

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

```
MintCarlos [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@carlos-VirtualBox:/home/carlos#
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 []:
    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' ...
```

```
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 []:
    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#
```

```
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 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
```

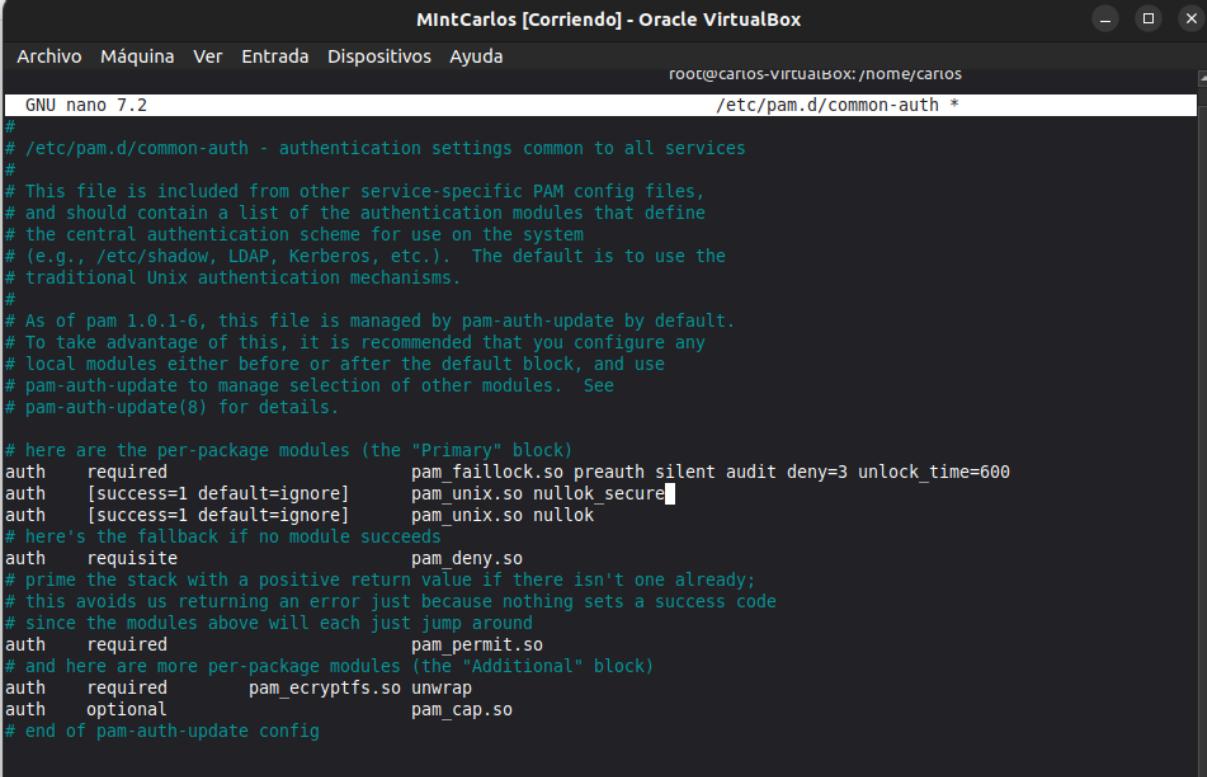
4. Verifica que pertenece a un grupo

```
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**:



```

GNU nano 7.2
root@carlos-virtualbox:/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

```

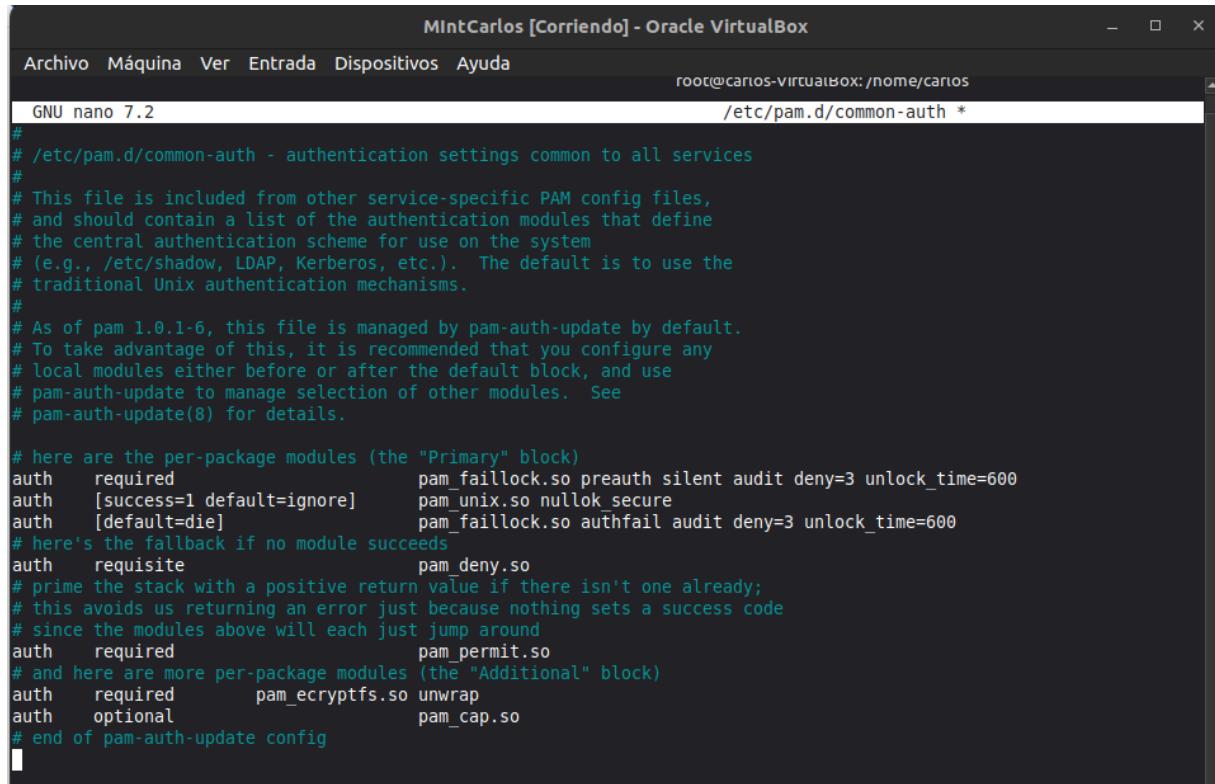
Debemos añadir las líneas de **pam_faillock.so** que bloquea y cuenta los intentos:

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

pam_unix.so nullok_secure

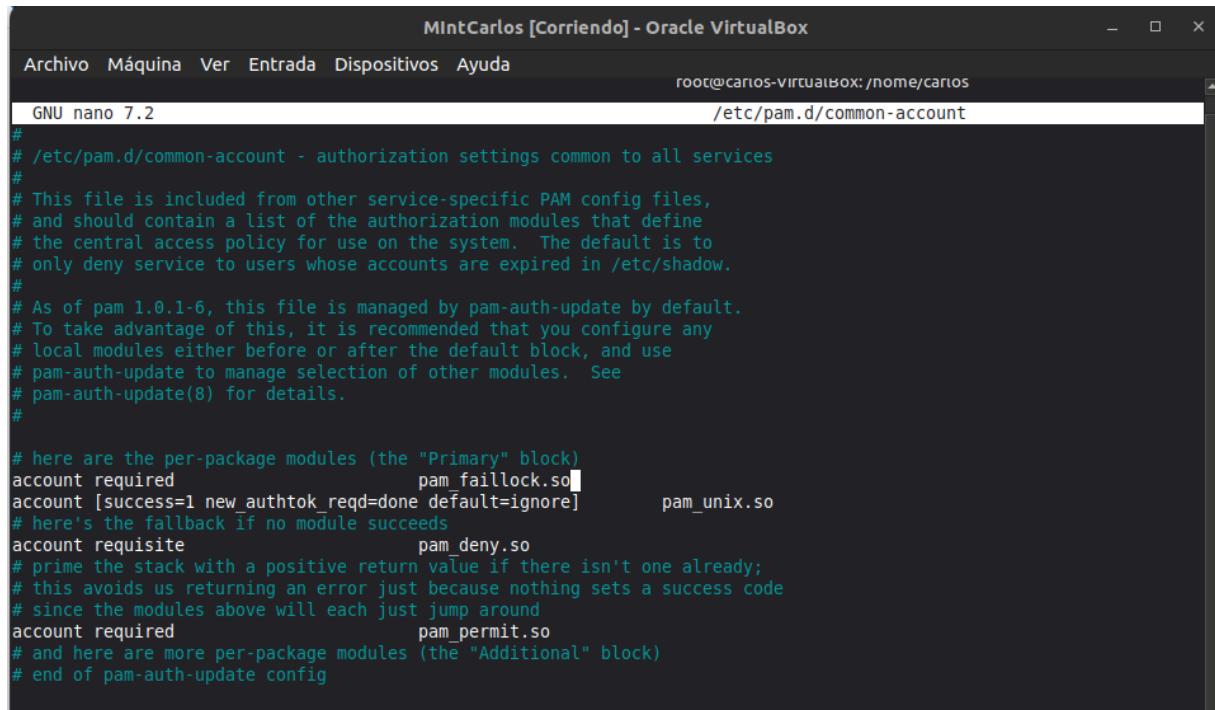
Estas líneas deben ponerse al principio de la sección **auth** (Sin embargo esto no era lo único que hacer, ya que ahora hay un conflicto con la línea **pam_unix.so nullok**). Para poder proseguir con la tarea y que funcione tuve que eliminar esa línea e insertar la línea **auth [default=die] pam_faillock.so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



