

## UNIDAD 4: UBUNTU Y SUS COMANDOS.

### I. Fundamentos de Linux.

#### I.1. Directorios y sistemas de archivos.

En Linux y Unix **todo es un fichero**. Los directorios son ficheros, los ficheros son ficheros, y los dispositivos son ficheros. A veces a los dispositivos se les llama nodos, pero siguen siendo ficheros.

Los sistemas de ficheros de Linux y Unix se organizan en una estructura jerárquica, de tipo árbol. El nivel más alto del sistema de ficheros es **/** o **directorio raíz**. Todos los demás ficheros y directorios están bajo el directorio raíz. Por ejemplo, `/home/lorena/apuntes.docx` muestra la ruta completa al fichero *apuntes.docx* que está en el directorio *lorena*, que a su vez está bajo el directorio *home*, que por su parte está bajo el directorio raíz (**/**).

Por debajo del directorio raíz (**/**) hay un importante grupo de directorios común a la mayoría de las distribuciones de GNU/Linux. A continuación hay una lista de los directorios que aparecen normalmente bajo el directorio raíz (**/**):

- **/bin** - aplicaciones *binarias* importantes
- **/boot** - Ficheros de configuración del arranque, núcleos y otros ficheros necesarios para el arranque (*boot*) del equipo.
- **/dev** - los ficheros de *dispositivo*
- **/etc** - ficheros de configuración, scripts de arranque, *etc.*
- **/home** - directorios personales (*home*) para los diferentes usuarios.
- **/initrd** - usado cuando se crea un proceso de arranque *initrd* personalizado.
- **/lib** - librerías del sistema (*libraries*)
- **/lost+found** - proporciona un sistema de "perdido+encontrado" (*lost+found*) para los ficheros que existen debajo del directorio raíz (**/**)
- **/media** - particiones montadas (cargadas) automáticamente en el disco duro y medios (media) extraíbles como CDs, cámaras digitales, etc.
- **/mnt** - sistemas de archivos *montados* manualmente en el disco duro.
- **/opt** - proporciona una ubicación donde instalar aplicaciones opcionales (de terceros)
- **/proc** - directorio dinámico especial que mantiene información sobre el estado del sistema, incluyendo los *procesos* actualmente en ejecución

- **/root** - directorio personal del usuario *root* (superusuario); también llamado "barra-root".
- **/sbin** - *binarios* importantes del sistema
- **/srv** - puede contener archivos que se *sirven* a otros sistemas
- **/sys** - archivos del sistema (*system*)
- **/tmp** - *temporary files*
- **/usr** - aplicaciones y archivos a los que puede acceder la mayoría de los *usuarios*
- **/var** - archivos *variables* como archivos de registros y bases de datos

## 1.2. Permisos.

Todos los archivos de un sistema Linux tienen permisos que permiten o impiden a otros verlos, modificarlos o ejecutarlos. El superusuario "root" tiene acceso a cualquier archivo del sistema. Cada archivo tiene restricciones de acceso, restricciones de usuario y está asociado a un propietario y un grupo.

Cada archivo está asegurado por las tres capas de permisos siguientes, en orden de importancia:

- **usuario:** se aplica al usuario que es el propietario del archivo
- **grupo:** se aplica al grupo asociado al archivo
- **otros:** se aplica a todos los demás usuarios

Los permisos reales están dentro de cada uno de los tres conjuntos de permisos. Los permisos, junto con los distintos modos en que afectan a los archivos y a los directorios, se describen a continuación:

- **lectura:** los *archivos* pueden ser visualizados/abiertos; el contenido del *directorio* se puede visualizar.
- **escritura:** los *archivos* se pueden modificar o eliminar; el contenido del *directorio* se puede modificar.
- **ejecución:** los *archivos ejecutables* se pueden arrancar como un programa; se puede entrar en los *directorios*.

