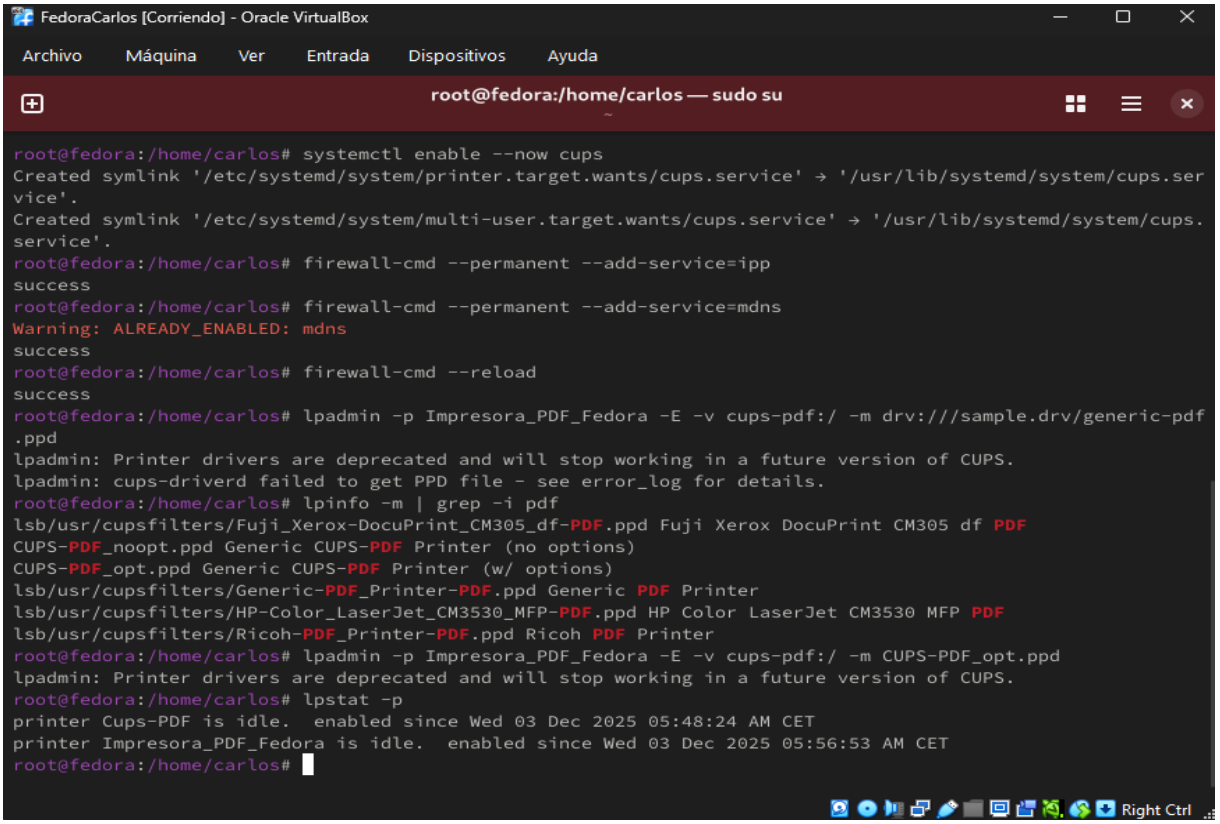
su - “u_supervisor/usuarioestandar”

ls /home/u_privado

Paso 6. Configuración de la Impresión

Primeramente, instalamos Cups para poder utilizar la impresora con **dnf**(gestor de paquetes de Fedora).





Una vez instalado, habilitamos y arrancamos el servicio de impresión y configuramos el **firewall** (algo que no hicimos en Mint pero que en Fedora es crucial). Debemos abrir los puertos para el protocolo de impresión (**IPP**) y el descubrimiento de impresoras (**mDNS**). Por último añadiremos la impresora con **lpadmin** utilizando un driver genérico (que obiamente no funciono) por lo que tuve que buscar el driver específico cuyo nombre por el que tenia que buscar era CUPS-PDF.ppd (se llamaba CUPS-PDF_opt.ppd). Utilicé el mismo comando de antes y se añadió con éxito. Por último, realizamos la comprobación de que esta en funcionamiento y ¡Listo!.

Comandos (si somos usuarios root no hace falta usar “sudo”):

 # 1. Instalamos el servidor CUPS y el paquete cups-pdf:

sudo dnf install cups cups-pdf

 # 3. Habilitamos el servicio de impresión para futuros arranques:

sudo systemctl enable --now cups

 # 4. Configurar el firewall (abrir puertos para el protocolo de impresión y también para el descubrimiento de impresoras):

