



Índice General

Práctica: Administración y Configuración de Sistemas Operativos.....	3
Introducción.....	3
<i>Linux Mint</i>	6
Paso 1. Configuración de Usuarios y Grupos Locales.....	6
Paso 2. Seguridad de Cuentas de Usuario.....	7
Paso 3. Seguridad de Contraseñas.....	11
Paso 4. Gestión del Entorno de Trabajo del Usuario.....	11
Paso 5. Acceso a Recursos y Permisos Locales.....	13
Paso 6. Configuración de la Impresión.....	14
Paso 7. Programación Básica de Shell Script.....	17
<i>Fedora</i>	18
Paso 1. Configuración de Usuarios y Grupos Locales.....	18
Paso 2. Seguridad de Cuentas de Usuario.....	18
Paso 3. Seguridad de Contraseñas.....	21
Paso 4. Gestión del Entorno de Trabajo del Usuario.....	21
Paso 5. Acceso a Recursos y Permisos Locales.....	23
Paso 6. Configuración de la Impresión.....	25
Paso 7. Programación Básica de Shell Script.....	27
<i>OpenSuse</i>	29
Paso 1. Configuración de Usuarios y Grupos Locales.....	29
Paso 2. Seguridad de Cuentas de Usuario.....	29
Paso 3. Seguridad de Contraseñas.....	31
Paso 4. Gestión del Entorno de Trabajo del Usuario.....	31
Paso 5. Acceso a Recursos y Permisos Locales.....	33
Paso 6. Configuración de la Impresión.....	34
Paso 7. Programación Básica de Shell Script.....	36

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.

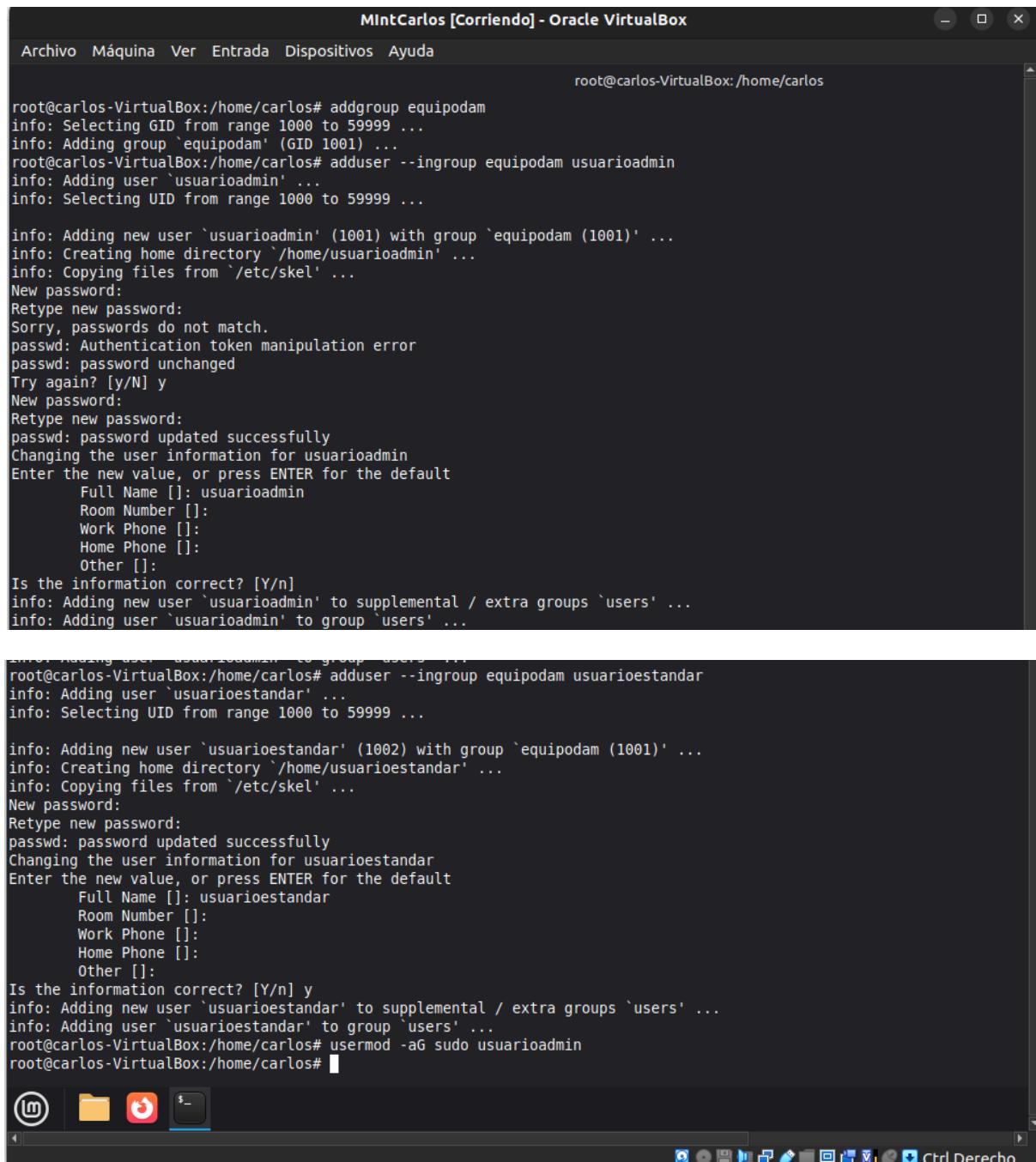
7. Programación Básica de Shell Script

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



The screenshot shows a terminal window titled "MintCarlos [Corriendo] - Oracle VirtualBox". The terminal is running as root on a Linux Mint system. It displays the process of creating two new users: "usuarioadmin" and "usuarioestandar".

Creating User 'usuarioadmin':

```
root@carlos-VirtualBox:/home/carlos# addgroup equipodam
info: Selecting GID from range 1000 to 59999 ...
info: Adding group `equipodam' (GID 1001) ...
root@carlos-VirtualBox:/home/carlos# adduser --ingroup equipodam usuarioadmin
info: Adding user `usuarioadmin' ...
info: Selecting UID from range 1000 to 59999 ...

info: Adding new user `usuarioadmin' (1001) with group `equipodam (1001)' ...
info: Creating home directory `/home/usuarioadmin' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for usuarioadmin
Enter the new value, or press ENTER for the default
  Full Name []: usuarioadmin
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n]
info: Adding new user `usuarioadmin' to supplemental / extra groups `users' ...
info: Adding user `usuarioadmin' to group `users' ...
```

Creating User 'usuarioestandar':

```
root@carlos-VirtualBox:/home/carlos# adduser --ingroup equipodam usuarioestandar
info: Adding user `usuarioestandar' ...
info: Selecting UID from range 1000 to 59999 ...

info: Adding new user `usuarioestandar' (1002) with group `equipodam (1001)' ...
info: Creating home directory `/home/usuarioestandar' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for usuarioestandar
Enter the new value, or press ENTER for the default
  Full Name []: usuarioestandar
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
info: Adding new user `usuarioestandar' to supplemental / extra groups `users' ...
info: Adding user `usuarioestandar' to group `users' ...
root@carlos-VirtualBox:/home/carlos# usermod -aG sudo usuarioadmin
root@carlos-VirtualBox:/home/carlos#
```

The terminal window includes standard Linux Mint icons in the toolbar at the bottom.

```
carlos@carlos-VirtualBox:~$ sudo usermod -aG sudo usuarioadmin
carlos@carlos-VirtualBox:~$ groups usuarioadmin
usuarioadmin : equipodam sudo users
carlos@carlos-VirtualBox:~$ groups usuarioestandar
usuarioestandar : equipodam users
carlos@carlos-VirtualBox:~$ █
```

Como usuarios **root** primeramente creamos un grupo y dos usuarios (**usuarioadmin** y **usuarioestandar**) y 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
```

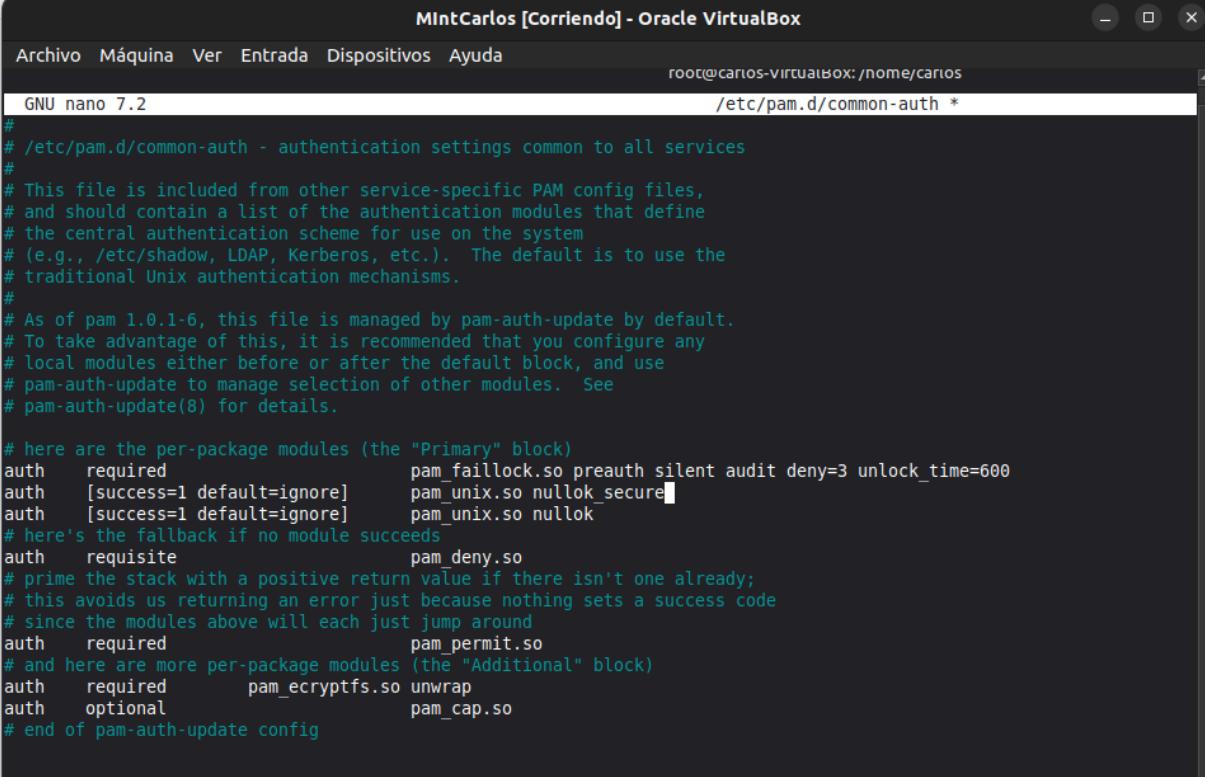
4. Verifica que pertenece a un grupo

```
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 bloqueo debemos acceder a los archivos PAM.