Para ver y editar los permisos de archivos y directorios, abra el terminal (**Ctrl+Alt+T**) y ejecute el comando **ls -l** como se muestra a continuación:

```
lorena@lorena-VirtualBox:~$ ls -l
total 180
drwxrwxr-x 2 lorena lorena 4096 ene 24 20:38 '?'
-rw-rw-r-- 1 lorena lorena 174 may 25 2022 caracterizacion_plantas.txt
-rw-rw-r-- 1 lorena lorena 102 may 25 2022 correcto.txt
drwxr-xr-x 2 lorena lorena 4096 may 16 2022 Descargas
drwxr-xr-x 2 lorena lorena 4096 jun 1 2022 Documentos
-rw-rw-r-- 1 lorena lorena 21 may 19 2022 ejemploded10.txt
-rw-rw-r-- 1 lorena lorena 137 may 19 2022 ejemploded11.txt
-rw-rw-r-- 1 lorena lorena 52 may 19 2022 ejemploded12.txt
-rw-rw-r-- 1 lorena lorena 58 may 20 2022 ejemploded13.txt
-rw-rw-r-- 1 lorena lorena 36 may 20 2022 ejemploded14.txt
-rw-rw-r-- 1 lorena lorena 52 may 20 2022 ejemploded16.txt
-rw-rw-r-- 1 lorena lorena 213 may 20 2022 ejemploded17.txt
-rw-rw-r-- 1 lorena lorena 173 may 18 2022 ejemploded1.txt
-rw-rw-r-- 1 lorena lorena 355 may 18 2022 ejemploded2.txt
-rw-rw-r-- 1 lorena lorena 184 may 19 2022 ejemploded3.txt
-rw-rw-r-- 1 lorena lorena 112 may 18 2022 ejemploded4.txt
-rw-rw-r-- 1 lorena lorena 211 may 19 2022 ejemploded5.txt
-rw-rw-r-- 1 lorena lorena 20 may 19 2022 ejemploded6.txt
-rw-rw-r-- 1 lorena lorena 152 may 19 2022 ejemploded7.txt
-rw-rw-r-- 1 lorena lorena 6 may 19 2022 ejemploded8.txt
-rw-rw-r-- 1 lorena lorena 104 may 19 2022 ejemploded9.txt
-rw-rw-r-- 1 lorena lorena 173 may 18 2022 ejemplodedres.txt
drwxr-xr-x 3 lorena lorena 4096 ene 24 19:56 Escritorio
-rw-rw-r-- 1 lorena lorena 1197 may 25 2022 fichero.txt
-rw-r--r-- 1 lorena root 46 may 18 2022 grep.txt
-rw-rw-r-- 1 lorena lorena 10979 may 25 2022 historia.txt
drwxr-xr-x 2 lorena lorena 4096 may 16 2022 Imágenes
drwxr-xr-x 2 lorena lorena 4096 may 16 2022 Música
-rw-rw-r-- 1 lorena lorena 30 may 18 2022 nombres.txt
-rw-rw-r-- 1 lorena lorena 67 may 18 2022 números.txt
-rwxrwxr-x 1 lorena lorena 1426 ene 24 20:41 operacionesconusuario.sh
-rw-rw-r-- 1 lorena lorena 382 may 25 2022 origen.txt
drwxr-xr-x 2 lorena lorena 4096 may 16 2022 Plantillas
drwxr-xr-x 2 lorena lorena 4096 may 16 2022 Público
drwx----- 3 lorena lorena 4096 jun 1 2022 snap
-rw-rw-r-- 1 lorena lorena 44 may 20 2022 tempo2.txt
-rw-rw-r-- 1 lorena lorena 15 may 20 2022 tempo3.txt
-rw-rw-r-- 1 lorena lorena 33 may 20 2022 tempo4.txt
-rw-rw-r-- 1 lorena lorena 53 may 20 2022 tempo5.txt
-rw-rw-r-- 1 lorena lorena 54 may 20 2022 tempo.txt
-rw-rw-r-- 1 lorena lorena 30 may 19 2022 test2.txt
-rw-rw-r-- 1 lorena lorena 163 may 19 2022 textofinal
drwxr-xr-x 2 lorena lorena 4096 may 16 2022 Videos
```

### 1.3. Usuario “root” y sudo.

El usuario **root** en GNU/Linux es el usuario que tiene acceso administrativo al sistema. Los usuarios normales no tienen este acceso por razones de seguridad. Es fácil comprender el peligro que supone trabajar de forma habitual con un usuario tan poderoso. Lo ideal sería

recurrir a él sólo en los momentos en los que resulte imprescindible, pero siempre han existido usuarios que se olvidaban de volver a su cuenta habitual después de realizar tareas administrativas.

Por ese motivo, y con el fin de salvarnos de nosotros mismos, Ubuntu realiza un tratamiento algo particular de la cuenta root. Sencillamente, la mantiene deshabilitada de forma predeterminada, mientras convierte a nuestro usuario habitual en un miembro del grupo administradores. De esta forma, podremos realizar tareas particulares de administración con la cuenta que usamos para trabajar a diario, pero, cada vez que concluya una tarea administrativa, la cuenta volverá a comportarse, automáticamente, como una cuenta normal.

¿Cómo se consigue que una cuenta del grupo administradores (**sudo**) haga una tarea administrativa? La respuesta es muy sencilla: Si nos encontramos en la interfaz gráfica, al tratar de ejecutar una tarea administrativa, aparecerá una ventana que nos solicita la contraseña (la propia de la cuenta de usuario). Y si nos encontramos en la línea de comandos, sólo tenemos que **anteponer la palabra sudo** a la orden que pretendamos ejecutar con privilegios administrativos.

Cuando ejecuta una aplicación que requiere privilegios de administrador, sudo le pedirá que escriba su contraseña de usuario normal. Esto asegura que aplicaciones incontroladas no puedan dañar su sistema, y sirve como recordatorio de que está a punto de realizar acciones administrativas que requieren que tenga cuidado.

Para usar sudo en la línea de comandos, simplemente escriba "sudo" antes del comando que desea ejecutar. *Sudo le pedirá su contraseña.*

Sudo recordará su contraseña durante un periodo de tiempo (predeterminado a 15 minutos). Esta característica se diseñó para permitir a los usuarios realizar múltiples tareas administrativas sin tener que escribir su contraseña cada vez.

## **Habilitar el usuario root para acceder desde la terminal**

Para **habilitar el usuario root** lo único que hay que hacer es asignarle una contraseña. Para ello utilizamos el comando sudo que hemos visto más arriba.

```
lorena@lorena-VirtualBox:~$ sudo passwd root
[sudo] contraseña para lorena:
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
```

## Habilitar el usuario root para acceder desde la interfaz gráfica

Después de poner en práctica el apartado anterior, podemos pensar que bastará con reiniciar el equipo para autenticarnos, con la cuenta **root**, en la interfaz gráfica de *Ubuntu 20.04 LTS*. Sin embargo, si lo intentas recibirás un error.