firewall-cmd --permanent –add-service=ipp

firewall-cmd --permanent --add-service=mdns

firewall-cmd –reload

 # 5. Añadimos la impresora PDF a lpadmin (forma correcta):

sudo lpadmin -p Impresora_PDF_Fedora -E -v cups-pdf:/ -m CUPS-PDF_opt.ppd

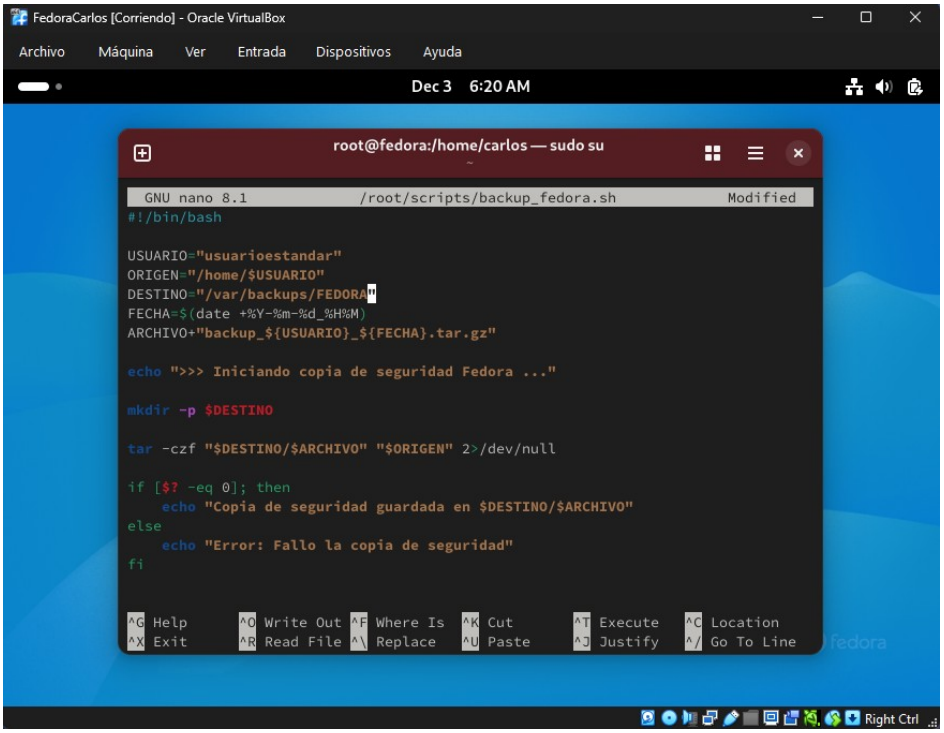
 # 6. Verificamos su estado:

lpstat -p

Paso 7. Programación Básica de Shell Script

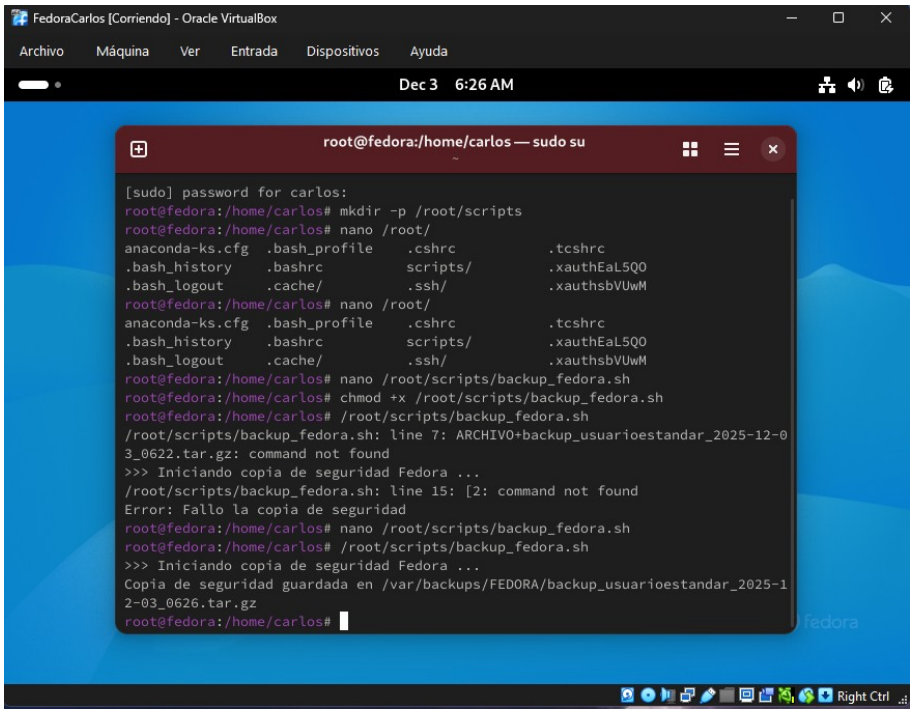
Primeramente creamos una carpeta donde guardaremos el script dentro de **root**.

Seguidamente, crearemos el archivo con nano dentro de esa carpeta, ese archivo se llamara **backup_fedora.sh** (con el nombre que quieras pero que termine por **.sh)**.



Realizaremos un pequeño código que realizara una copia de seguridad al ejecutarlo, en caso se que algo falle, mandara un mensaje de advertencia.

Para finalizar, le otorgaremos permisos al archivo y lo ejecutaremos para comprobar que funcione (Al principio sale un error ya que no puse un signo o letra) y finalmente terminamos la práctica de Linux Fedora.