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



```

MintCarlos [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@carlos-virtualbox:/home/carlos
GNU nano 7.2
/etc/pam.d/common-auth *

#
# /etc/pam.d/common-auth - authentication settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.). The default is to use the
# traditional Unix authentication mechanisms.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
auth    required        pam_faillock.so preauth silent audit deny=3 unlock_time=600
auth    [success=1 default=ignore]    pam_unix.so nullok_secure
auth    [success=1 default=ignore]    pam_unix.so nullok
# here's the fallback if no module succeeds
auth    requisite      pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
auth    required        pam_permit.so
# and here are more per-package modules (the "Additional" block)
auth    required        pam_encryptfs.so unwrap
auth    optional       pam_cap.so
# end of pam-auth-update config

```

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 había 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**.

```

MIntCarlos [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@carlos-virtualbox:/home/carlos
GNU nano 7.2
/etc/pam.d/common-auth *

#
# /etc/pam.d/common-auth - authentication settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.). The default is to use the
# traditional Unix authentication mechanisms.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
auth    required          pam_faillock.so preauth silent audit deny=3 unlock_time=600
auth    [success=1 default=ignore]      pam_unix.so nullok_secure
auth    [default=die]            pam_faillock.so authfail audit deny=3 unlock_time=600
# here's the fallback if no module succeeds
auth    requisite          pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
auth    required          pam_permit.so
# and here are more per-package modules (the "Additional" block)
auth    required          pam_encryptfs.so unwrap
auth    optional           pam_cap.so
# end of pam-auth-update config

```

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").

```

MIntCarlos [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@carlos-virtualbox:/home/carlos
GNU nano 7.2
/etc/pam.d/common-account

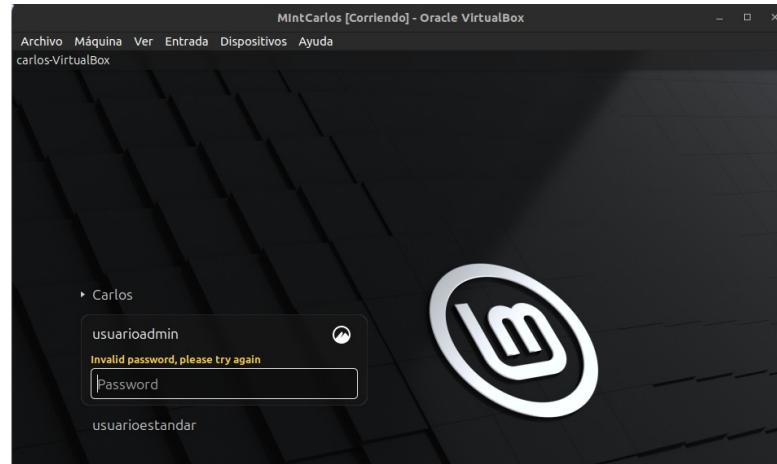
#
# /etc/pam.d/common-account - authorization settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authorization modules that define
# the central access policy for use on the system. The default is to
# only deny service to users whose accounts are expired in /etc/shadow.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.

#
# here are the per-package modules (the "Primary" block)
account required          pam_faillock.so
account [success=1 new_authtok_reqd=done default=ignore]      pam_unix.so
# here's the fallback if no module succeeds
account requisite          pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
account required          pam_permit.so
# and here are more per-package modules (the "Additional" block)
# end of pam-auth-update config

```

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

```
carlos@carlos-VirtualBox:~$ su usuarioestandar
Password:
su: Authentication failure
carlos@carlos-VirtualBox:~$ su usuarioestandargured300
su: user usuarioestandargured300 does not exist or the user entry does not contain all the required fields
carlos@carlos-VirtualBox:~$ su usuarioestandar
Password:
su: Authentication failure
carlos@carlos-VirtualBox:~$
```



Paso 3. Seguridad de Contraseñas

```
carlos@carlos-VirtualBox:~$ sudo su
root@carlos-VirtualBox:/home/carlos# chage -M 30 usuarioestandar
root@carlos-VirtualBox:/home/carlos# chage -l usuarioestandar
Último cambio de contraseña : dic 02, 2025
La contraseña caduca : ene 01, 2026
Contraseña inactiva : nunca
La cuenta caduca : nunca
Número de días mínimo entre cambio de contraseña : 0
Número de días máximo entre cambio de contraseña : 30
Número de días de aviso antes de que caduque la contraseña : 7
root@carlos-VirtualBox:/home/carlos#
```

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 “**Islarga**” 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 Islarga='ls -latr'
```

```
export CURSO="Sistemasinformaticosmint"
```

```

GNU nano 7.2
root@carlos-VirtualBox: /home/carlos
/home/usuarioestandar/.bashrc

alias fgrep='grep --color=auto'
alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands.  Use like so:
# sleep 10; alert
alias alert="notify-send --urgency=low -i \"$([ $? = 0 ] && echo terminal || echo error)\" \"$(history 1)\"

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/bash_aliases ]; then
. ~/bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -q posix; then
if [ -f /usr/share/bash-completion/bash_completion ]; then
. /usr/share/bash-completion/bash_completion
elif [ -f /etc/bash_completion ]; then
. /etc/bash_completion
fi
fi

alias lslarga='ls -lastr'
export CURSO="Sistemasinformaticosmint"

```

Y ya por último, lo comprobamos:

```

usuarioestandar@carlos-VirtualBox: ~
root@carlos-VirtualBox:/home/carlos# nano /home/usuarioestandar/.bashrc
root@carlos-VirtualBox:/home/carlos# su - usuarioestandar
usuarioestandar@carlos-VirtualBox:~$ lslarga
total 40
4 drwxr-xr-x 5 root      root      4096 dic  2 13:28 ..
4 drwxr-xr-x 3 usuarioestandar equipodam 4096 dic  2 13:28 .local
4 -rw-r--r-- 1 usuarioestandar equipodam  220 dic  2 13:28 .bash_logout
4 -rw-r--r-- 1 usuarioestandar equipodam  516 dic  2 13:28 .gtkrc-xfce
4 -rw-r--r-- 1 usuarioestandar equipodam  807 dic  2 13:28 .profile
4 drwxr-xr-x 3 usuarioestandar equipodam 4096 dic  2 13:28 .config
4 -rw-r--r-- 1 usuarioestandar equipodam   22 dic  2 13:28 .gtkrc-2.0
4 -rw-r--r-- 1 usuarioestandar equipodam 3838 dic  2 13:47 .bashrc
4 drwx----- 2 usuarioestandar equipodam 4096 dic  2 13:47 .cache
4 drwxr-x--- 5 usuarioestandar equipodam 4096 dic  2 13:47 .
usuarioestandar@carlos-VirtualBox:~$ echo $CURSO
Sistemasinformaticosmint
usuarioestandar@carlos-VirtualBox:~$ 

```

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

1. Accedemos 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:

Islarga

```
echo $CURSO
```

Paso 5. Acceso a Recursos y Permisos Locales

```
u_supervisor@carlos-VirtualBox: ~
root@carlos-VirtualBox:/home/carlos# mkdir -p /srv/recursos_dam
root@carlos-VirtualBox:/home/carlos# chgrp equipodam /srv/recursos_dam
root@carlos-VirtualBox:/home/carlos# chmod 770 /srv/recursos_dam
root@carlos-VirtualBox:/home/carlos# ls -ld /srv/recursos_dam
drwxrwx--- 2 root equipodam 4096 dic  2 13:49 /srv/recursos_dam
root@carlos-VirtualBox:/home/carlos# useradd -m -s /bin/bash u_privado
root@carlos-VirtualBox:/home/carlos# useradd -m -s /bin/bash u_supervisor
root@carlos-VirtualBox:/home/carlos# passwd u_privado
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
root@carlos-VirtualBox:/home/carlos# passwd u_supervisor
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
root@carlos-VirtualBox:/home/carlos# chmod 700 /home/u_privado
root@carlos-VirtualBox:/home/carlos# usermod -aG u_privado u_supervisor
Orden «usermod» no encontrada. Quizá quiso decir:
  la orden «usermod» del paquete deb «passwd (1:4.13+dfsg1-4ubuntu3.2)»
Pruebe con: apt install <nombre del paquete deb>
root@carlos-VirtualBox:/home/carlos# usermod -aG u_privado u_supervisor
root@carlos-VirtualBox:/home/carlos# chmod 750 /home/u_privado
root@carlos-VirtualBox:/home/carlos# su - u_supervisor
u_supervisor@carlos-VirtualBox:~$ ls -ld /home/u-privado
ls: no se puede acceder a '/home/u-privado': No existe el archivo o el directorio
u_supervisor@carlos-VirtualBox:~$ ls -ld /home/u-privado
```

Parte 1:

Primeramente creamos un directorio (carpeta) donde únicamente el grupo **equipodam** (asignandolo como 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 usuario supervisor al grupo de "**u_privado**" (que ya existe). Y ahora sí permitimos los permisos para el grupo con **750** (permitiendo que pueda acceder y leer).

```
carlos@carlos-VirtualBox:~$ su - u_supervisor
Contraseña:
u_supervisor@carlos-VirtualBox:~$ ls -ld /home/u_privado
drwxr-x--- 4 u_privado u_privado 4096 dic  2 13:51 /home/u_privado
u_supervisor@carlos-VirtualBox:~$ exit
cerrar sesión
carlos@carlos-VirtualBox:~$ su - usuarioestandar
Contraseña:
usuarioestandar@carlos-VirtualBox:~$ ls -ld /home/u_privado
drwxr-x--- 4 u_privado u_privado 4096 dic  2 13:51 /home/u_privado
usuarioestandar@carlos-VirtualBox:~$ ls /home/u_privado
ls: no se puede abrir el directorio '/home/u_privado': Permiso denegado
usuarioestandar@carlos-VirtualBox:~$
```

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# 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 funcionó y tuve que encontrar una solución para ello.

```

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# lpinfo -m | grep -i "cups-pdf.ppd"
root@carlos-VirtualBox:/home/carlos# lpinfo -m | grep pdf
lsb/usr/cupsfilters/Fuji Xerox DocuPrint CM305 df-PDF.ppd Fuji Xerox DocuPrint CM305 df PDF
lsb/usr/cups-pdf/CUPS-PDF_nopt.ppd Generic CUPS-PDF Printer (no options)
lsb/usr/cups-pdf/CUPS-PDF_opt.ppd Generic CUPS-PDF Printer (w/ options)
lsb/usr/cupsfilters/Generic-PDF.ppd Generic PDF Printer
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-C7521n PDF.ppd Gestetner C7521n PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-C7526dn PDF.ppd Gestetner C7526dn PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-C7531dn PDF.ppd Gestetner C7531dn PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-C7640nD PDF.ppd Gestetner C7640nD PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-C8140nD PDF.ppd Gestetner C8140nD PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-C8150nD PDF.ppd Gestetner C8150nD PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-CS555 PDF.ppd Gestetner CS555 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc424 PDF.ppd Gestetner DSc424 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc432 PDF.ppd Gestetner DSc432 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc460 PDF.ppd Gestetner DSc460 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1028 PDF.ppd Gestetner DSc1028 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1029 PDF.ppd Gestetner DSc1029 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1030 PDF.ppd Gestetner DSc1030 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1045 PDF.ppd Gestetner DSc1045 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1060 PDF.ppd Gestetner DSc1060 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1120 PDF.ppd Gestetner DSc1120 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1220 PDF.ppd Gestetner DSc1220 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1220ex PDF.ppd Gestetner DSc1220ex PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1225 PDF.ppd Gestetner DSc1225 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1225ex PDF.ppd Gestetner DSc1225ex PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1230 PDF.ppd Gestetner DSc1230 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1230ex PDF.ppd Gestetner DSc1230ex PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1245 PDF.ppd Gestetner DSc1245 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1245ex PDF.ppd Gestetner DSc1245ex PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1260 PDF.ppd Gestetner DSc1260 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSc1260ex PDF.ppd Gestetner DSc1260ex PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSm_2625 PDF.ppd Gestetner DSm 2625 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSm_2630 PDF.ppd Gestetner DSm 2630 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSm_2635 PDF.ppd Gestetner DSm 2635 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSm_2640 PDF.ppd Gestetner DSm 2640 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSm_2650 PDF.ppd Gestetner DSm 2650 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSm_2660 PDF.ppd Gestetner DSm 2660 PDF
openprinting-pdps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSm725 PDF.ppd Gestetner DSm725 PDF
openprinting-nodps:0/ffd/openprinting/gestetner/PDF/Gestetner-DSm730 PDF.nod Gestetner DSm730 PDF

```

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 **lpadmin** de nuevo con el nuevo nombre y verificamos que no da error y que aparece en la lista.

