

Sistemas Operativos en Red

UD08 - Repaso - Administración básica de Linux



Autor: Sergi García

Actualizado Diciembre 2024



Licencia



Reconocimiento - No comercial - CompartirIgual (BY-NC-SA): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se ha de hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán diferentes símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

Importante

Atención

Interesante

ÍNDICE

1. ¿Qué es Linux?	4
1.1 GNU/Linux	4
1.2 ¿Qué es una distribución?	5
1.3 ¿Dónde encontramos Linux?	5
2. Licencias de software	5
3. Instalación	6
3.1 Descarga	6
3.2 Instalación	6
3.3 El disco duro	9
3.4 Particiones	10
3.5 El sector de arranque	12
3.6 Creando nombre de usuario de administrador	13
4. Aplicaciones	13
4.1 Aplicaciones básicas	13
4.1.1 Explorador de archivos	13
4.1.2 Editor de texto	14
4.1.3 La terminal	15
4.2 Cómo instalar programas en Linux	15
4.2.1 Instalación usando paquetes	15
4.2.2 Instalación usando paquetes completos	17
5. Usuarios en Linux	17
5.1 Archivos "/etc/passwd" y "/etc/shadow"	17
5.2 Comando "sudo" y lista de sudoers	18
5.3 Comando "su"	18
5.4 Creando usuarios en Linux	19

6. Grupos en Linux	19
6.1 Fichero "/etc/grupo"	19
6.2 Creando grupos en Linux	19
7. Ficheros y directorios en Linux	19
7.1 Tipos de archivos	19
7.2 Ficheros ocultos	20
8. Permisos en Linux	20
8.1 Algoritmo de concesión de permisos	21
8.2 Usando el comando "chmod" para establecer permisos	21
8.3 Permisos especiales	21
9. Principales comandos de Linux	22
10. Bibliografía	24


UD08 - REPASO - ADMINISTRACIÓN BÁSICA DE LINUX

1. ¿QUÉ ES LINUX?

La forma más sencilla de definir Linux es que es un sistema operativo estilo Unix. A estas alturas ya sabemos que es un sistema operativo, pero ¿qué es Unix?

Aunque no es el primer sistema operativo, Unix es, sin duda, el primer **GRAN** sistema operativo. Sus características más importantes fueron que es portable (puede ejecutarse en diferentes sistemas informáticos), multitarea y multiusuario. Además, arquitectónicamente hablando, fue creado basándose en conceptos como simplicidad y modularidad para que el código fuera fácilmente mantenido y ampliado por otros programadores.

Con el paso de los años, la empresa creadora (los Bell Labs¹) fue licenciando el producto a otras empresas que, para adaptarlo a entornos más específicos, fueron realizando modificaciones y creando diferentes versiones. De ahí nacieron productos como Xenix (Microsoft), HP-UX (HP), IRIX (Silicon Graphics), SCO (Novell), AIX (IBM), etc.


 **Interesante:** Todos los sistemas de la familia UNIX suelen denominarse *IX.


El problema de todas ellas es que son versiones privativas, por lo que el código no está disponible para su estudio (aparte de la gran cantidad de dinero que cuesta una licencia). Es por eso que, con un objetivo puramente educativo, a finales de los años 80, un profesor de la Universidad de Ámsterdam (Andrew Tanenbaum) decidió crear “Minix”, un sistema operativo basado en la filosofía UNIX pero reescrito desde cero y de código abierto. Debido a su carácter educativo, el autor decidió no permitir modificaciones que muy probablemente complicarían mucho más el código.

Es en este momento cuando un estudiante finlandés de informática decide, basándose en “Minix”, crear un clon gratuito que funcione en sistemas PC. Este estudiante fue Linus Torvalds, y lo llamó sistema operativo “Linux”.