Comandos (si somos usuarios root no hace falta usar “sudo”):

 # 1. Crearemos el directorio donde guardar el script:
sudo mkdir -p /root/scripts

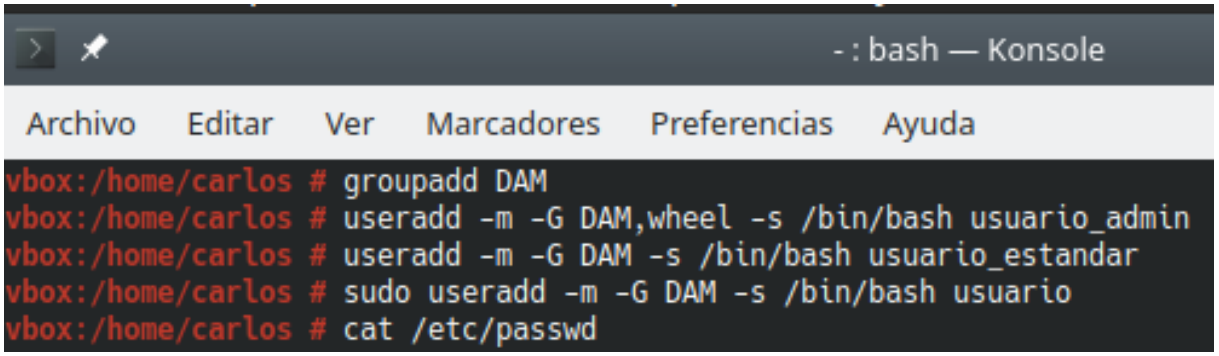
 # 2. Crearemos el archivo que realizara la copia de seguridad:
sudo nano /root/scripts/backup_fedora.sh

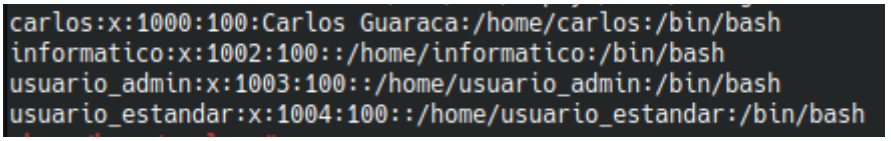
 # 3. Otorgaremos permisos al archivo:
sudo chmod +x /root/scripts/backup_fedora.sh

 # 4. Ejecutamos el script:
/root/scripts/backup_fedora.sh

OpenSuse

Paso 1. Configuración de Usuarios y Grupos Locales





Como usuarios **root** primeramente creamos el grupo y dos usuarios (**usuario_admin** y **usuario_estandar**) y los metemos en el grupo "**DAM**", al primero le asignamos privilegios (**con wheel**) y verificamos que los usuarios han sido creados.

Comandos (si somos usuarios root no hace falta usar "sudo"):

 # 1. Crear el grupo compartido (como root)

sudo groupadd DAM

 # 2. Crea usuarios y los asigna a un grupo (equipodam)

sudo useradd -m -G "DAM",wheel -s /bin/bash "usuario_admin" (Privilegios)

sudo useradd -m -G "DAM" -s /bin/bash "usuario_estandar"

 # 4. Verifica los usuarios existen

sudo cat a/etc/passwd

Paso 2. Seguridad de Cuentas de Usuario (utilizaremos el comando "nano" para acceder a estos archivos)

Para poder modificar la política de bloque debemos acceder a los archivo PAM.

Primeramente, intenté ejecutar el comando **sudo pam-config --add --faillock --faillock-deny=3 –faillock-unlock-time=600** (3 intentos y tiempo de bloqueo 600 segundos). Sin embargo, me dio un error, posiblemente porque la versión de OpenSuse que instalé era antigua y no erá compatible con faillock del todo (pam-config puesto que no reconoce el parámetro faillock).

```
[sudo] password for root:
vbox:/home/carlos # pam-config --add --faillock --faillock-deny=3 --faillock-unlock-time=600
pam-config: invalid option -- --faillock
Try `pam-config --help' or `pam-config --usage' for more information.
vbox:/home/carlos #
```

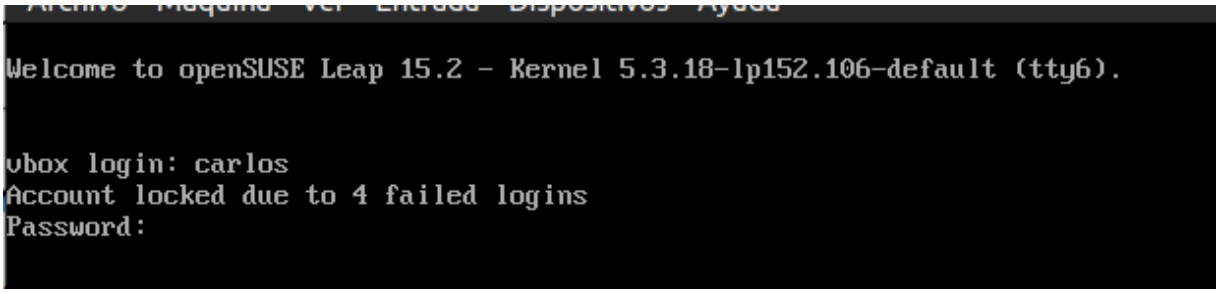
Pues ahora intentaremos cambiar el protocolo modificando el archivo **/etc/pam.d/login**:

```
#%PAM-1.0
auth       required   pam_faillock.so preauth audir silent deny=3 unlock_time=600
auth       requisite  pam_nologin.so
auth       include    common-auth
account    include    common-account
account    required   pam_faillock.so
password   include    common-password
session    required   pam_loginuid.so
session    optional   pam_keyinit.so force revoke
session    include    common-session
#session   optional   pam_lastlog.so nowtmp showfailed
session    optional   pam_mail.so standard
```