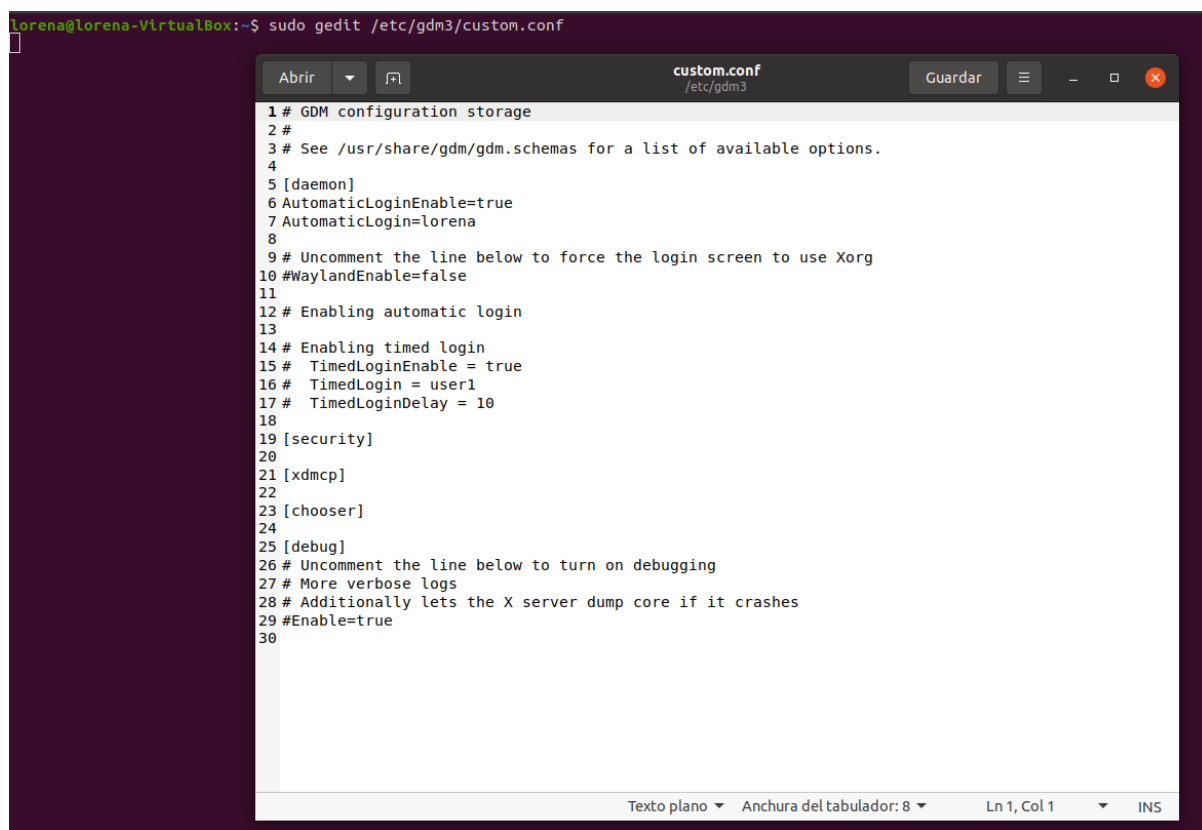
Esto es por el modo en el que funciona *GDM*, que es el gestor de sesiones que utiliza *Ubuntu* de forma predeterminada.

GDM viene preconfigurado para no permitir el inicio de sesión con la cuenta root, haciendo lo siguiente.

Antes de comenzar instalaremos el editor de texto gedit:

```
lorena@lorena-VirtualBox:~$ sudo apt install gedit
```

Empezamos editando su archivo de configuración con una orden como esta:



```
lorena@lorena-VirtualBox:~$ sudo gedit /etc/gdm3/custom.conf

custom.conf
/et/gdm3

1 # GDM configuration storage
2 #
3 # See /usr/share/gdm/gdm.schemas for a list of available options.
4
5 [daemon]
6 AutomaticLoginEnable=true
7 AutomaticLogin=lorena
8
9 # Uncomment the line below to force the login screen to use Xorg
10 #WaylandEnable=false
11
12 # Enabling automatic login
13
14 # Enabling timed login
15 # TimedLoginEnable = true
16 # TimedLogin = user1
17 # TimedLoginDelay = 10
18
19 [security]
20
21 [xdmcp]
22
23 [chooser]
24
25 [debug]
26 # Uncomment the line below to turn on debugging
27 # More verbose logs
28 # Additionally lets the X server dump core if it crashes
29 #Enable=true
30

Texto plano Anchura del tabulador: 8 Ln 1, Col 1 INS
```

Esto nos abrirá una nueva ventana con el procesador de textos.

En él, debemos localizar la categoría *[security]* que, como ves en la imagen anterior, aún no tiene contenido. Bajo el epígrafe de la categoría, debemos incluir el siguiente texto:

```
lorenalorena-VirtualBox:~$ sudo gedit /etc/gdm3/custom.conf

custom.conf
/etc/gdm3

1 # GDM configuration storage
2 #
3 # See /usr/share/gdm/gdm.schemas for a list of available options.
4
5 [daemon]
6 AutomaticLoginEnable=true
7 AutomaticLogin=lorena
8
9 # Uncomment the line below to force the login screen to use Xorg
10 #WaylandEnable=false
11
12 # Enabling automatic login
13
14 # Enabling timed login
15 # TimedLoginEnable = true
16 # TimedLogin = user1
17 # TimedLoginDelay = 10
18
19 [security]
20
21 [xdmcp]
22
23 [chooser]
24
25 [debug]
26 # Uncomment the line below to turn on debugging
27 # More verbose logs
28 # Additionally lets the X server dump core if it crashes
29 #Enable=true
30
```

A continuación realizaremos otro pequeño cambio en el archivo *gdm-password*. Para editarlo, comenzaremos escribiendo algo como esto:

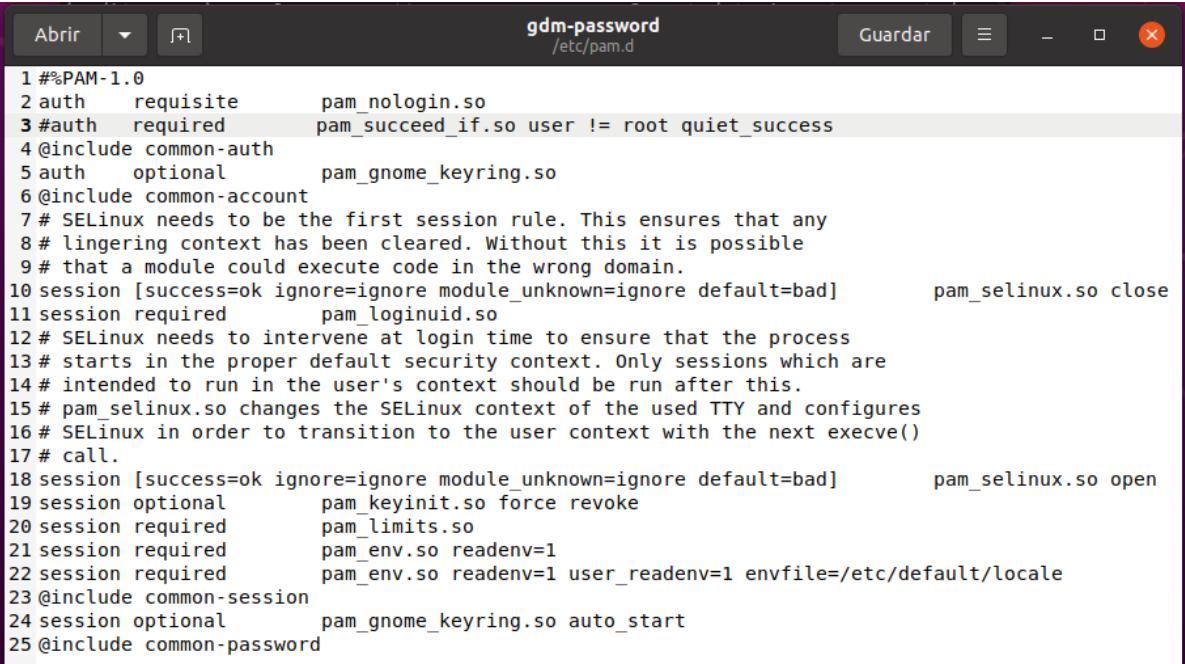
```
lorenalorena-VirtualBox:~$ sudo gedit /etc/pam.d/gdm-password
```

Cuando aparezca la ventana de gEdit con el contenido del archivo, localizamos una línea como esta: **auth required pam\_succeed\_if.so user != root quiet\_success**

```
gdm-password
/etc/pam.d

1 #%PAM-1.0
2 auth requisite pam_nologin.so
3 auth required pam_succeed_if.so user != root quiet_success
4 @include common-auth
5 auth optional pam_gnome_keyring.so
6 @include common-account
7 # SELinux needs to be the first session rule. This ensures that any
8 # lingering context has been cleared. Without this it is possible
9 # that a module could execute code in the wrong domain.
10 session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so close
11 session required pam_loginuid.so
12 # SELinux needs to intervene at login time to ensure that the process
13 # starts in the proper default security context. Only sessions which are
14 # intended to run in the user's context should be run after this.
15 # pam_selinux.so changes the SELinux context of the used TTY and configures
16 # SELinux in order to transition to the user context with the next execve()
17 # call.
18 session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so open
19 session optional pam_keyinit.so force revoke
20 session required pam_limits.so
21 session required pam_env.so readenv=1
22 session required pam_env.so readenv=1 user_readenv=1 envfile=/etc/default/locale
23 @include common-session
24 session optional pam_gnome_keyring.so auto_start
25 @include common-password
```

A continuación, le añadimos delante un carácter almohadilla (#). De esta forma la convertimos en un comentario, haciendo que su contenido no tenga efecto.



```
1 #%PAM-1.0
2 auth requisite pam_nologin.so
3 #auth required pam_succeed_if.so user != root quiet_success
4 @include common-auth
5 auth optional pam_gnome_keyring.so
6 @include common-account
7 # SELinux needs to be the first session rule. This ensures that any
8 # lingering context has been cleared. Without this it is possible
9 # that a module could execute code in the wrong domain.
10 session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so close
11 session required pam_loginuid.so
12 # SELinux needs to intervene at login time to ensure that the process
13 # starts in the proper default security context. Only sessions which are
14 # intended to run in the user's context should be run after this.
15 # pam_selinux.so changes the SELinux context of the used TTY and configures
16 # SELinux in order to transition to the user context with the next execve()
17 # call.
18 session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so open
19 session optional pam_keyinit.so force revoke
20 session required pam_limits.so
21 session required pam_env.so readenv=1
22 session required pam_env.so readenv=1 user_readenv=1 envfile=/etc/default/locale
23 @include common-session
24 session optional pam_gnome_keyring.so auto_start
25 @include common-password
```

Y para terminar, debemos realizar un último ajuste en el archivo `.profile`. Por lo que comenzaremos por abrirlo:

```
lorena@lorena-VirtualBox:~$ sudo gedit /root/.profile
```

Cuando lo tengamos en pantalla, localizamos una línea con el siguiente contenido:

```
mesg n 2> /dev/null || true
```