1.1 GNU/Linux

En el apartado anterior comentábamos que Linux era un sistema operativo, pero eso no es exactamente cierto. Todo sistema operativo consta de un kernel o núcleo y un conjunto de aplicaciones que ayudan a hacer posible el funcionamiento del sistema operativo. De hecho, una posible clasificación de los diferentes programas que acompañan al kernel en un sistema operativo podría ser: el shell o terminal (que permite la interacción con el usuario en modo texto), servicios o demonios (que son programas que se ejecutan en segundo plano), un servidor gráfico (que permite dibujar elementos en pantalla) o un escritorio (que aprovecha las funciones del servidor gráfico para proporcionar acceso gráfico al usuario).

 **Atención:** Linux es simplemente el kernel del sistema, kernel que para formar el SO va acompañado de muchas utilidades GNU³. Ese sistema operativo no se llama Linux, se llama GNU/Linux.

 **Atención:** normalmente, todo el mundo simplifica y llama a Linux el sistema operativo (erróneamente).

1.2 ¿Qué es una distribución?

El núcleo y las utilidades básicas constituyen el núcleo del sistema operativo, pero sabemos que hoy en día un sistema operativo se ve abandonado por una gran cantidad de software que no es necesario para la computadora en sí (programas de dibujo, editores de texto, etc.). El hecho de que el kernel y las utilidades básicas sean de distribución gratuita permite que cualquiera pueda coger estos elementos y acompañarlos de otras utilidades (ya sean básicas o no) según le guste o sus necesidades. Así surgen las distribuciones de Linux o, más comúnmente, distros.

Hay cientos de ellos en el mercado, pero por mencionar algunos de los más conocidos: “Ubuntu” (y sus diferentes “variantes” como “Lubuntu”, “Xubuntu”, “Kubuntu”, “Ubuntu Server”, etc.), “Open Suse”, “Arch”, “Fedora”, “Debian”, “Red Hat”, “Mint”, “Lliurex”.

Usaremos “Lubuntu” a lo largo de este curso. “Lubuntu” es una distribución basada en Ubuntu que pretende ser liviana, y es el principal motivo para elegirla, porque es más fácil de virtualizar.

1.3 ¿Dónde encontramos Linux?

Una de las ideas más extendidas es la de que Unix es un sistema operativo que solo se utiliza en entornos académicos o de alto nivel técnico, y nada más lejos de la realidad. Los sistemas basados en versiones libres de Unix están implantados en muchos sistemas informáticos. Aunque posiblemente en entornos de escritorio los sistemas de Microsoft sigan siendo los que dominan la mayor parte del mercado, en dispositivos móviles como en servidores la realidad es bien distinta.

Por ejemplo, los dos sistemas operativos móviles por excelencia (Android e iOS), son sistemas derivados de Linux o FreeBSD4 (así como sistemas como macOS).

2. LICENCIAS DE SOFTWARE

En varias ocasiones, a lo largo de apartados anteriores hemos comentado que una de las características fundamentales de Linux es el hecho de que es libre. Pero, ¿qué significa ser libre?

Por el uso de la misma palabra en inglés (“free” significa tanto “libre” como “gratis”), muchas personas piensan que ser libre implica ser gratis (no cobrar). El precio del software no tiene nada que ver con el tipo de licencia al que te sometas.

La licencia de software es un contrato que define todas las reglas que rigen el uso de un programa en particular. El contrato se realiza entre el propietario del programa y el usuario de la aplicación.

Sus cláusulas determinan cuestiones como el plazo de cesión de derechos, el ámbito geográfico de validez del contrato, límites de responsabilidad por fallos, número de copias permitidas, posibilidad o no de transferir el software a terceros, etc.

Existen varios tipos de licencias, pero en general todas podrían incluirse en alguno de estos tipos:

- **Licencia propietaria:** uso de software en una o más máquinas por un costo determinado. No suele estar incluido en el código del programa, solo el archivo ejecutable.
- **Licencia “shareware”:** uso de software con funcionalidades limitadas por un tiempo (o incluso de forma definitiva) y necesidad de pago para ampliarlas.
- **Licencia “freeware”:** uso ilimitado y copia a coste cero.
- **Software libre:** permite el uso, copia, modificación y distribución gratuita con acceso al código fuente. Dentro de estos últimos, podemos encontrar dos tipos:
 - **Licencias permisivas:** aquellas que dan total libertad al usuario, para que con el código modificado pueda hacer lo que quiera. Es decir, aunque el original sea código libre, la modificación puede ser privada. Una licencia de este tipo es BSD.