En este archivo inserte los comandos: **auth required pam_faillock.so preauth audit deny=3 unlock_time=600** (revisa el número de fallos durante la autentificación, 3 fallos posibles) y **account required pam_faillock.so** (reinicia el contador de fallos después de iniciar sesión). Sin embargo, esto no solucionó el problema. Tuve que buscar información y me entere que las líneas que puse deben estar en un orden en específico, seguí cambiando las líneas al principio de sus sección o al final. Pero no realizaba el conteo. Hasta que finalmente, mientras seguía buscando información, vi que a veces esas líneas se modificaban para la versión **pam_tally2.com** (solo para versiones antiguas) mientras que yo utilizaba las líneas de pam_faillock.so (una versión más moderna).

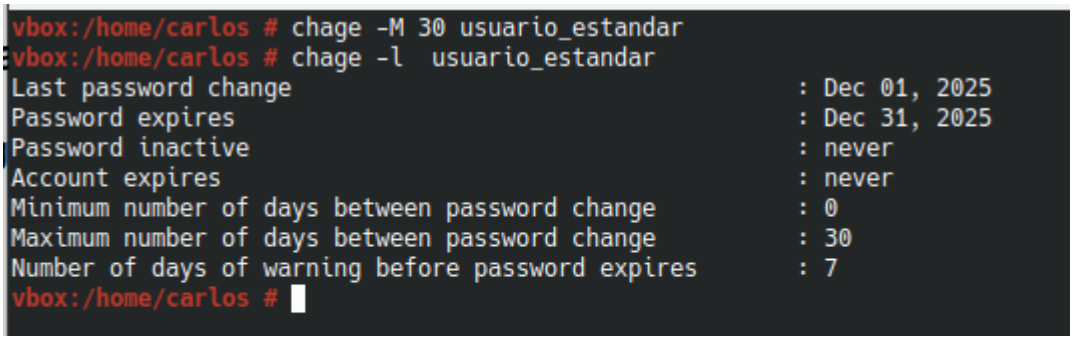
```
Archivo   Editar   Ver   Marcadores   Preferencias   Ayuda
 GNU nano 4.9.2                          /etc/pam.d/login
#%PAM-1.0
auth       requisite  pam_nologin.so
auth       required   pam_tally2.so deny=3 unlock_time=600
auth       include    common-auth
#auth      optional   pam_faillock.so authfail audit deny=3 unlock_time=600
account    include    common-account
#account   required   pam_faillock.so
account    required   pam_tally2.so reset
password   include    common-password
session    required   pam_loginuid.so
session    optional   pam_keyinit.so force revoke
session    include    common-session
#session   optional   pam_lastlog.so nowtmp showfailed
session    optional   pam_mail.so standard
```

Finalmente inserte las líneas: **auth required pam_tally2.so deny=3 unlock_time=600** (encima de la línea commo-auth) y **account required pam_tally2.so reset** (la última de la sección account), las otras las establecí como comentarios.



Al realizar los 3 intentos, al intento 4 me apareció esto (logrando el cometido).

Paso 3. Seguridad de Contraseñas



Para que la contraseña del usuario "**usuario_estandar**" caduque en 30 días utilizamos un comando en concreto y luego lo verificamos (como vemos, la contraseña caduca 30 días después de la modificación).

Comandos (si somos usuarios root no hace falta usar "sudo"):

 # 1. Establece la caducidad máxima (30 días):

sudo chage -M 30 "usuario_estandar"

(-M = cantidad máxima de días de validez)

 # 2. Verifica la configuración:

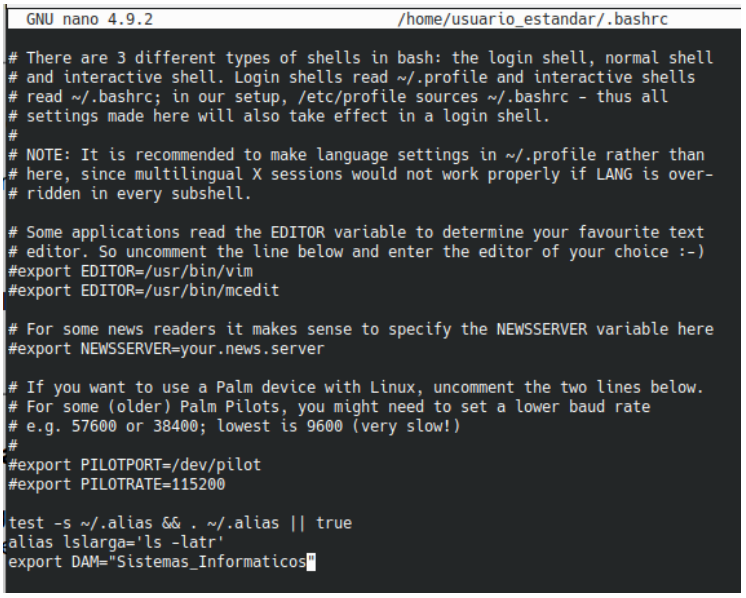
sudo chage -l "usuario_estandar"