Y lo sustituimos por esto:

```
if `tty -s`; then
    mesg n
fi

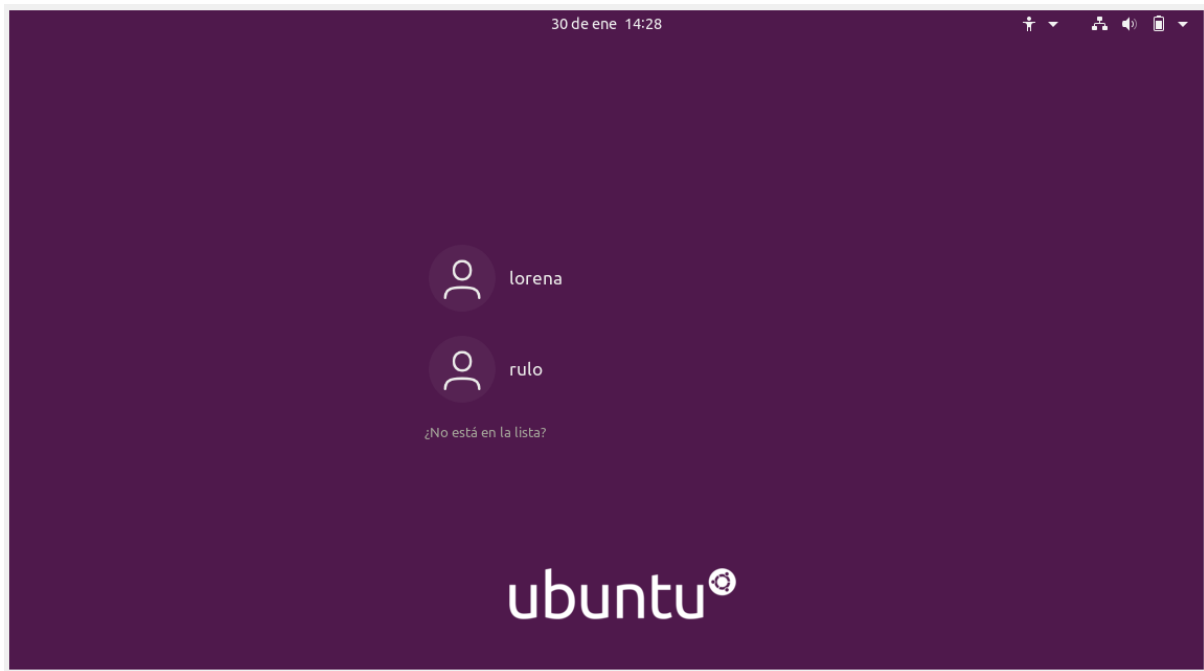
# ~/.profile: executed by Bourne-compatible login shells.

if [ "$BASH" ]; then
    if [ -f ~/.bashrc ]; then
        . ~/.bashrc
    fi
fi

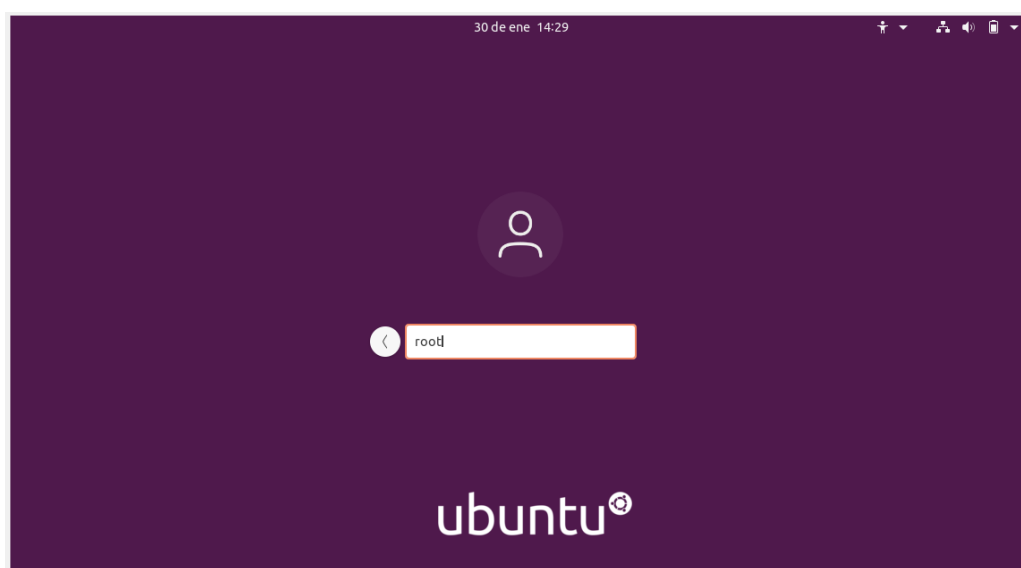
if `tty -s`; then
    mesg n
fi
```

## Iniciar sesión con el usuario root en la interfaz gráfica

La próxima vez que inicies tu ordenador, comprobarás que el usuario **root** sigue sin aparecer en la pantalla de autenticación.