```
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_ecryptfs.so unwrap  
auth    optional           pam_cap.so  
# end of pam-auth-update config
```

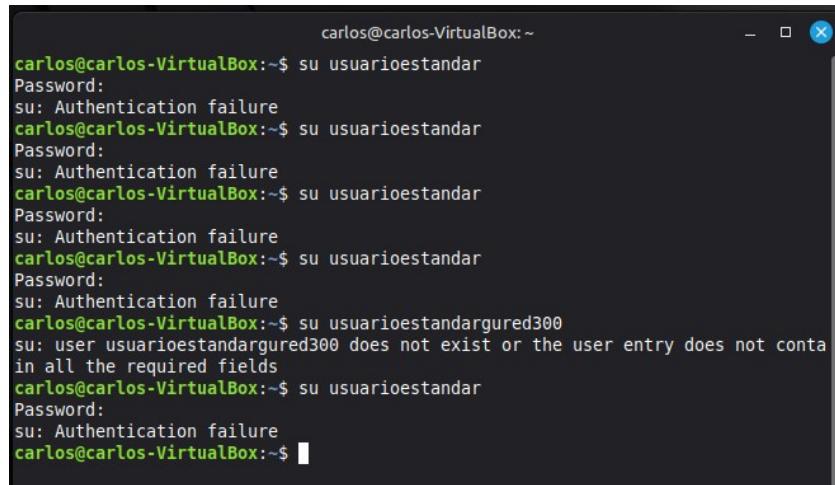


```

GNU nano 7.2
#
# /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_unix.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

```

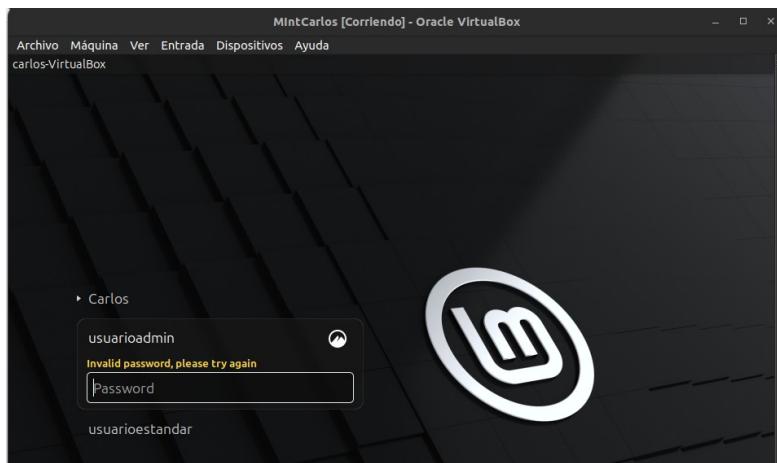
Por último, comprobamos que funcione (Como vemos, luego del 4º Intento, aunque ponga la contraseña correcta, no me deja iniciar):



```

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 "$(echo terminal || echo error)" "$(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"

^G Ayuda      ^O Guardar      ^W Buscar      ^X Cortar      ^T Ejecutar      ^C Ubicación
^X Salir      ^R Leer fich.   ^Y Reemplazar   ^U Pegar       ^J Justificar   ^/ Ir a línea

```

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:

lslarga

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 «usemod» 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** (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).