Paso 4. Gestión del Entorno de Trabajo del Usuario

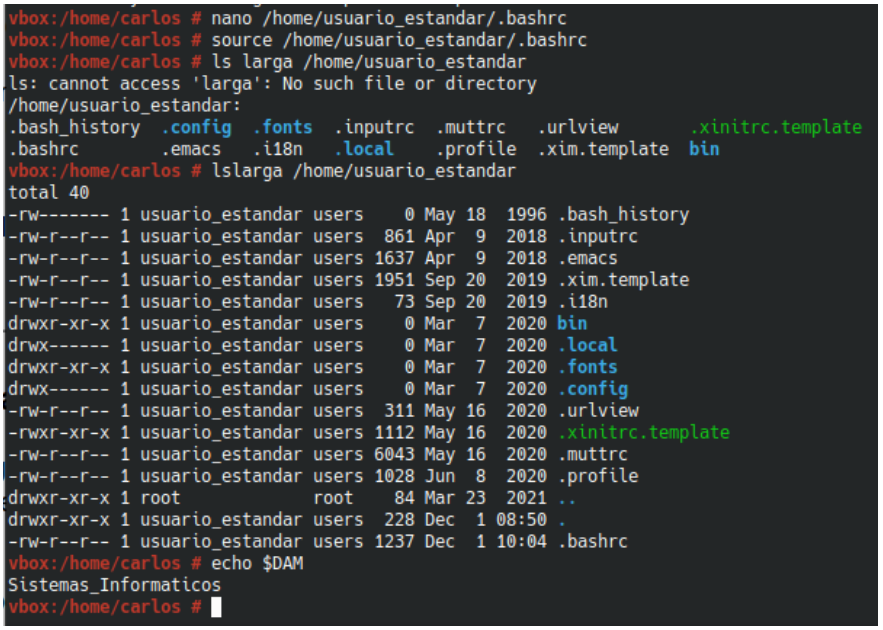
Crearemos un alias "**lslarga**" (como todas la anteriores) y una variable "**DAM**" para el usuario "**usuario_estandar**". Para ello modificaremos el archivo personal del usuario **/home/usuario_estandar/.bashrc** e insertaremos estas 2 líneas al final del archivo:

alias lslarga='ls -latr'

export DAM="Sistemas_Informaticos"



Y ya por último, lo comprobamos:



Comandos (si somos usuarios root no hace falta usar “sudo”):

 # 1. Accedomos al archivo .bashrc:

sudo nano /home/usuario_estandar/.bashrc

 # 2. Aplicamos los cambios hechos en .bashrc:

source /home/usuario_estandar/.bashrc

 # 3. Comprobamos que funcionen el alias y la variable:

lslarga /home/usuario_estandar

echo $DAM

Paso 5. Acceso a Recursos y Permisos Locales

```
vbox:/home/carlos # mkdir /srv/recursos_dam
vbox:/home/carlos # chgrp DAM /srv/recursos_dam
vbox:/home/carlos # chmod 770 /srv/recursos_dam
vbox:/home/carlos # ls -ld /srv/recursos_dam
drwxrwx--- 1 root DAM 0 Dec  1 10:08 /srv/recursos_dam
vbox:/home/carlos #
```

Parte 1:

Primeramente creamos un directorio (carpeta) donde unicamente el grupo **DAM** (Asignarlo a la carpeta creada) pueda modificar. Luego, configuramos los permisos (el número **770** hace referencia a: 7(Dueño/root)rwx, 7(Grupo asignado)rwx, 0(Otros)) y verificamos los permisos (como podemos observar, tanto **root** como el grupo "**DAM**" tiene permisos).

Comandos (Parte 1) (si somos usuarios root no hace falta usar "sudo"):

 # 1. Creamos el directorio que queremos modificar los permisos:

sudo mkdir /srv/recursos_dam

 # 2. Asignamos al grupo propietario:

sudo chgrp DAM /srv/recursos_dam

 # 3. Asignamos privilegios al directorio/carpeta:

sudo chmod 770 /srv/recursos_dam

 # 4. Verificamos los permisos:

ls -ld /srv/recursos_dam

Parte 2:

Creamos los usuarios **u_supervisor**(puede entrar a la carpeta de privado) y **u_privado**. (Dato: al crear el usuario **u_privado** , algunos SO antiguos no crea automáticamente un grupo con el mismi nombre)

```
vbox:/home/carlos # useradd -m u_privado
vbox:/home/carlos # useradd -m u_supervisor
vbox:/home/carlos # passwd u_privado
New password:
Retype new password:
passwd: password updated successfully
vbox:/home/carlos # passwd u_supervisor
New password:
Retype new password:
passwd: password updated successfully
vbox:/home/carlos # chmod 700 /home/u_privado
vbox:/home/carlos # usermod -aG u_privado u_supervisor
usermod: group 'u_privado' does not exist
vbox:/home/carlos # usermod -aG u_privado u_supervisor
usermod: group 'u_privado' does not exist
vbox:/home/carlos # groupadd g_privado
vbox:/home/carlos # usermod -aG g_privado u_supervisor
vbox:/home/carlos # chgrp g_privado /home/u_privado
vbox:/home/carlos # chmod 750 /home/u_privado
vbox:/home/carlos # su - u_supervisor
u_supervisor@vbox:~> ls -ld /home/u_privado
drwxr-x--- 1 u_privado g_privado 240 dic  1 10:12 /home/u_privado
u_supervisor@vbox:~> exit
logout
vbox:/home/carlos # su - usuario_estandar
usuario_estandar@vbox:~> ls -ld /home/u_privado
drwxr-x--- 1 u_privado g_privado 240 dic  1 10:12 /home/u_privado
usuario_estandar@vbox:~> ls /home/u_privado
ls: no se puede abrir el directorio '/home/u_privado': Permiso denegado
usuario_estandar@vbox:~>
```

Pasaremos a crear los usuarios con sus contraseñas y pasamos a configurar los permisos de la carpeta de **u_privado** con **700** (Bloqueamos el acceso a los datos a todos, incluido a **u_supervisor**). Luego, pasaremos a añadir al "**u_supervisor**" al grupo de "**u_privado**" (no se puede ya que no existe el grupo **u_privado**). Primeramente creamos un nuevo grupo y meteremos en ese grupo a **u_supervisor**. Ahora sí añadiremos la carpeta u_privado al grupo **g_privado**. Asignamos permisos para que los del mismo grupo puedan ver el contenido (**750**) y verificamos que nos permita ver.

Comandos (Parte 2) (si somos usuarios root no hace falta usar "sudo"):

 # 1. Creamos los usuarios y le asignamos una contraseña:

sudo useradd -m "u_privado/u_supervisor"

sudo passwd "u_privado/u_supervisor"

 # 2. Aseguramos la privacidad inicial del grupo "u_privado":

sudo chmod 700 /home/u_privado

 # 3. Creamos un nuevo grupo para añadir a los dos usuarios:

sudo groupadd g_privado

 # 4. Añadiremos al grupo a u_supervisor y la capeta de u_privado:

sudo usermod -aG g_privado u_supervisor

chgrp g_privado /home/u_privado

 # 5. Permitimos el acceso al grupo "usuario_privado" al usuario "usuario_supervisor":

sudo usermod -aG u_privado u_supervisor

 # 6. Habilitamos los permisos:

sudo chmod 750 /home/u_privado

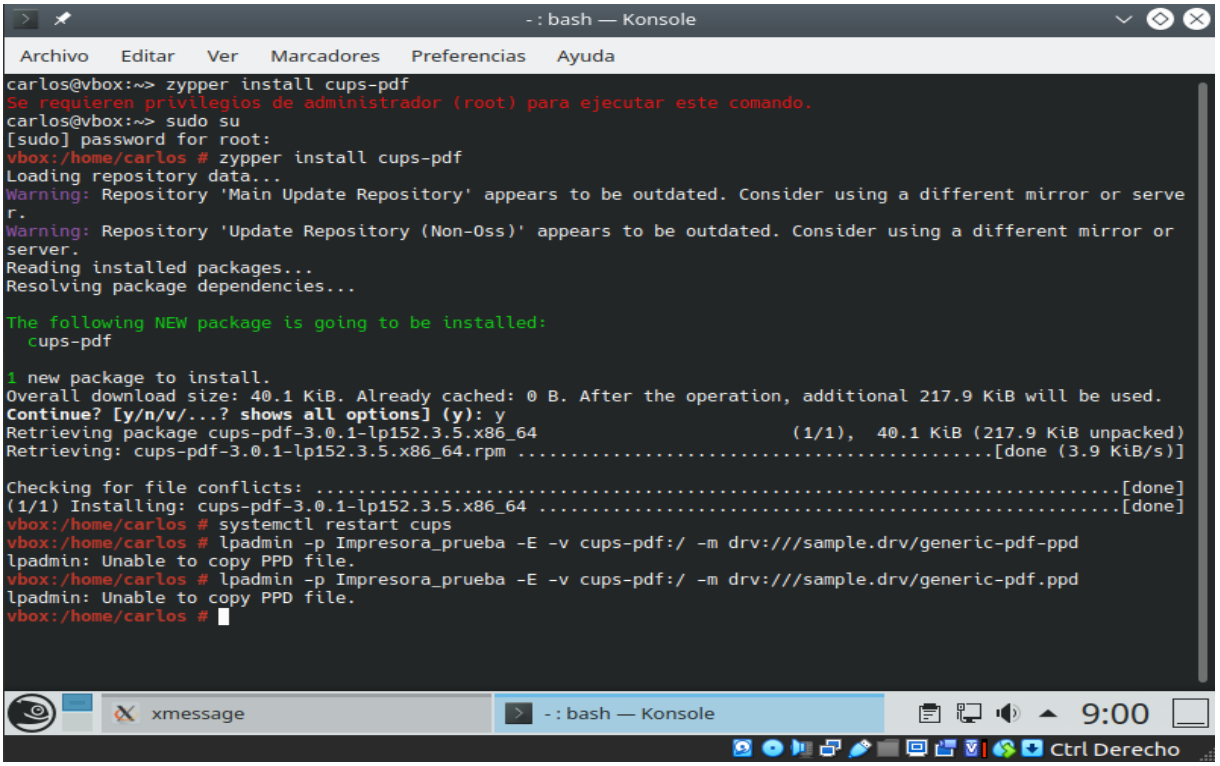
 # 7. Realizamos la comprobación necesaria:

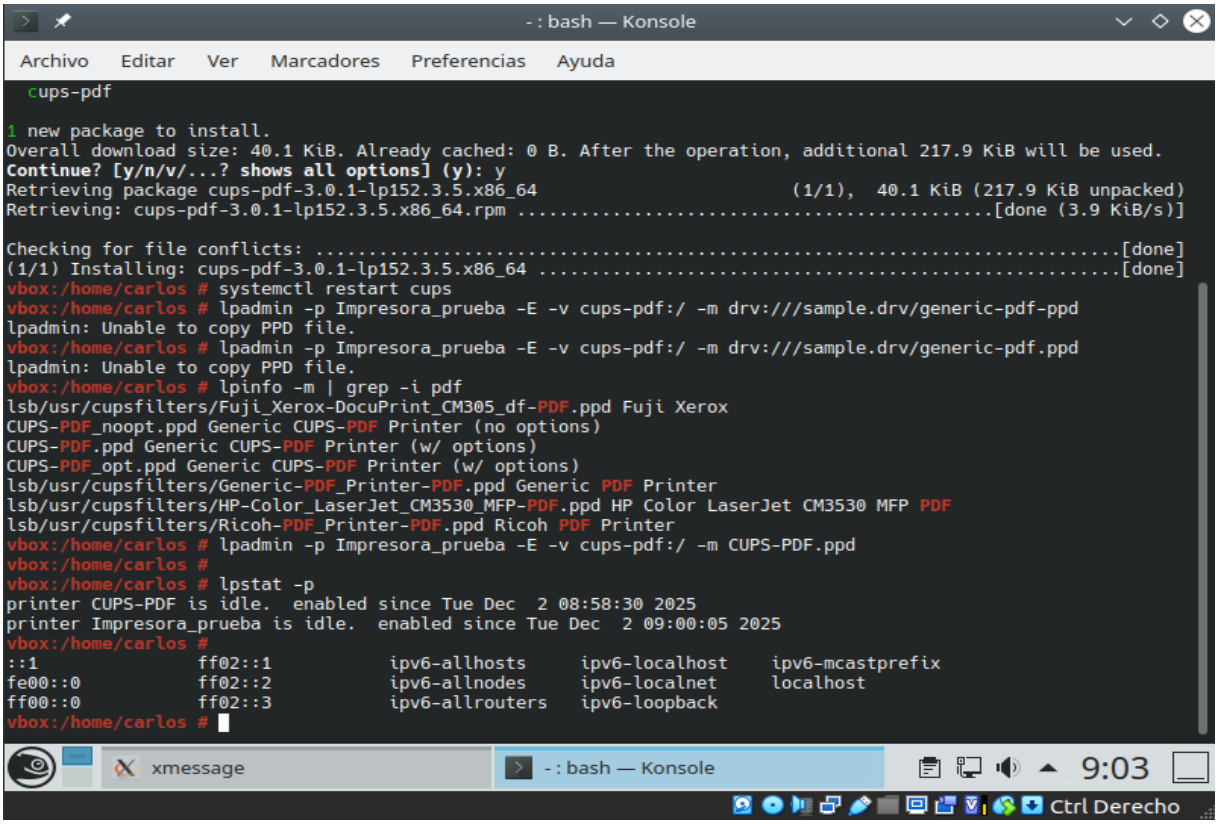
su - "u_supervisor/usuario_estandar"

ls -ld /home/u_privado

Paso 6. Configuración de la Impresión

Primeramente, instalamos Cups para poder utilizar la impresora con **zypper**(gestor de paquetes de OpenSuse).





Una vez instalado, lo reiniciamos para que aplique los nuevos drivers instalados. Ahora utilizaremos **lpadmin** para crear la impresora (como siempre, utilizamos un driver genérico que no es el que necesitamos, por lo que tenemos que buscarlo con **lpinfo -m | grep i pdf**), buscamos el driver que necesitamos (siempre termina por .ppd) en este caso se llama **CUPS-PDF.ppd**, reintentamos añadir la impresora y finalmente nos permite. Por último solo falta comprobar que existe y vemos que esta activado.