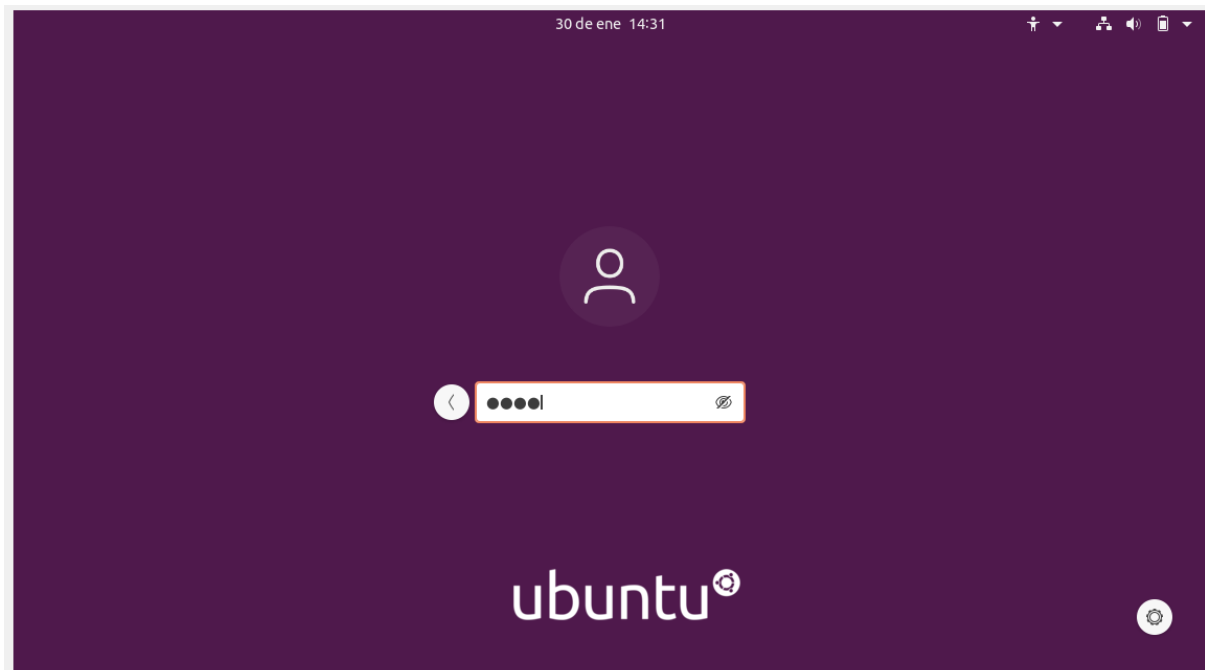
Sin embargo, para iniciar sesión, ahora solo tendremos que hacer clic sobre el enlace *¿No está en la lista?*. Al hacerlo, el sistema nos solicita el nombre de la cuenta de usuario que queremos usar.



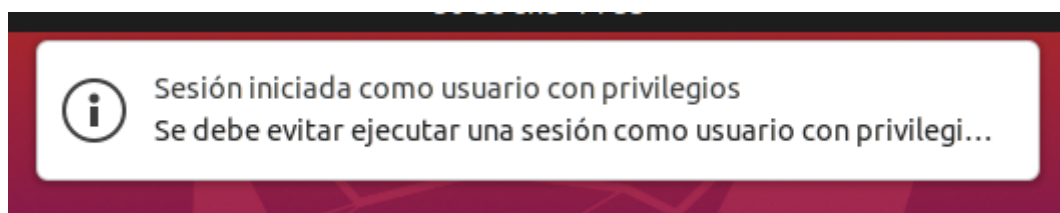


Escribimos root y pulsamos INTRO o damos al botón siguiente:

A continuación, deberemos escribir la contraseña que asignamos en el primer apartado de este artículo.



Y volvemos a pulsar INTRO.

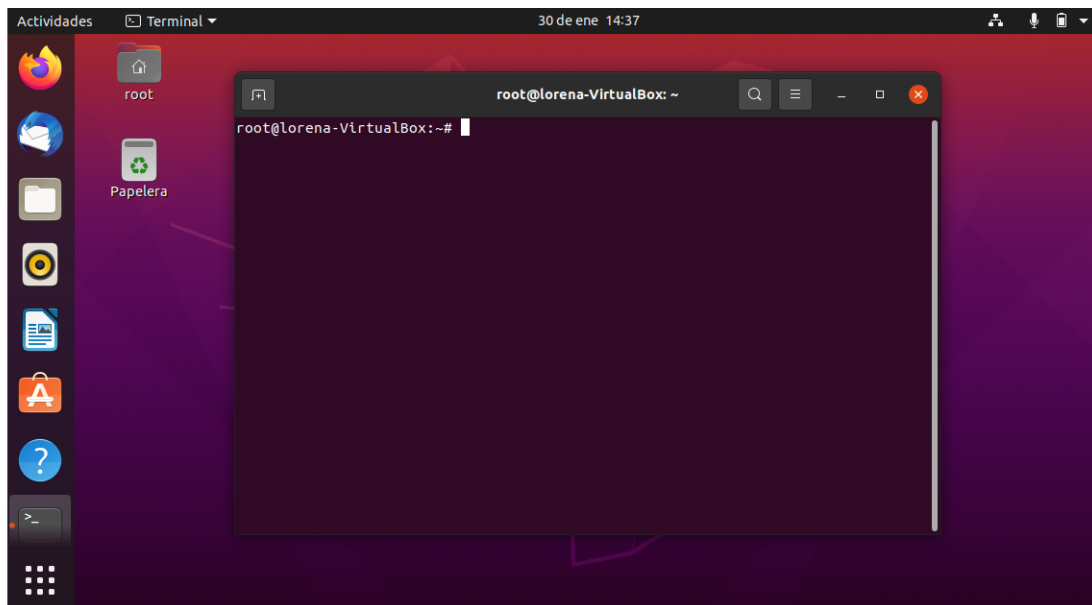


Al iniciar la sesión, veremos que el sistema nos muestra una notificación avisándonos del peligro que supone iniciar sesión con la cuenta **root**.