```

MIntCarlos parte 2 [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@carlos-VirtualBox:/home/carlos# lpadmin -p Impresora_PDF_Mint -E -v cups-pdf:/ -m lsb/usr/cups-pdf/CUPS-PDF_opt.ppd
lpadmin: Printer drivers are deprecated and will stop working in a future version of CUPS.
root@carlos-VirtualBox:/home/carlos# lpstat -p
la impresora Impresora_PDF_Mint está inactiva. activada desde mar 02 dic 2025 14:20:50
la impresora PDF está inactiva. activada desde mar 02 dic 2025 14:12:07
root@carlos-VirtualBox:/home/carlos# systemctl status cups
● cups.service - CUPS Scheduler
  Loaded: loaded (/usr/lib/systemd/system/cups.service; enabled; preset: enabled)
  Active: active (running) since Tue 2025-12-02 14:12:07 CET; 9min ago
    TriggeredBy: ● cups.socket
      ● cups.path
     Docs: man:cupsd(8)
   Main PID: 3282 (cupsd)
     Status: "Scheduler is running..."
    Tasks: 3 (limit: 2255)
   Memory: 20.1M (peak: 161.3M)
      CPU: 2.013s
     CGroup: /system.slice/cups.service
             └─3282 /usr/sbin/cupsd -l
                 ├─3303 /usr/lib/cups/notifier/dbus dbus://
                 ├─3304 /usr/lib/cups/notifier/dbus dbus://
dic 02 14:12:07 carlos-VirtualBox systemd[1]: Starting cups.service - CUPS Scheduler...
dic 02 14:12:07 carlos-VirtualBox systemd[1]: Started cups.service - CUPS Scheduler.
root@carlos-VirtualBox:/home/carlos#

```

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 activo).

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 “Ipadmin”:

```
sudo usermod -aG Ipadmin usuario_admin
```

3. Habilitamos el servicio de impresión para futuros arranques:

```
sudo systemctl enable --now cups
```

4. Añadimos la impresora PDF a Ipadmin (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



```
root@carlos-VirtualBox:/home/carlos
[carlos@carlos-VirtualBox:~$ sudo su
[sudo] contraseña para carlos:
root@carlos-VirtualBox:/home/carlos# sudo mkdir -p /root/scripts
root@carlos-VirtualBox:/home/carlos# nano /root/scripts/backup_mint.sh
root@carlos-VirtualBox:/home/carlos# chmod +x /root/
.bash_history .bashrc .cache/.lessht .local/.profile scripts/.ssh/
root@carlos-VirtualBox:/home/carlos# chmod +x /root/scripts/backup_mint.sh
root@carlos-VirtualBox:/home/carlos# /root/scripts/backup_mint.sh
>>> Iniciando copia de seguridad en Linux Mint...
Copia guardada en /var/backups/MINT/backup_usuarioestandar_2025-12-03_0848.tar.gz
root@carlos-VirtualBox:/home/carlos# ]
```

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**).

```
root@carlos-VirtualBox:/home/carlos
/root/scripts/backup_mint.sh *

GNU nano 7.2
#!/bin/bash

USUARIO="usuarioestandar"
ORIGEN="/home/$USUARIO"
DESTINO="/var/backups/MINT"
FECHA=$(date +%Y-%m-%d %H%M)
ARCHIVO="backup_${USUARIO}_${FECHA}.tar.gz"

echo ">>> Iniciando copia de seguridad en Linux Mint..."
mkdir -p $DESTINO
tar -czf "$DESTINO/$ARCHIVO" "$ORIGEN" 2>/dev/null

if [ $? -eq 0 ]; then
    echo "Copia guardada en $DESTINO/$ARCHIVO"
else
    echo "Falló la copia de seguridad"
fi
```

Realizaremos un pequeño código que realizara una copia de seguridad al ejecutarlo, en caso de 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.

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_mint.sh
```

3. Otorgaremos permisos al archivo:

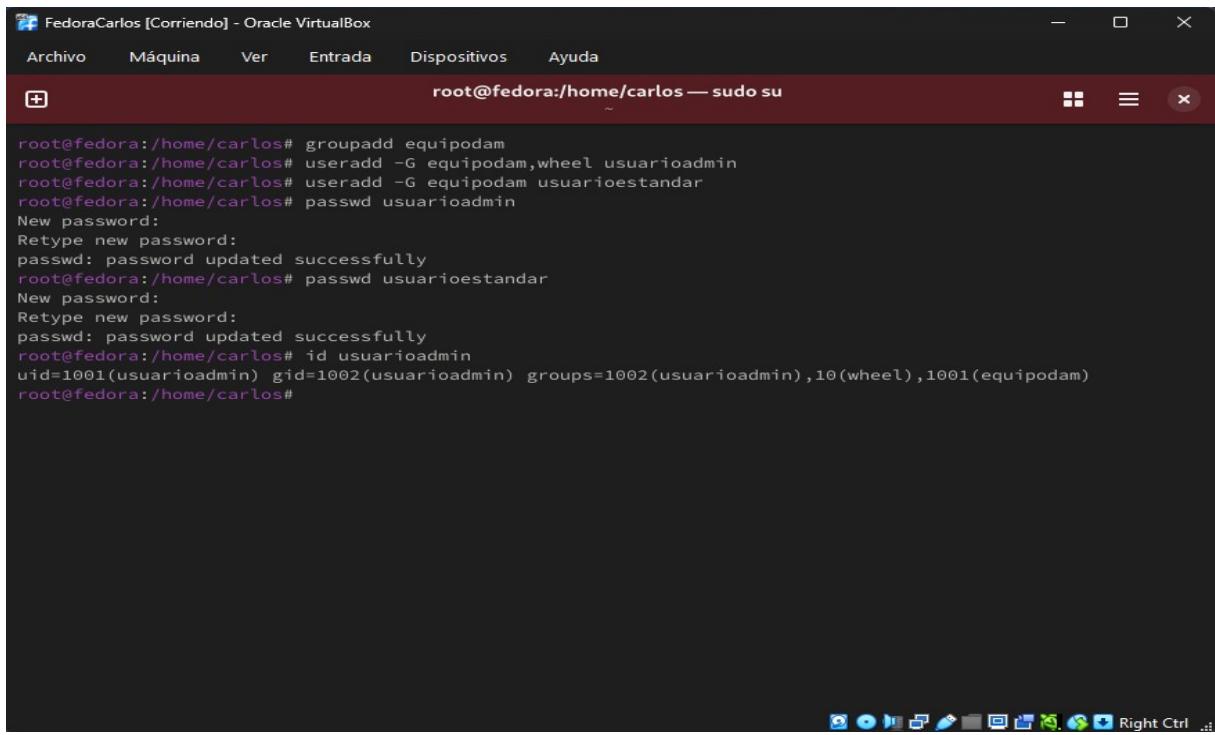
```
sudo chmod +x /root/scripts/backup_mint.sh
```

4. Verificamos que se haya realizado correctamente el backup:

```
ls -lh /var/backups/MINT
```

Fedora

Paso 1. Configuración de Usuarios y Grupos Locales



The screenshot shows a terminal window titled "FedoraCarlos [Corriendo] - Oracle VirtualBox". The window has a dark theme with a red header bar. The title bar includes the window name, the host name "root@fedora:/home/carlos — sudo su", and standard window control buttons. The main area of the terminal displays the following command-line session:

```
root@fedora:/home/carlos# groupadd equipodam
root@fedora:/home/carlos# useradd -G equipodam,wheel usuarioadmin
root@fedora:/home/carlos# useradd -G equipodam usuarioestandar
root@fedora:/home/carlos# passwd usuarioadmin
New password:
Retype new password:
passwd: password updated successfully
root@fedora:/home/carlos# passwd usuarioestandar
New password:
Retype new password:
passwd: password updated successfully
root@fedora:/home/carlos# id usuarioadmin
uid=1001(usuarioadmin) gid=1002(usuarioadmin) groups=1002(usuarioadmin),10(wheel),1001(equipodam)
root@fedora:/home/carlos#
```

The terminal window is set against a desktop background with various icons in the taskbar.

Como usuarios **root** primeramente creamos el grupo "**equipodam**" 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"
```

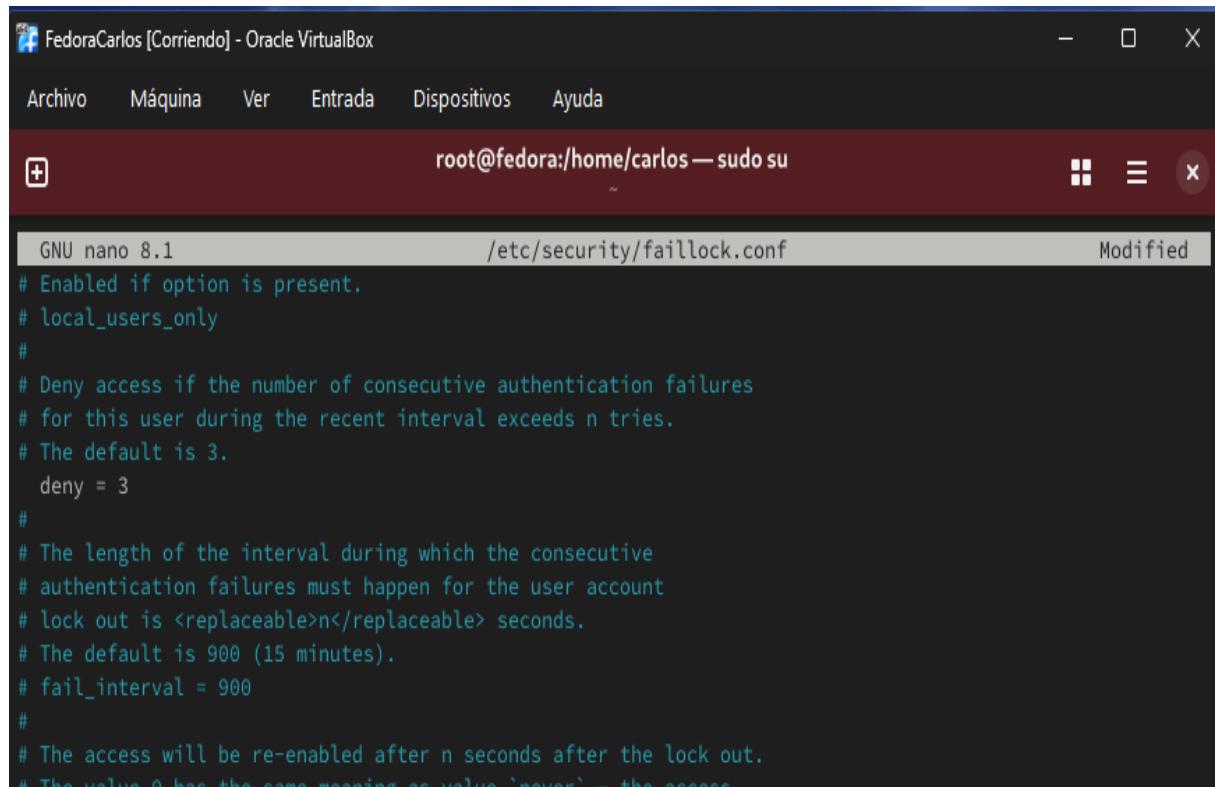
4. Verifica que pertenece a un grupo

```
sudo id "usuarioadmin/usuarioestandar"
```

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

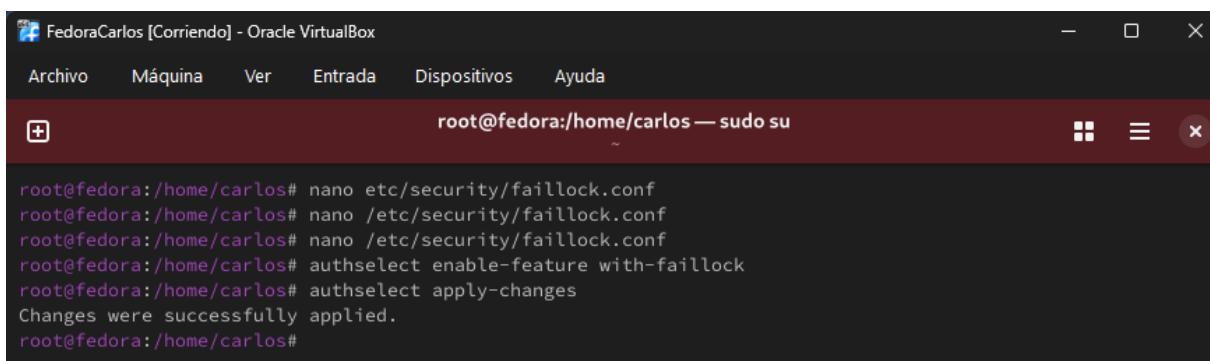
Para poder modificar la política de bloqueo debemos acceder a los archivo PAM.

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



```
GNU nano 8.1          /etc/security/faillock.conf      Modified
# Enabled if option is present.
# local_users_only
#
# Deny access if the number of consecutive authentication failures
# for this user during the recent interval exceeds n tries.
# The default is 3.
deny = 3
#
# The length of the interval during which the consecutive
# authentication failures must happen for the user account
# lock out is <replaceable>n</replaceable> seconds.
# The default is 900 (15 minutes).
# fail_interval = 900
#
# The access will be re-enabled after n seconds after the lock out.
# The value 0 has the same meaning as value 'never' - the access
```

Una vez dentro del archivo, lo único que tenemos que hacer es descomentar la línea “**#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**), guardar el archivo y aplicarlo los cambios con el comando **authselect apply-changes**.