```
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-drived 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-drived failed to get PPD file - see error_log for details.
root@carlos-VirtualBox:/home/carlos# lpinfo -m | grep -i "cups-pdf.ppd"
lsb/usr/cups-filters/Fuji Xerox-DocuPrint CM305 df PDF.ppd Fuji Xerox DocuPrint CM305 df PDF
lsb/usr/cups-pdf/CUPS-PDF_noopt.ppd Generic CUPS-PDF Printer (no options)
lsb/usr/cups-pdf/CUPS-PDF_opt.ppd Generic CUPS-PDF Printer (w/ options)
lsb/usr/cups-filters/Generic-PDF_Printer-PDF.ppd Generic PDF Printer
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-C7521n PDF.ppd Gestetner C7521n PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-C7526dn PDF.ppd Gestetner C7526dn PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-C7531dn PDF.ppd Gestetner C7531dn PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-C7640n PDF.ppd Gestetner C7640n PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-C8140n PDF.ppd Gestetner C8140n PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-C8150n PDF.ppd Gestetner C8150n PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-CS555 PDF.ppd Gestetner CS555 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSC424 PDF.ppd Gestetner DSC424 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSC432 PDF.ppd Gestetner DSC432 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSC460 PDF.ppd Gestetner DSC460 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSc1020 PDF.ppd Gestetner DSc1020 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSc1025 PDF.ppd Gestetner DSc1025 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSc1030 PDF.ppd Gestetner DSc1030 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSc1045 PDF.ppd Gestetner DSc1045 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSc1060 PDF.ppd Gestetner DSc1060 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSc1120 PDF.ppd Gestetner DSc1120 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSc1220 PDF.ppd Gestetner DSc1220 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSc1220ex PDF.ppd Gestetner DSc1220ex PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSc1230 PDF.ppd Gestetner DSc1230 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSc1230ex PDF.ppd Gestetner DSc1230ex PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSc1245 PDF.ppd Gestetner DSc1245 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSc1245ex PDF.ppd Gestetner DSc1245ex PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSc1260 PDF.ppd Gestetner DSc1260 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSc1260ex PDF.ppd Gestetner DSc1260ex PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSm_2625 PDF.ppd Gestetner DSm 2625 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSm_2630 PDF.ppd Gestetner DSm 2630 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSm_2635 PDF.ppd Gestetner DSm 2635 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSm_2640 PDF.ppd Gestetner DSm 2640 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSm_2650 PDF.ppd Gestetner DSm 2650 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSm_2660 PDF.ppd Gestetner DSm 2660 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSm725 PDF.ppd Gestetner DSm725 PDF
openprinting-pdfs:0/0/0/openprinting/Gestetner/PDF/Gestetner-DSm730 PDF.ppd 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 para de **lpadmin** con este 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#
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://
                  └─3305 /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 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
```

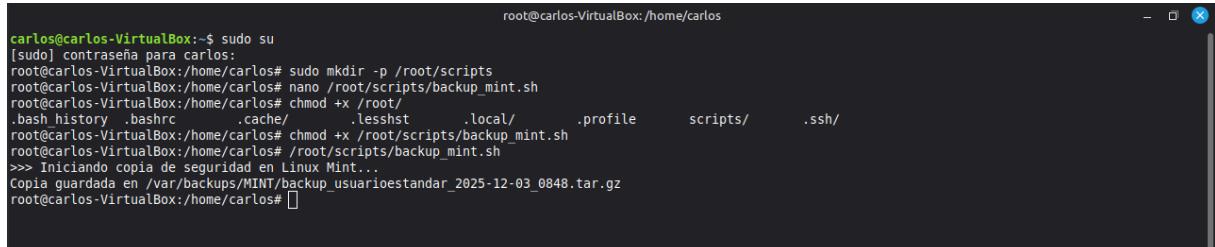
4. Añadimos la impresora PDF a lpadmin (forma correcta):

```
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



```
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
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 se que algo fallé, 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
```