Además, como ocurre siempre que iniciamos sesión por primera vez con una cuenta de usuario, el sistema nos ofrece una *pantalla de bienvenida* con cinco etapas. En ella nos ofrece información sobre diferentes aspectos del sistema.

Podemos avanzar usando el botón que aparece en la parte superior derecha de la ventana. Al principio, el botón muestra el texto *Omitir*, en las etapas que la siguen aparece *Siguiente* y, en la última, *Hecho*.

Y para comprobar que hemos iniciado sesión con el usuario correcto, basta con abrir una ventana de terminal.



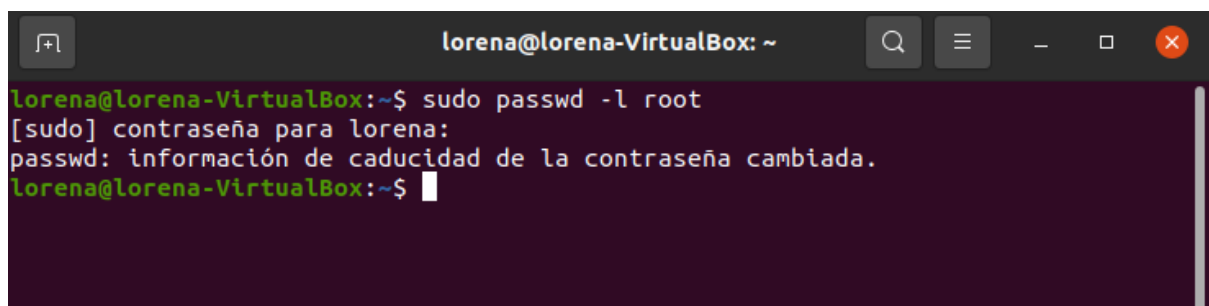
### Volver a deshabilitar la cuenta de usuario root

Como puedes suponer, no se recomienda que utilices la cuenta **root** de forma continuada.

Si la has activado por un corto periodo de tiempo, para realizar tareas de administración, es muy aconsejable volver a desactivarla en cuanto sea posible.

Para conseguirlo, podrías deshacer todos los cambios realizados más arriba. Sin embargo, puede bastar con abrir la consola desde la cuenta de usuario predeterminado y ejecutar la siguiente orden:

***sudo passwd -l root***



De este modo, la próxima vez que tratemos de iniciar sesión, recibiremos un error de autenticación.

Lo que hemos hecho es bloquear la cuenta **root** deshabilitando su contraseña. En realidad, se ha cambiado su valor por otro que no coincida con ningún posible valor cifrado. En concreto, se añade un carácter '!' al principio de la contraseña.

De cualquier modo, la cuenta **root** no ha quedado deshabilitada. De hecho, si estuviesen activos otros métodos de autenticación, como una clave *SSH*, sería posible seguir iniciando sesión.

Otra alternativa es utilizar el comando **usermod** con una sintaxis como esta:

```
lorena@lorena-VirtualBox:~$ sudo usermod --lock --expiredate 1970-01-02 root  
[sudo] contraseña para lorena:
```

Con esto, establecemos la fecha de caducidad de la cuenta root el 2 de enero de 1970.



## 1.4. Consola

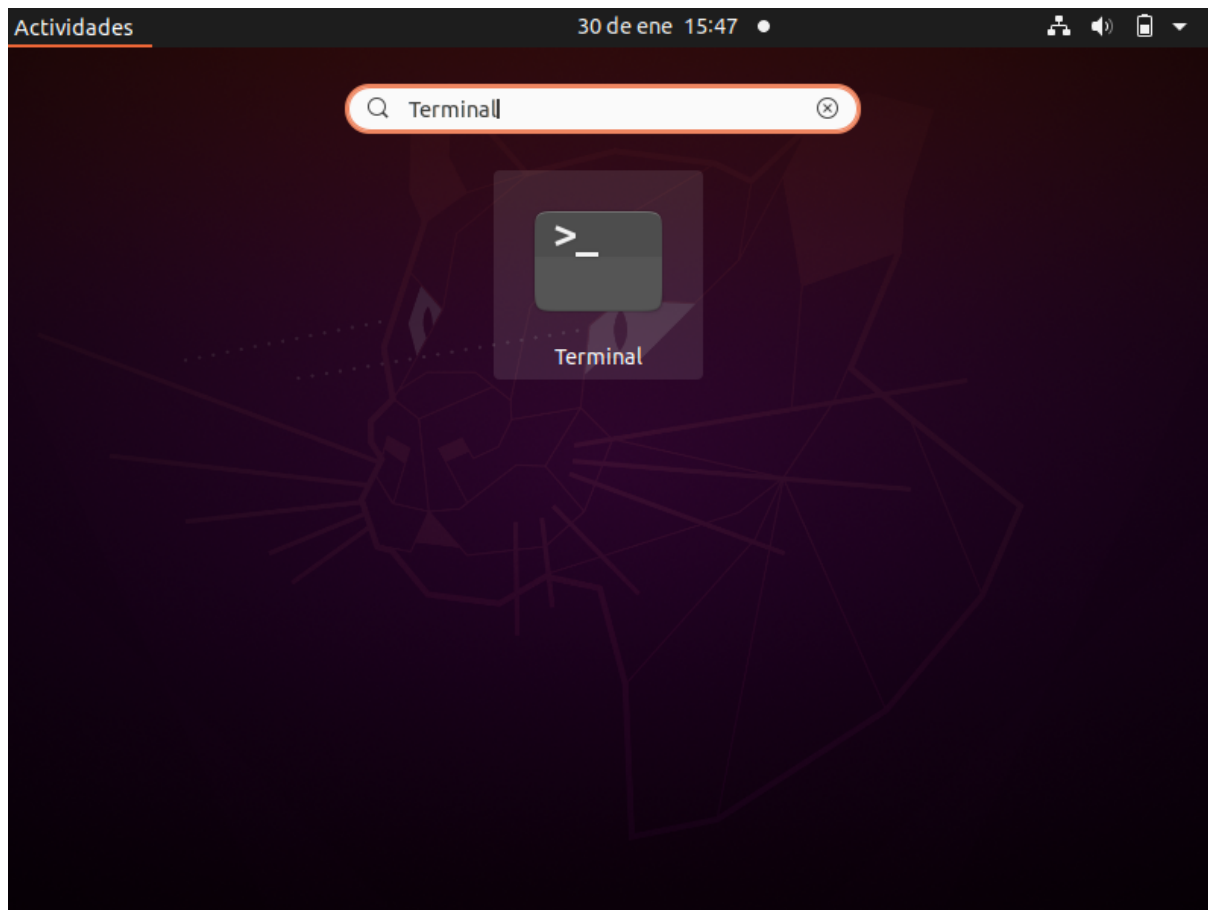
Trabajar con la línea de comandos no es una tarea tan desalentadora como muchos pudieran pensar. No se requieren conocimientos especiales para usar la línea de comandos, pues es un programa como otro cualquiera. La mayoría de las acciones realizadas en Linux pueden llevarse a cabo usando la línea de comandos. Aunque existen herramientas gráficas para la mayoría de programas, a veces esto no es suficiente. Entonces es cuando la línea de comandos cobra su utilidad.