```
root@fedora:/home/carlos# nano etc/security/faillock.conf
root@fedora:/home/carlos# nano /etc/security/faillock.conf
root@fedora:/home/carlos# nano /etc/security/faillock.conf
root@fedora:/home/carlos# authselect enable-feature with-faillock
root@fedora:/home/carlos# authselect apply-changes
Changes were successfully applied.
root@fedora:/home/carlos#
```

The screenshot shows a terminal window titled "FedoraCarlos [Corriendo] - Oracle VirtualBox". The terminal session starts with several failed password attempts:

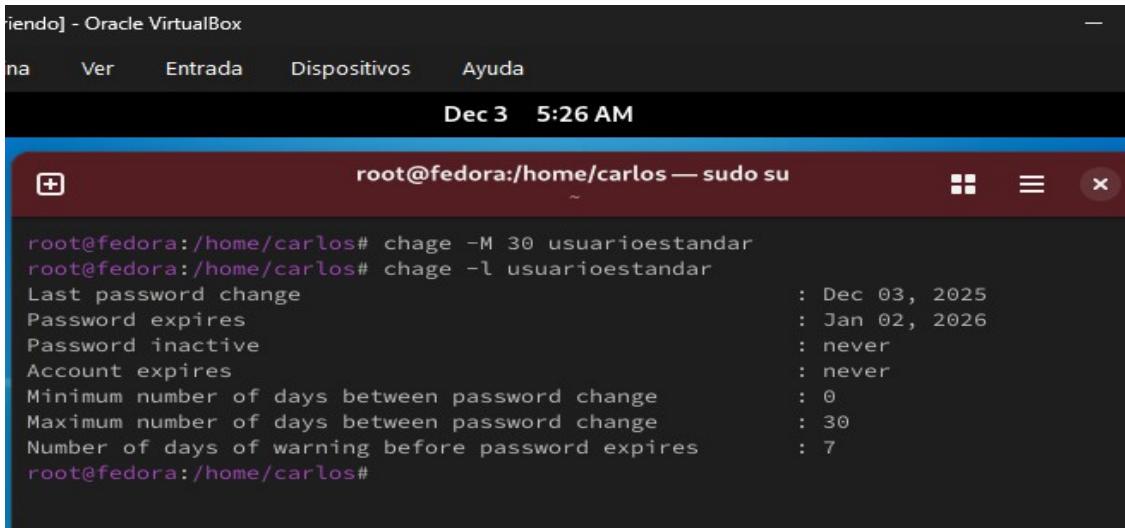
```
carlos@fedora:~$ su usuarioestandar
Password:
su: Authentication failure
carlos@fedora:~$ faillock --user usuarioestandar
usuarioestandar:
When          Type  Source          Valid
2025-12-03 05:22:10 TTY   /dev/pts/0      V
2025-12-03 05:22:16 TTY   /dev/pts/0      V
2025-12-03 05:22:23 TTY   /dev/pts/0      V
carlos@fedora:~$ faillock --user usuarioestandar --reset
carlos@fedora:~$ su usuarioestandar
Password:
usuarioestandar@fedora:/home/carlos$
```

After running `faillock --user usuarioestandar --reset`, the user is successfully logged in again.

Mediante la verificación (intentamos acceder al `usuarioestandar`) comprobamos que luego del tercer intento, al insertar la contraseña correcta, de igual manera nos 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



```
root@fedora:/home/carlos# chage -M 30 usuarioestandar
root@fedora:/home/carlos# chage -l usuarioestandar
Last password change : Dec 03, 2025
Password expires       : Jan 02, 2026
Password inactive     : never
Account expires        : never
Minimum number of days between password change : 0
Maximum number of days between password change : 30
Number of days of warning before password expires: 7
root@fedora:/home/carlos#
```

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 “**Islarga**” 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 Islarga='ls -latr'
```

```
export Groupfedora="Sistemasinformaticosfedora"
```

```

GNU nano 8.1          /home/usuarioestandar/.bashrc      Modified
if ! [[ "$PATH" =~ "$HOME/.local/bin:$HOME/bin:" ]]; then
    PATH="$HOME/.local/bin:$HOME/bin:$PATH"
fi
export PATH

# Uncomment the following line if you don't like systemctl's auto-paging feature
# export SYSTEMD_PAGER=

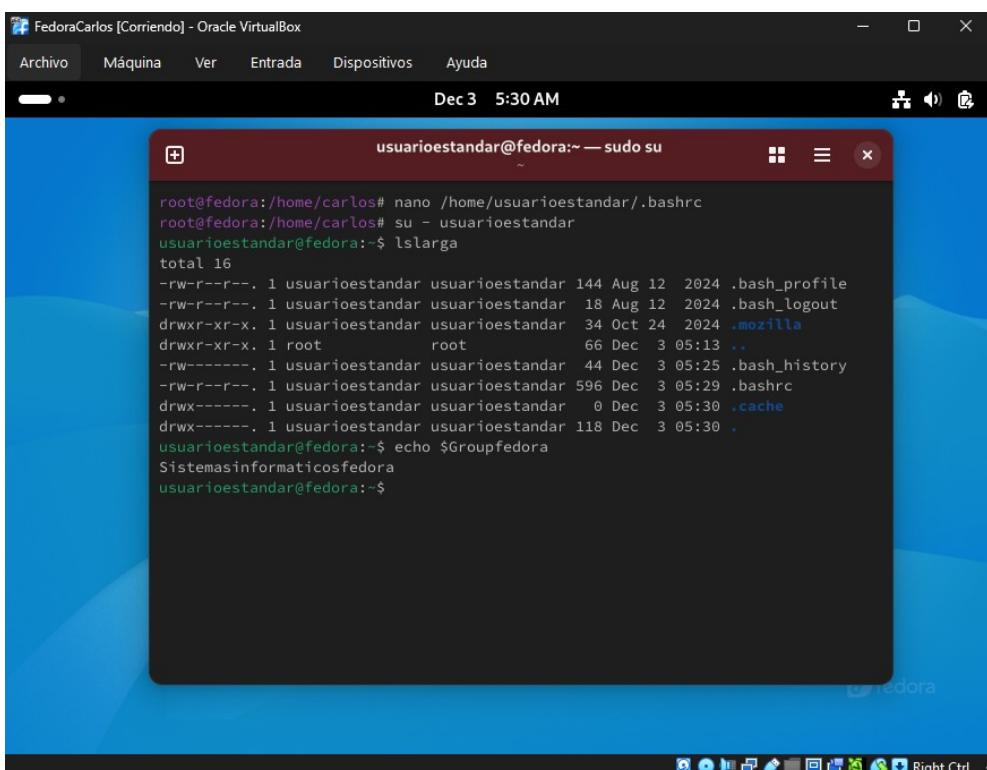
# User specific aliases and functions
if [ -d ~/.bashrc.d ]; then
    for rc in ~/.bashrc.d/*; do
        if [ -f "$rc" ]; then
            . "$rc"
        fi
    done
fi
unset rc

alias lslarga='ls -latr'
export Groupfedora="Sistemasinformaticosfedora"

^G Help      ^O Write Out ^F Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste   ^J Justify  ^/ Go To Line

```

Y ya por último, lo comprobamos:



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

1. Accedemos 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:

Islarga

echo \$Groupfedora

Paso 5. Acceso a Recursos y Permisos Locales



```
root@fedora:/home/carlos — sudo su
root@fedora:/home/carlos# mkdir -p /srv/recursos_dam
root@fedora:/home/carlos# chgrp equipodam /srv/recursos_dam
root@fedora:/home/carlos# chmod 770 /srv/recursos_dam
root@fedora:/home/carlos# ls -ld /srv/recursos_dam
drwxrwx---. 1 root equipodam 0 Dec  3 05:34 /srv/recursos_dam
```

Parte 1:

Primeramente creamos un directorio (carpeta) donde únicamente el grupo **equipodam** (asignado como 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:

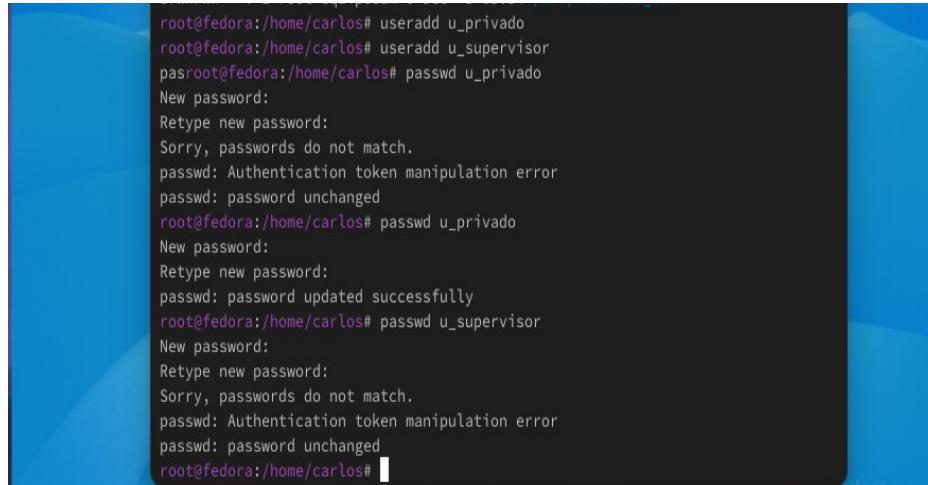
```
sudo chmod 770 /srv/recursos_dam
```

4. Verificamos los permisos:

```
ls -ld /srv/recursos_dam
```

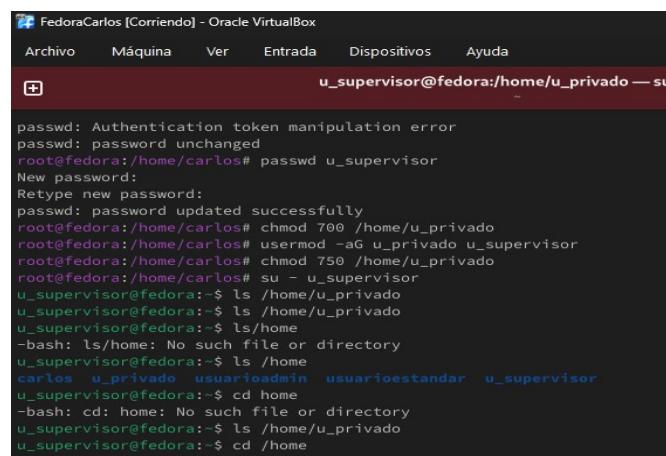
Parte 2:

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.