Comandos (si somos usuarios root no hace falta usar “sudo”):

 # 1. Instalamos el servidor CUPS y el paquete cups-pdf:

sudo zypper install cups-pdf

 # 3. Reiniciamos el servicio de impresión:

sudo systemctl restart cups

 # 5. Añadimos la impresora PDF a lpadmin (forma correcta):

sudo lpadmin -p Impresora_prueba -E -v cups-pdf:/ -m CUPS-PDF.ppd

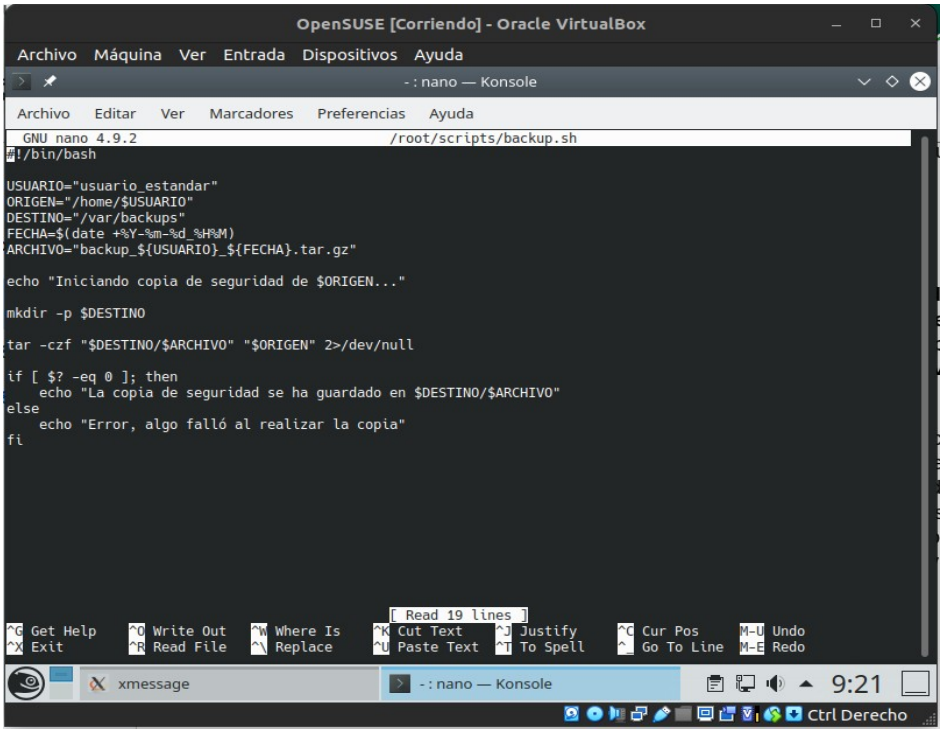
 # 6. Verificamos su estado:

lpstat -p

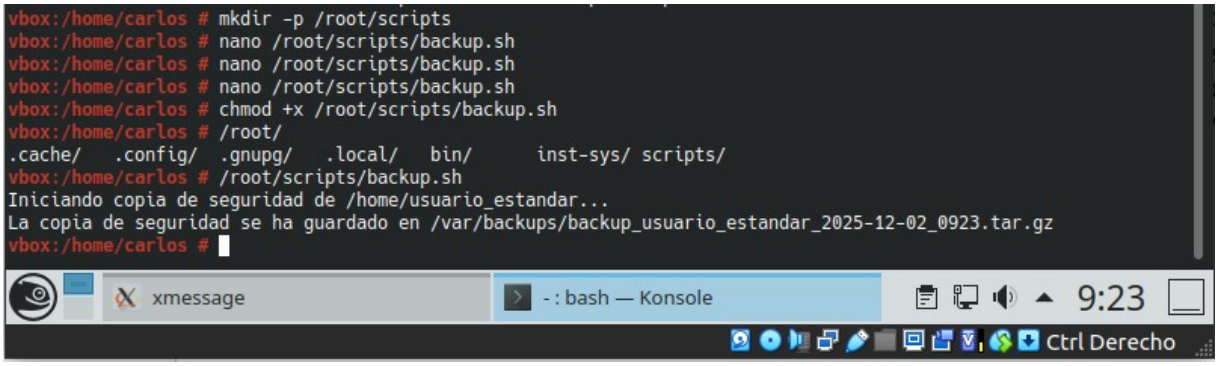
Paso 7. Programación Básica de Shell Script

Primeramente creamos una carpeta donde guardaremos el script dentro de **root**.

Seguidamente, crearemos el archivo con nano dentro de esa carpeta, ese archivo se llamara **backup.sh** (con el nombre que quieras pero que termine por **.sh)**.



Realizaremos un pequeño código que realizara una copia de seguridad al ejecutarlo, en caso se que algo falle, mandara un mensaje de advertencia.



Para finalizar, le otorgaremos permisos al archivo y lo ejecutaremos para comprobar que funcione y finalmente terminamos la práctica de OpenSuse.

Comandos (si somos usuarios root no hace falta usar “sudo”):

 # 1. Crearemos el directorio donde guardar el script:
sudo mkdir -p /root/scripts

 # 2. Crearemos el archivo que realizara la copia de seguridad:
sudo nano /root/scripts/backup.sh

 # 3. Otorgaremos permisos al archivo:
sudo chmod +x /root/scripts/backup.sh

 # 4. Ejecutamos el script:
/root/scripts/backup.sh