La terminal es llamada a menudo la línea de comandos, el "prompt", o el "shell". Antes, éste era el único método por el cual el usuario interactuaba con el ordenador; sin embargo, muchos usuarios de Linux encuentran más rápido el uso de la terminal que un método gráfico e incluso hoy día tiene algunas ventajas. Aquí aprenderá cómo usar la terminal.

El uso original de la terminal era como administrador de ficheros y de hecho todavía es usado con este fin. Usted puede usar la terminal como un navegador de archivos para administrar sus ficheros y deshacer los cambios realizados.

### Iniciando la consola

Para abrir el terminal lo podemos hacer pulsando Ctrl+Alt+T o  , que desplegará la siguiente pantalla, donde escribiremos Terminal y nos aparecerá  el icono de terminal.



## Órdenes comunes

- **Ver directorios:** `ls`. El comando `ls` (LiSta) muestra los archivos diferenciados por colores y texto con formato
- **Crear directorios:** `mkdir` (nombre del directorio). El comando `mkdir` (del inglés MaKeDIRectory) crea un directorio.
- **Cambiar de Directorio:** `cd` (/directorio/ubicación). El comando `cd` (Cambiar de Directorio) cambia de su directorio actual a cualquier directorio especificado.
- **Copiar Archivos/Directorios:** `cp` (nombre de archivo o directorio de origen) (directorio o archivo de destino). La orden `cp` (CoPiar) copia los archivos que usted especifique. La orden `cp -r` copia cualquier directorio que usted especifique.
- **Eliminar Archivos/Directorios:** `rm` (el nombre del archivo o directorio). El comando `rm` (del Inglés ReMove, eliminar) borra el fichero con el nombre especificado. El comando `rm -r` borra el directorio especificado junto con todo su contenido.
- **Mover/Renombrar Archivos/Directorios:** `mv` (archivo o directorio). El comando `mv` (MoVer) mueve/renombrar el archivo o directorio especificado.

- **Buscar Archivos/Directorios:** `locate` (nombre del archivo o directorio). El comando **`locate`** busca cualquier archivo especificado. Emplea un índice de los archivos que hay en su sistema para acelerar la búsqueda. Para actualizar este índice, ejecute el comando **`sudo updatedb`**. Este comando se ejecuta de manera automática cada día si deja su ordenador encendido. Para ejecutarlo se requieren privilegios de administrador.

También puede emplear comodines para encontrar uno o mas archivos, como `"*"` (para todos los archivos) o `"?"` (para encontrar un caracter).

## Cambiar a modo Consola

El método habitual para acceder a la línea de comandos en Ubuntu es iniciando una sesión de terminal, pero a veces es útil entrar en la auténtica consola:

- Use la combinación de teclas **`Ctrl-Alt-F1`** para pasar a la primera consola.
- Para regresar al modo Escritorio, use el siguiente atajo de teclado: **`Ctrl-Alt-F7`**.
- Existen seis consolas disponibles. Para acceder a cada una de ellas use las combinaciones entre **`Ctrl-Alt-F1`** y **`Ctrl-Alt-F6`**.

### 1.5. Edición de texto.

Todas las configuraciones y ajustes en Linux son almacenados en ficheros de texto. A pesar de que es posible editar las configuraciones a través de la interfaz gráfica, en ocasiones puede que tenga que editarlas a mano. **Gedit** es, por defecto, el editor de texto de Ubuntu, que puede ser lanzado como hemos explicado anteriormente.

Hay que tener en cuenta que si necesitamos que ese fichero sea abierto con privilegios de administrador debemos abrirlo con la orden `sudo` previa a la llamada.

Si necesitas usar un editor de texto en modo consola, puedes emplear **nano**, un editor de texto de uso sencillo. Cuando lo ejecute desde la línea de comandos, emplea siempre el siguiente comando para evitar que el editor introduzca saltos de línea: **`nano -w`**

Para más información sobre el editor de texto nano ir a : <https://www.nano-editor.org/>