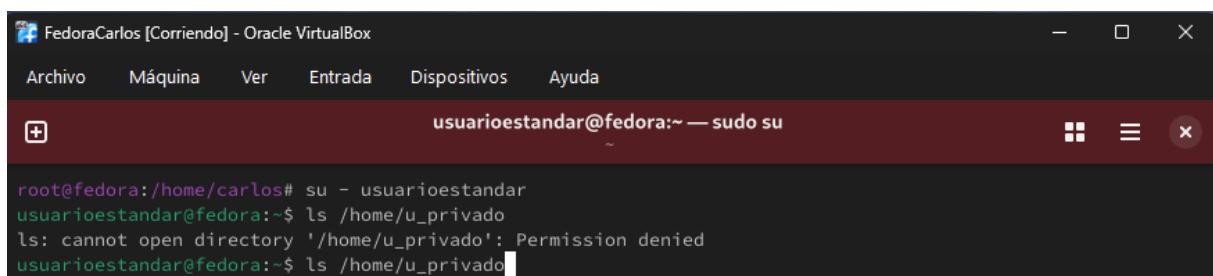


```
root@fedora:/home/carlos# useradd u_privado
root@fedora:/home/carlos# useradd u_supervisor
pasroot@fedora:/home/carlos# passwd u_privado
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
root@fedora:/home/carlos# passwd u_privado
New password:
Retype new password:
passwd: password updated successfully
root@fedora:/home/carlos# passwd u_supervisor
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
root@fedora:/home/carlos#
```

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).



```
passwd: Authentication token manipulation error
passwd: password unchanged
root@fedora:/home/carlos# passwd u_supervisor
New password:
Retype new password:
passwd: password updated successfully
root@fedora:/home/carlos# chmod 700 /home/u_privado
root@fedora:/home/carlos# usermod -aG u_privado u_supervisor
root@fedora:/home/carlos# chmod 750 /home/u_privado
root@fedora:/home/carlos# su - u_supervisor
u_supervisor@fedora: ~ ls /home/u_privado
u_supervisor@fedora: ~ ls /home/u_privado
u_supervisor@fedora: ~ ls /home
-bash: ls /home: No such file or directory
u_supervisor@fedora: ~ ls /home
carlos u_privado usuarioadmin usuarioestandar u_supervisor
u_supervisor@fedora: ~ cd home
-bash: cd: home: No such file or directory
u_supervisor@fedora: ~ ls /home/u_privado
u_supervisor@fedora: ~ cd /home
```



```
root@fedora:/home/carlos# su - usuarioestandar
usuarioestandar@fedora: ~ ls /home/u_privado
ls: cannot open directory '/home/u_privado': Permission denied
usuarioestandar@fedora: ~ ls /home/u_privado
```

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 las 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 diferencias 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
```

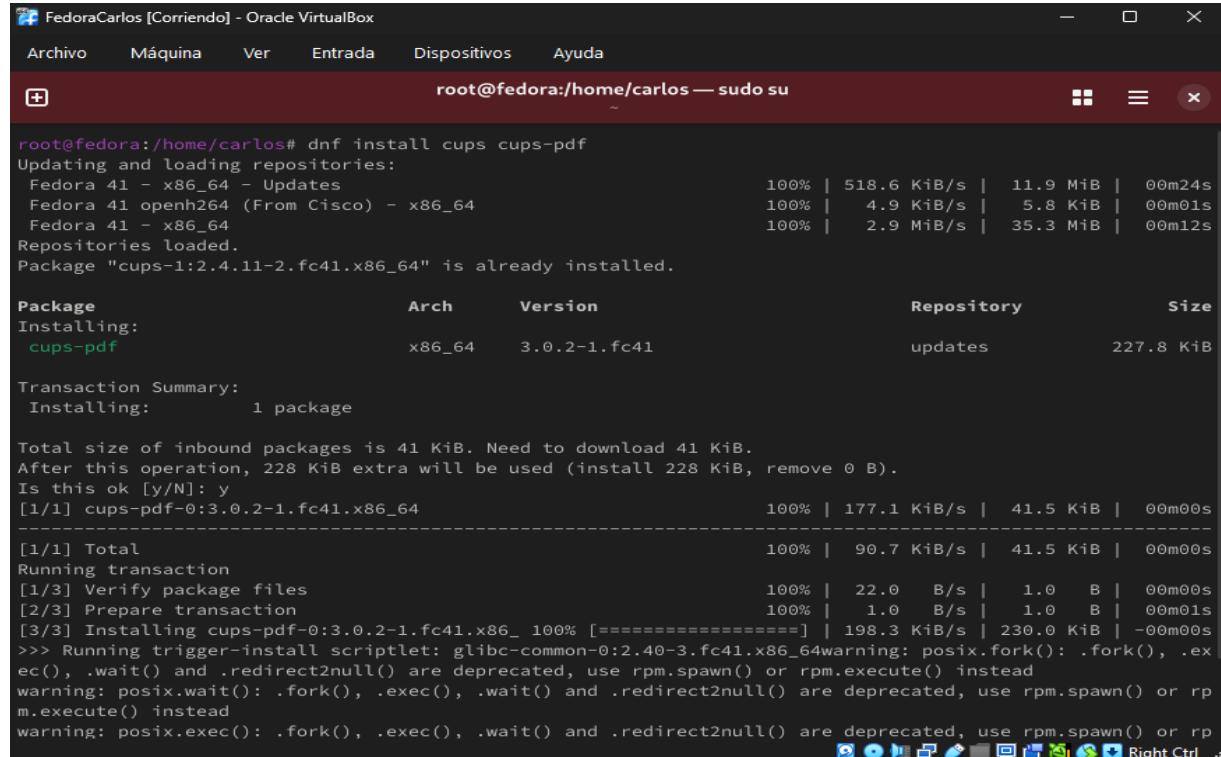
5. Realizamos la comprobación necesaria:

```
su - "u_supervisor/usuarioestandar"
```

```
ls /home/u_privado
```

Paso 6. Configuración de la Impresión

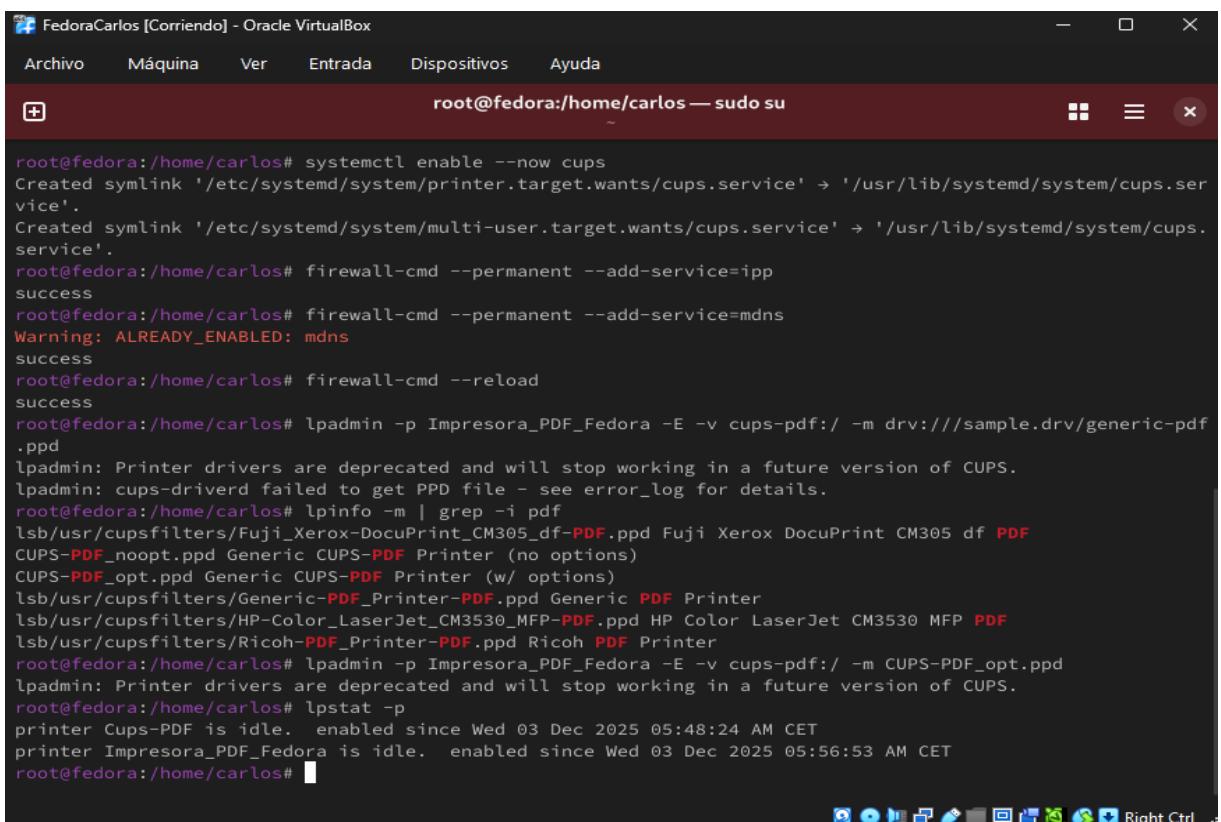
Primeramente, instalamos Cups para poder utilizar la impresora con **dnf**(gestor de paquetes de Fedora).



```
root@fedora:/home/carlos# dnf install cups cups-pdf
Updating and loading repositories:
Fedora 41 - x86_64 - Updates                               100% | 518.6 KiB/s | 11.9 MiB | 00m24s
Fedora 41 openh264 (From Cisco) - x86_64                  100% | 4.9 KiB/s | 5.8 KiB | 00m01s
Fedora 41 - x86_64                                         100% | 2.9 MiB/s | 35.3 MiB | 00m12s
Repositories loaded.
Package "cups-1:2.4.11-2.fc41.x86_64" is already installed.

Transaction Summary:
Installing:         1 package

Total size of inbound packages is 41 KiB. Need to download 41 KiB.
After this operation, 228 KiB extra will be used (install 228 KiB, remove 0 B).
Is this ok [y/N]: y
[1/1] cups-pdf-0:3.0.2-1.fc41.x86_64                      100% | 177.1 KiB/s | 41.5 KiB | 00m00s
-----[1/1] Total                                         100% | 90.7 KiB/s | 41.5 KiB | 00m00s
Running transaction
[1/3] Verify package files                                100% | 22.0 B/s | 1.0 B | 00m00s
[2/3] Prepare transaction                                 100% | 1.0 B/s | 1.0 B | 00m01s
[3/3] Installing cups-pdf-0:3.0.2-1.fc41.x86_64_ 100% [=====] | 198.3 KiB/s | 230.0 KiB | -00m00s
>>> Running trigger-install scriptlet: glibc-common-0:2.40-3.fc41.x86_64warning: posix.fork(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
warning: posix.wait(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
warning: posix.exec(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
```



```
root@fedora:/home/carlos# systemctl enable --now cups
Created symlink '/etc/systemd/system/printer.target.wants/cups.service' → '/usr/lib/systemd/system/cups.service'.
Created symlink '/etc/systemd/system/multi-user.target.wants/cups.service' → '/usr/lib/systemd/system/cups.service'.
root@fedora:/home/carlos# firewall-cmd --permanent --add-service=ipp
success
root@fedora:/home/carlos# firewall-cmd --permanent --add-service=mdns
Warning: ALREADY_ENABLED: mdns
success
root@fedora:/home/carlos# firewall-cmd --reload
success
root@fedora:/home/carlos# lpadmin -p Impresora_PDF_Fedora -E -v cups-pdf:/ -m drv:///sample.drv/generic-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@fedora:/home/carlos# lpinfo -m | grep -i pdf
lsb/usr/cupsfilters/Fuji_Xerox-DocuPrint_CM305_df-PDF.ppd Fuji Xerox DocuPrint CM305 df PDF
CUPS-PDF_noopt.ppd Generic CUPS-PDF Printer (no options)
CUPS-PDF_opt.ppd Generic CUPS-PDF Printer (w/ options)
lsb/usr/cupsfilters/Generic-PDF_Printer-PDF.ppd Generic PDF Printer
lsb/usr/cupsfilters/HP-Color_LaserJet_CM3530_MFP-PDF.ppd HP Color LaserJet CM3530 MFP PDF
lsb/usr/cupsfilters/Ricoh-PDF_Printer-PDF.ppd Ricoh PDF Printer
root@fedora:/home/carlos# lpadmin -p Impresora_PDF_Fedora -E -v cups-pdf:/ -m CUPS-PDF_opt.ppd
lpadmin: Printer drivers are deprecated and will stop working in a future version of CUPS.
root@fedora:/home/carlos# lpstat -p
printer Cups-PDF is idle. enabled since Wed 03 Dec 2025 05:48:24 AM CET
printer Impresora_PDF_Fedora is idle. enabled since Wed 03 Dec 2025 05:56:53 AM CET
root@fedora:/home/carlos#
```

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 obviamente no funcionó) por lo que tuve que buscar el driver específico cuyo nombre por el que tenía 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
```

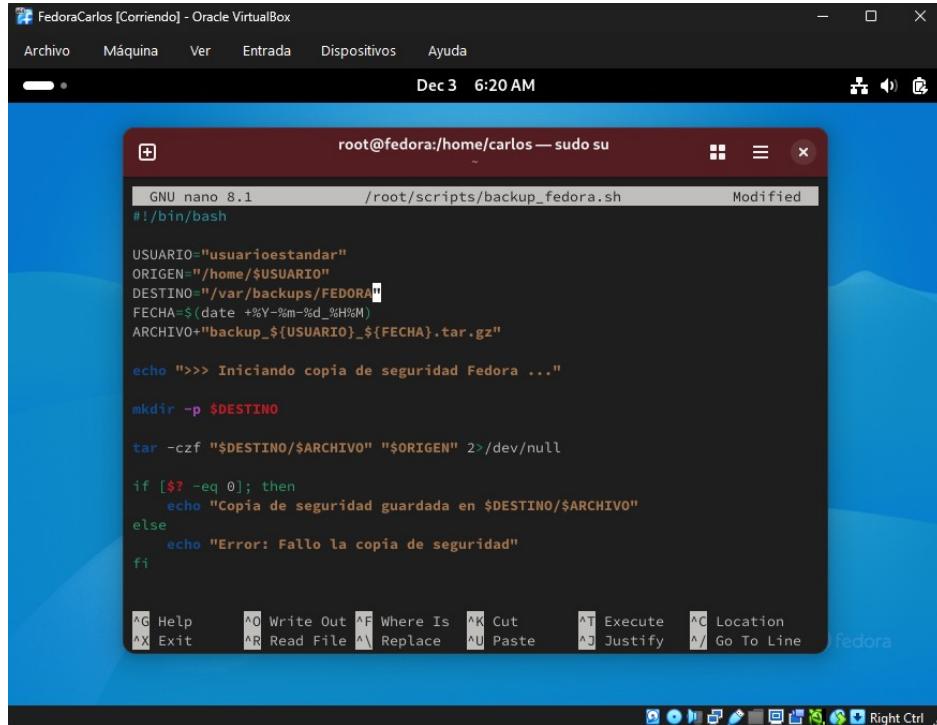
6. Verificamos su estado:

```
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 llamará **backup_fedora.sh** (con el nombre que quieras pero que termine por **.sh**).



```
GNU nano 8.1      /root/scripts/backup_fedora.sh      Modified
#!/bin/bash

USUARIO="usuarioestandar"
ORIGEN="/home/$USUARIO"
DESTINO="/var/backups/FEDORA"
FECHA=$(date +%Y-%m-%d_%H%M)
ARCHIVO+="backup_${USUARIO}_${FECHA}.tar.gz"

echo ">>> Iniciando copia de seguridad Fedora ..."

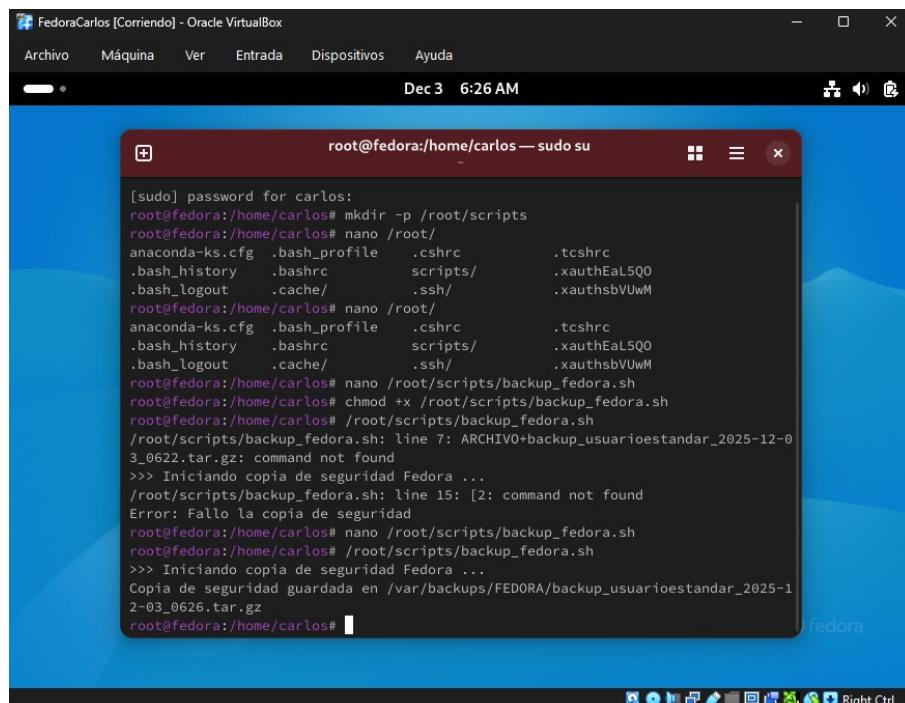
mkdir -p $DESTINO

tar -czf "$DESTINO/$ARCHIVO" "$ORIGEN" 2>/dev/null

if [ $? -eq 0]; then
    echo "Copia de seguridad guardada en $DESTINO/$ARCHIVO"
else
    echo "Error: Fallo la copia de seguridad"
fi
```

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.



```
[sudo] password for carlos:
root@fedora:/home/carlos# mkdir -p /root/scripts
root@fedora:/home/carlos# nano /root/
anaconda-ks.cfg .bash_profile .cshrc .tcshrc
.bash_history .bashrc scripts/ .xauthEaL5Q0
.bash_logout .cache/ .ssh/ .xauthsbVUwM
root@fedora:/home/carlos# nano /root/
anaconda-ks.cfg .bash_profile .cshrc .tcshrc
.bash_history .bashrc scripts/ .xauthEaL5Q0
.bash_logout .cache/ .ssh/ .xauthsbVUwM
root@fedora:/home/carlos# nano /root/scripts/backup_fedora.sh
root@fedora:/home/carlos# chmod +x /root/scripts/backup_fedora.sh
root@fedora:/home/carlos# /root/scripts/backup_fedora.sh
/root/scripts/backup_fedora.sh: line 7: ARCHIVO+backup_usuarioestandar_2025-12-0
3_0622.tar.gz: command not found
>>> Iniciando copia de seguridad Fedora ...
/root/scripts/backup_fedora.sh: line 15: [: command not found
Error: Fallo la copia de seguridad
root@fedora:/home/carlos# nano /root/scripts/backup_fedora.sh
root@fedora:/home/carlos# /root/scripts/backup_fedora.sh
>>> Iniciando copia de seguridad Fedora ...
Copia de seguridad guardada en /var/backups/FEDORA/backup_usuarioestandar_2025-1
2-03_0626.tar.gz
root@fedora:/home/carlos#
```

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:

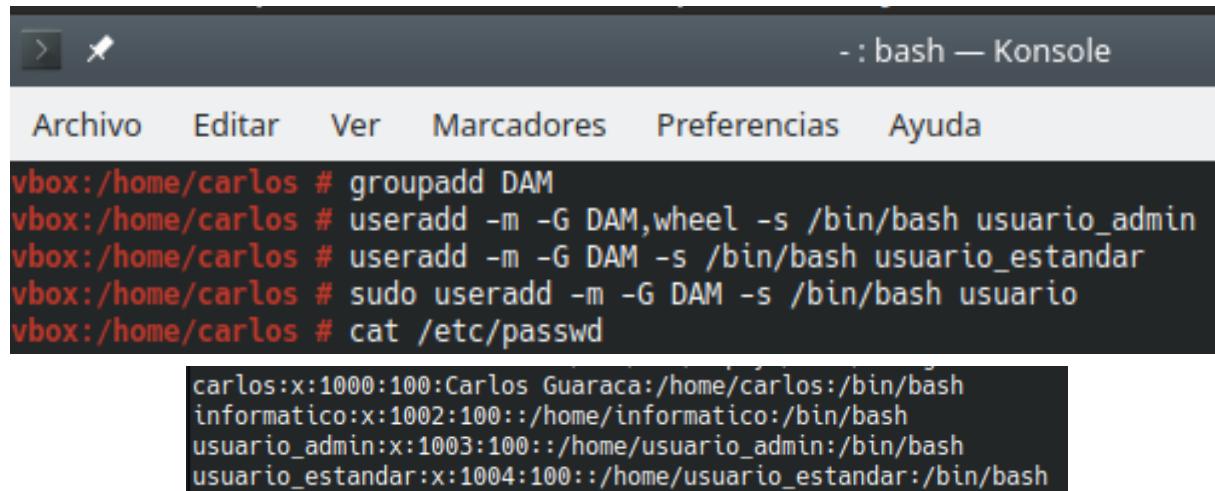
```
sudo chmod +x /root/scripts/backup_fedora.sh
```

4. Ejecutamos el script:

```
/root/scripts/backup_fedora.sh
```

OpenSuse

Paso 1. Configuración de Usuarios y Grupos Locales



```
vbox:/home/carlos # groupadd DAM
vbox:/home/carlos # useradd -m -G DAM,wheel -s /bin/bash usuario_admin
vbox:/home/carlos # useradd -m -G DAM -s /bin/bash usuario_estandar
vbox:/home/carlos # sudo useradd -m -G DAM -s /bin/bash usuario
vbox:/home/carlos # cat /etc/passwd
carlos:x:1000:100:Carlos Guaraca:/home/carlos:/bin/bash
informatico:x:1002:100::/home/informatico:/bin/bash
usuario_admin:x:1003:100::/home/usuario_admin:/bin/bash
usuario_estandar:x:1004:100::/home/usuario_estandar:/bin/bash
```

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 /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 era compatible con faillock del todo (pam-config puesto que no reconoce el parámetro faillock).

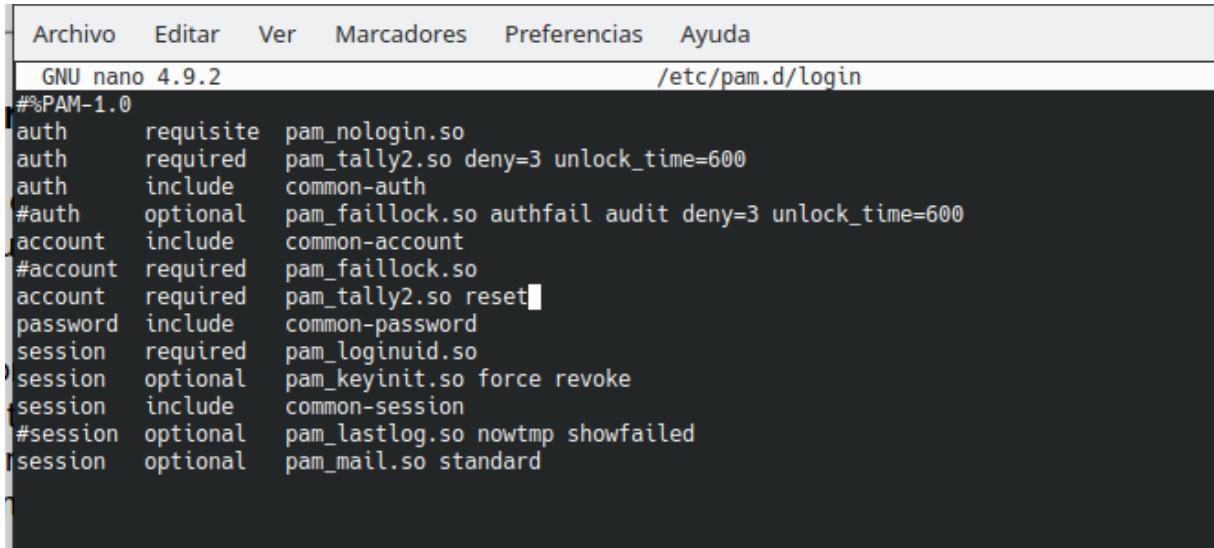
```
[root] password for root:  
vbox:/home/carlos # pam-config --add --faillock --faillock-deny=3 --faillock-unlock-time=600  
pam-config: invalid option -- --faillock  
Try 'pam-config --help' or 'pam-config --usage' for more information.  
vbox:/home/carlos #
```

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

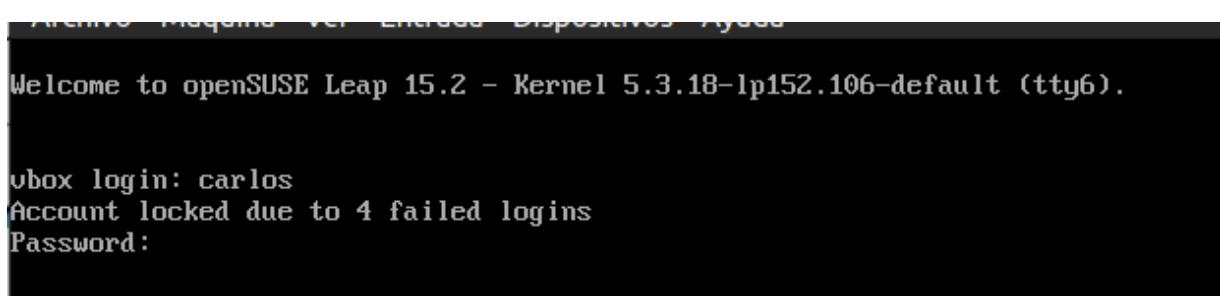
```
#%PAM-1.0  
auth    required  pam_faillock.so preauth audir silent deny=3 unlock_time=600  
auth    requisite pam_nologin.so  
auth    include  common-auth  
account  include  common-account  
account  required pam_faillock.so  
password include  common-password  
session  required pam_loginuid.so  
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 autenticación, 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).

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.



```
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
```

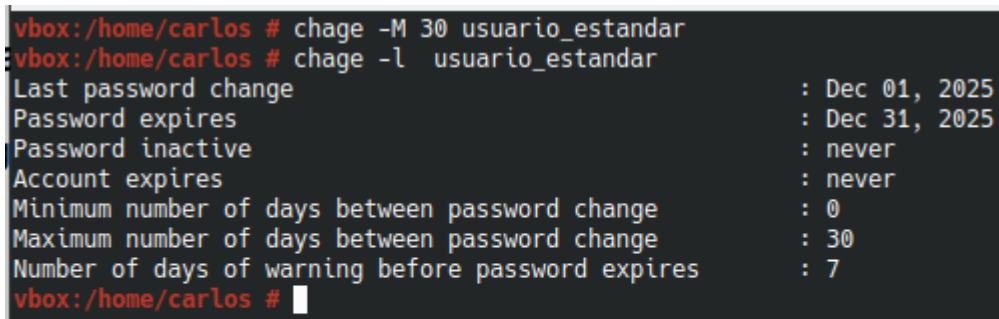


```
Welcome to openSUSE Leap 15.2 - Kernel 5.3.18-1p152.106-default (tty6).

vbox login: carlos
Account locked due to 4 failed logins
Password:
```

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

Paso 3. Seguridad de Contraseñas



```
vbox:/home/carlos # chage -M 30 usuario_estandar
vbox:/home/carlos # chage -l  usuario_estandar
Last password change : Dec 01, 2025
Password expires       : Dec 31, 2025
Password inactive      : never
Account expires        : never
Minimum number of days between password change : 0
Maximum number of days between password change : 30
Number of days of warning before password expires: 7
vbox:/home/carlos #
```

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:

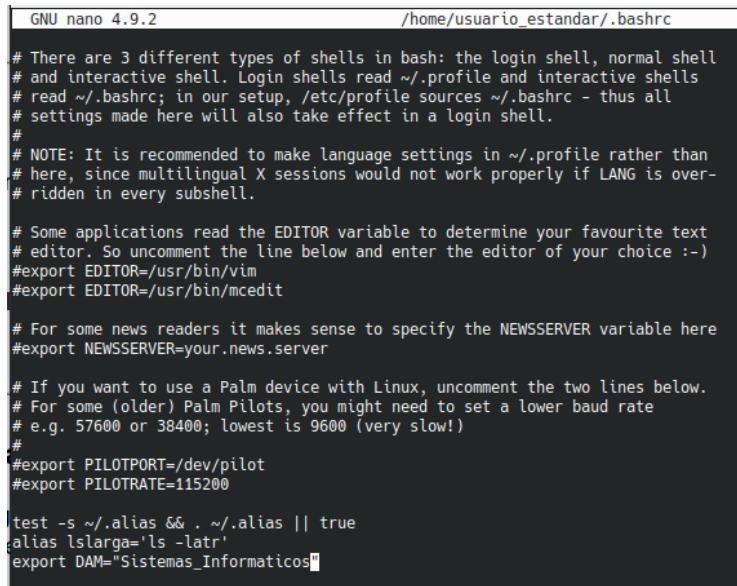
```
sudo chage -l "usuario_estandar"
```

Paso 4. Gestión del Entorno de Trabajo del Usuario

Crearemos un alias “**Islarga**” (como todas las 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 Islarga='ls -latr'
```

```
export DAM="Sistemas_Informaticos"
```



```
GNU nano 4.9.2 /home/usuario_estandar/.bashrc

# There are 3 different types of shells in bash: the login shell, normal shell
# and interactive shell. Login shells read ~/.profile and interactive shells
# read ~/.bashrc; in our setup, /etc/profile sources ~/.bashrc - thus all
# settings made here will also take effect in a login shell.
#
# NOTE: It is recommended to make language settings in ~/.profile rather than
# here, since multilingual X sessions would not work properly if LANG is over-
# ridden in every subshell.

# Some applications read the EDITOR variable to determine your favourite text
# editor. So uncomment the line below and enter the editor of your choice :-)
#export EDITOR=/usr/bin/vim
#export EDITOR=/usr/bin/mcedit

# For some news readers it makes sense to specify the NEWSERVER variable here
#export NEWSERVER=your.news.server

# If you want to use a Palm device with Linux, uncomment the two lines below.
# For some (older) Palm Pilots, you might need to set a lower baud rate
# e.g. 57600 or 38400; lowest is 9600 (very slow!)
#
#export PILOTPORT=/dev/pilot
#export PILOTRATE=115200

test -s ~/.alias && . ~/.alias || true
alias Islarga='ls -latr'
export DAM="Sistemas_Informaticos"
```

Y ya por último, lo comprobamos:

```

vbox:/home/carlos # nano /home/usuario_estandar/.bashrc
vbox:/home/carlos # source /home/usuario_estandar/.bashrc
vbox:/home/carlos # ls larga /home/usuario_estandar
ls: cannot access 'larga': No such file or directory
/home/usuario_estandar:
.bash_history .config .inputrc .muttrc .urlview .xinitrc.template
.bashrc .emacs .i18n .local .profile .xim.template bin
vbox:/home/carlos # lslarga /home/usuario_estandar
total 40
-rw----- 1 usuario_estandar users 0 May 18 1996 .bash_history
-rw-r--r-- 1 usuario_estandar users 861 Apr 9 2018 .inputrc
-rw-r--r-- 1 usuario_estandar users 1637 Apr 9 2018 .emacs
-rw-r--r-- 1 usuario_estandar users 1951 Sep 20 2019 .xim.template
-rw-r--r-- 1 usuario_estandar users 73 Sep 20 2019 .i18n
drwxr-xr-x 1 usuario_estandar users 0 Mar 7 2020 bin
drwx----- 1 usuario_estandar users 0 Mar 7 2020 .local
drwxr-xr-x 1 usuario_estandar users 0 Mar 7 2020 .fonts
drwx----- 1 usuario_estandar users 0 Mar 7 2020 .config
-rw-r--r-- 1 usuario_estandar users 311 May 16 2020 .urlview
-rwxr-xr-x 1 usuario_estandar users 1112 May 16 2020 .xinitrc.template
-rw-r--r-- 1 usuario_estandar users 6043 May 16 2020 .muttrc
-rw-r--r-- 1 usuario_estandar users 1028 Jun 8 2020 .profile
drwxr-xr-x 1 root root 84 Mar 23 2021 ..
drwxr-xr-x 1 usuario_estandar users 228 Dec 1 08:50 .
-rw-r--r-- 1 usuario_estandar users 1237 Dec 1 10:04 .bashrc
vbox:/home/carlos # echo $DAM
Sistemas_Informaticos
vbox:/home/carlos #

```

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

1. Accedemos 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:

```
lslargu /home/usuario_estandar
```

```
echo $DAM
```

Paso 5. Acceso a Recursos y Permisos Locales

```

Sistemas_Informaticos
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 únicamente 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 mismo 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
```

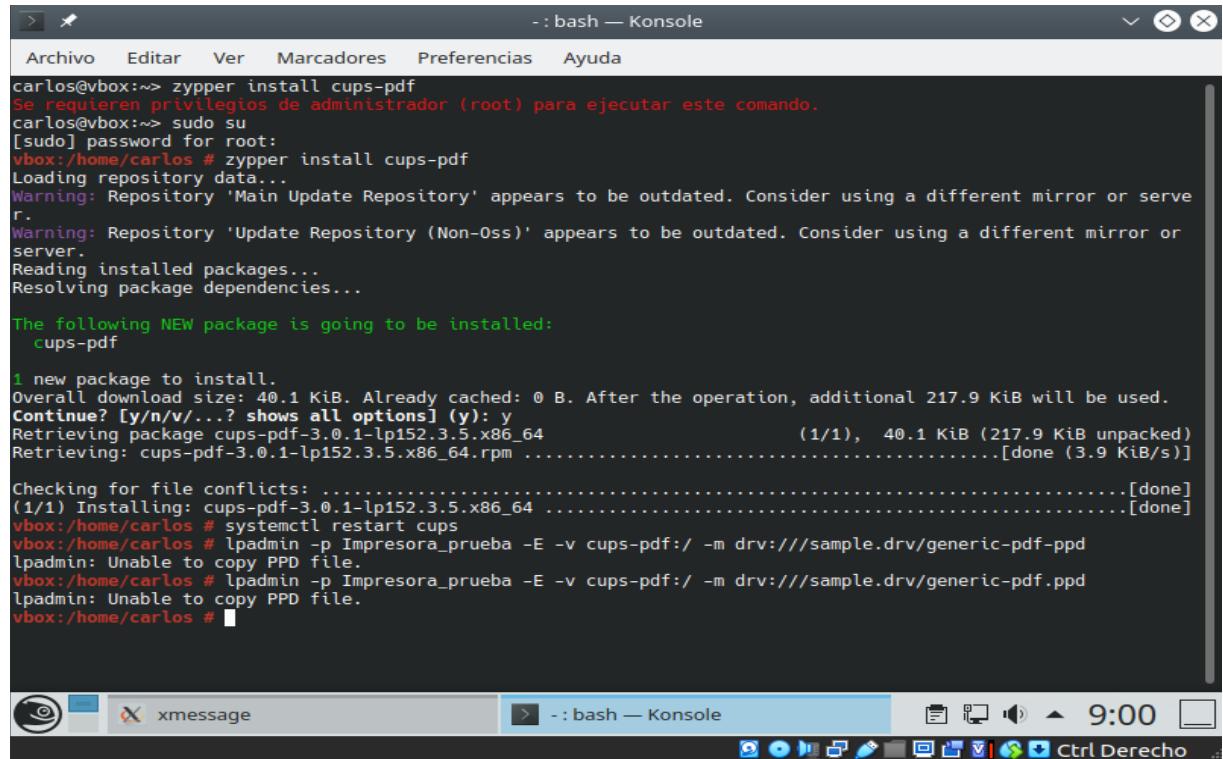
7. Realizamos la comprobación necesaria:

```
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).



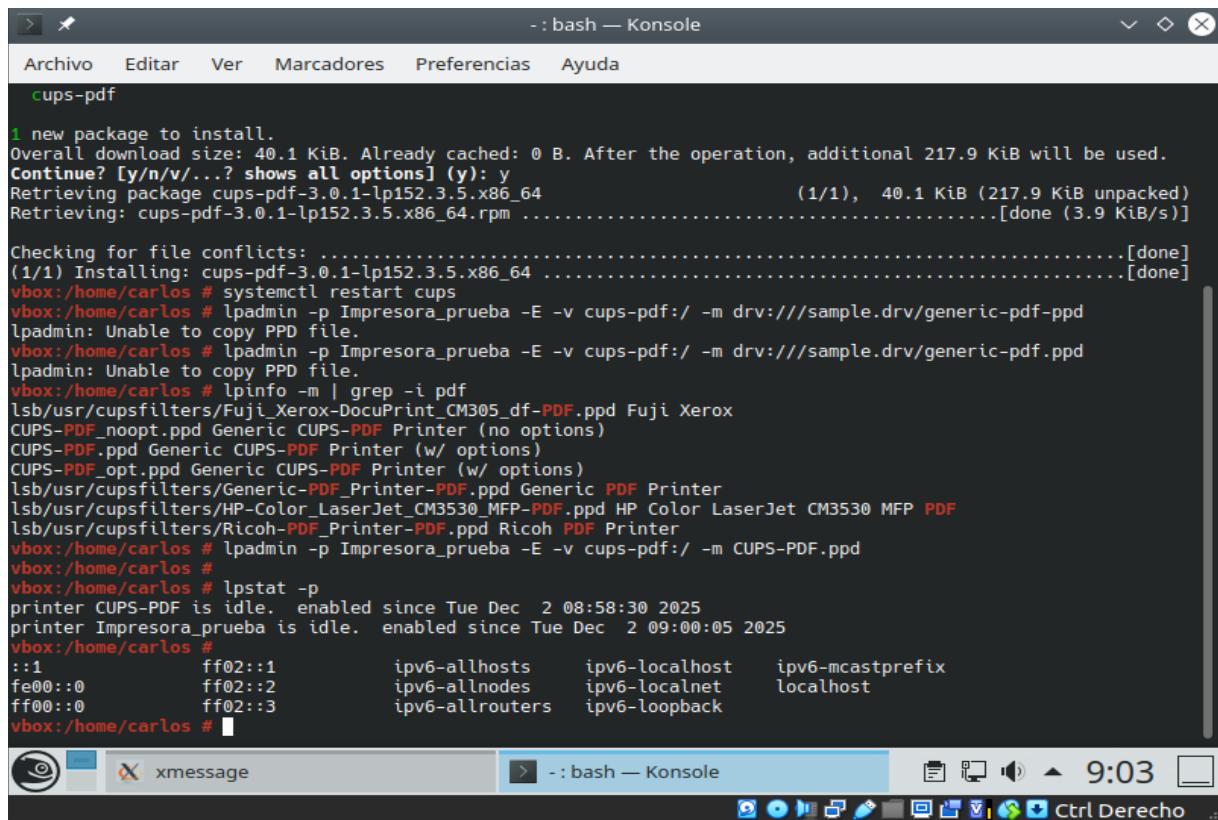
The screenshot shows a terminal window titled '- : bash — Konsole'. The window contains the following command and its output:

```
carlos@vbox:~> zypper install cups-pdf
Se requieren privilegios de administrador (root) para ejecutar este comando.
carlos@vbox:~> sudo su
[sudo] password for root:
vbox:/home/carlos # zypper install cups-pdf
Loading repository data...
Warning: Repository 'Main Update Repository' appears to be outdated. Consider using a different mirror or server.
Warning: Repository 'Update Repository (Non-Oss)' appears to be outdated. Consider using a different mirror or server.
Reading installed packages...
Resolving package dependencies...

The following NEW package is going to be installed:
  cups-pdf

1 new package to install.
Overall download size: 40.1 KiB. Already cached: 0 B. After the operation, additional 217.9 KiB will be used.
Continue? [y/n/v...? shows all options] (y): y
Retrieving package cups-pdf-3.0.1-lp152.3.5.x86_64                               (1/1), 40.1 KiB (217.9 KiB unpacked)
Retrieving: cups-pdf-3.0.1-lp152.3.5.x86_64.rpm .....[done (3.9 KiB/s)]

Checking for file conflicts: .....[done]
(1/1) Installing: cups-pdf-3.0.1-lp152.3.5.x86_64 .....[done]
vbox:/home/carlos # systemctl restart cups
vbox:/home/carlos # lpadmin -p Impresora_prueba -E -v cups-pdf:/ -m drv:///sample.drv/generic-pdf.ppd
lpadmin: Unable to copy PPD file.
vbox:/home/carlos # lpadmin -p Impresora_prueba -E -v cups-pdf:/ -m drv:///sample.drv/generic-pdf.ppd
lpadmin: Unable to copy PPD file.
vbox:/home/carlos #
```



```
cups-pdf
1 new package to install.
Overall download size: 40.1 KiB. Already cached: 0 B. After the operation, additional 217.9 KiB will be used.
Continue? [y/n/v/...? shows all options] (y): y
Retrieving package cups-pdf-3.0.1-1p152.3.5.x86_64 (1/1), 40.1 KiB (217.9 KiB unpacked)
Retrieving: cups-pdf-3.0.1-1p152.3.5.x86_64.rpm .....[done (3.9 KiB/s)]
Checking for file conflicts: .....[done]
(1/1) Installing: cups-pdf-3.0.1-1p152.3.5.x86_64 .....[done]
vbox:/home/carlos # systemctl restart cups
vbox:/home/carlos # lpadmin -p Impresora_prueba -E -v cups-pdf:/ -m drv:///sample.drv/generic-pdf.ppd
lpadmin: Unable to copy PPD file.
vbox:/home/carlos # lpadmin -p Impresora_prueba -E -v cups-pdf:/ -m drv:///sample.drv/generic-pdf.ppd
lpadmin: Unable to copy PPD file.
vbox:/home/carlos # lpinfo -m | grep -i pdf
lsb/usr/cupsfilters/Fuji_Xerox-DocuPrint_CM305_df-PDF.ppd Fuji Xerox
CUPS-PDF_noopt.ppd Generic CUPS-PDF Printer (no options)
CUPS-PDF.ppd Generic CUPS-PDF Printer (w/ options)
CUPS-PDF_opt.ppd Generic CUPS-PDF Printer (w/ options)
lsb/usr/cupsfilters/Generic-PDF_Printer-PDF.ppd Generic PDF Printer
lsb/usr/cupsfilters/HP-Color_LaserJet_CM3530_MFP-PDF.ppd HP Color LaserJet CM3530 MFP PDF
lsb/usr/cupsfilters/Ricoh-PDF_Printer-PDF.ppd Ricoh PDF Printer
vbox:/home/carlos # lpadmin -p Impresora_prueba -E -v cups-pdf:/ -m CUPS-PDF.ppd
vbox:/home/carlos #
vbox:/home/carlos # lpstat -p
printer CUPS-PDF is idle. enabled since Tue Dec 2 08:58:30 2025
printer Impresora_prueba is idle. enabled since Tue Dec 2 09:00:05 2025
vbox:/home/carlos #
vbox:/home/carlos # ::1 ff02::1 ipv6-allhosts ipv6-localhost ipv6-mcastprefix
fe00::0 ff02::2 ipv6-allnodes ipv6-localnet localhost
ff00::0 ff02::3 ipv6-allrouters ipv6-loopback
vbox:/home/carlos #
```

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
```

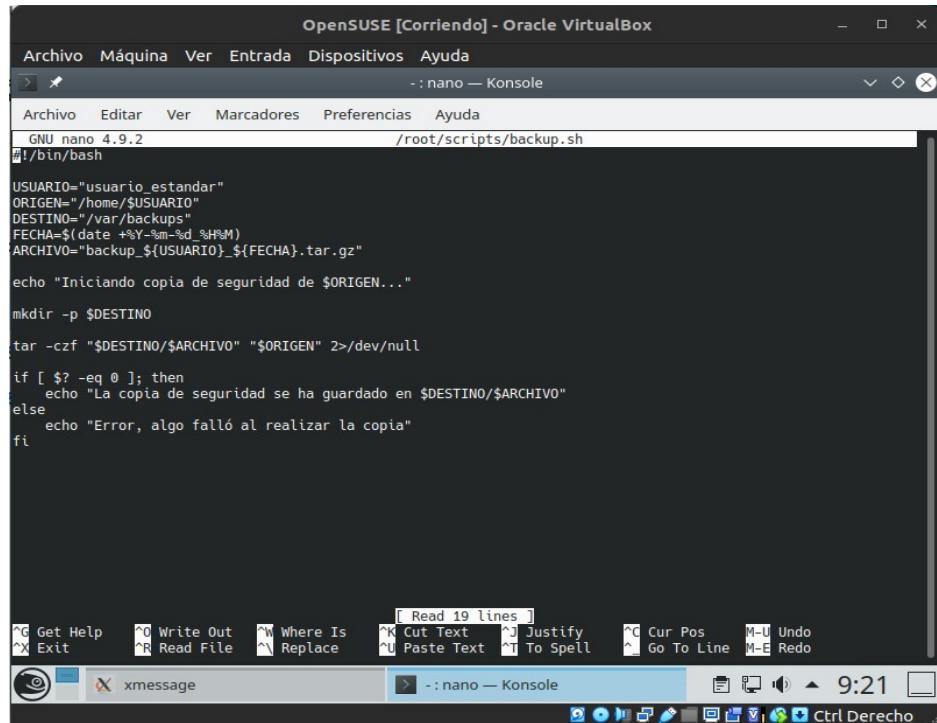
6. Verificamos su estado:

```
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 llamará **backup.sh** (con el nombre que quieras pero que termine por **.sh**).



```
GNU nano 4.9.2          /root/scripts/backup.sh
#!/bin/bash

USUARIO="usuario_estandar"
ORIGEN="/home/$USUARIO"
DESTINO="/var/backups"
FECHA=$(date +%Y-%m-%d_%H%M)
ARCHIVO="backup_${USUARIO}_${FECHA}.tar.gz"

echo "Iniciando copia de seguridad de $ORIGEN..."

mkdir -p $DESTINO
tar -czf "$DESTINO/$ARCHIVO" "$ORIGEN" 2>/dev/null

if [ $? -eq 0 ]; then
    echo "La copia de seguridad se ha guardado en $DESTINO/$ARCHIVO"
else
    echo "Error, algo falló al realizar la copia"
fi
```

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

```
vbox:/home/carlos # mkdir -p /root/scripts
vbox:/home/carlos # nano /root/scripts/backup.sh
vbox:/home/carlos # nano /root/scripts/backup.sh
vbox:/home/carlos # nano /root/scripts/backup.sh
vbox:/home/carlos # chmod +x /root/scripts/backup.sh
vbox:/home/carlos # /root/
.cache/.config/.gnupg/.local/bin/inst-sys/scripts/
vbox:/home/carlos # /root/scripts/backup.sh
Iniciando copia de seguridad de /home/usuario_estandar...
La copia de seguridad se ha guardado en /var/backups/backup_usuario_estandar_2025-12-02_0923.tar.gz
vbox:/home/carlos #
```

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
```