- **Licencias no permisivas:** aquellas que exijan que la copia, modificación y posterior distribución se realice en las mismas condiciones que el original. Es decir, si el original es código libre, la copia debe seguir siéndolo. Una licencia de este tipo es la GPL, que utilizan los productos GNU.

3. INSTALACIÓN

3.1 Descarga

El primer paso es obtener y grabar el ISO de “Lubuntu”. Puedes descargarlo desde:

<https://lubuntu.me/>

Después de la descarga, el siguiente paso es grabar la imagen ISO en un CD/DVD (con un programa de grabación como Brasero, Nero, Toast, ISO Burner, etc.) o en una memoria USB (con UltraISO, Rufus, etc.).

3.2 Instalación

Reinicia el ordenador con el DVD grabado con “Lubuntu” en el lector de DVD o, si usas un USB, con el pendrive conectado. Si dejamos que el sistema se inicie, acabará iniciando una versión de “Lubuntu” en modo gráfico con opciones para instalarlo en el disco duro.

! Atención: para arrancar desde DVD o USB es necesario tener activada esta opción y/o modificar el orden de arranque del sistema. Estas opciones se pueden configurar desde SETUP.

Para llegar a él, en el menú de inicio elegimos la opción Probar o Instalar Lubuntu” y luego, cuando Live USB haya iniciado, seleccionamos la herramienta “Instalar Lubuntu” como se ve en las siguientes dos figuras:

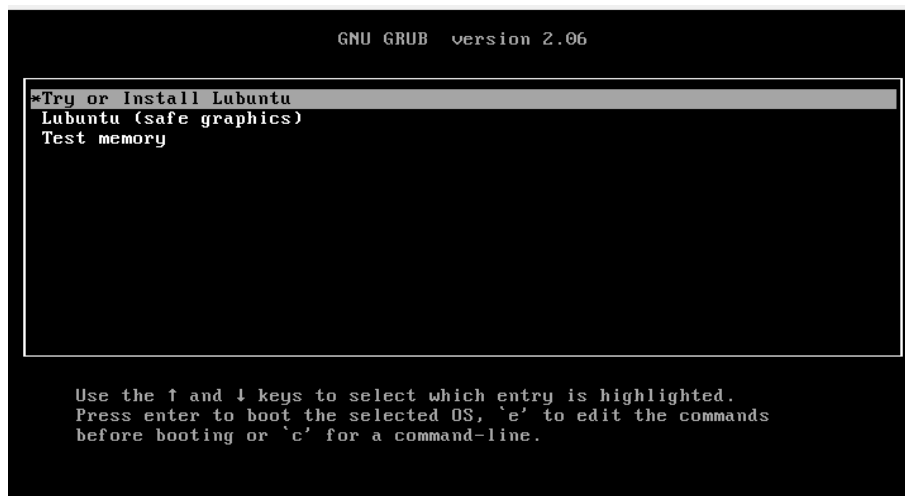


Figura 1. Lubuntu Live USB/DVD Boot menu



Figura 2. Escritorio Lubuntu Live USB/DVD con herramienta “Instalar Lubuntu”.

Cuando hayamos realizado esos pasos, comenzará la instalación. Hoy en día, la instalación de Linux es muy sencilla. De hecho, en la gran mayoría de los casos, aceptando las opciones por defecto y asignando un usuario, el sistema se instala correctamente.

Pero para que la instalación sea más flexible y se adapte mejor a las necesidades del usuario, conviene entrar en detalle en la ventana de tipos de instalación.

En primer lugar, la instalación nos preguntará nuestro idioma, nuestra región y nuestra distribución de teclado, como podéis ver en las siguientes figuras:

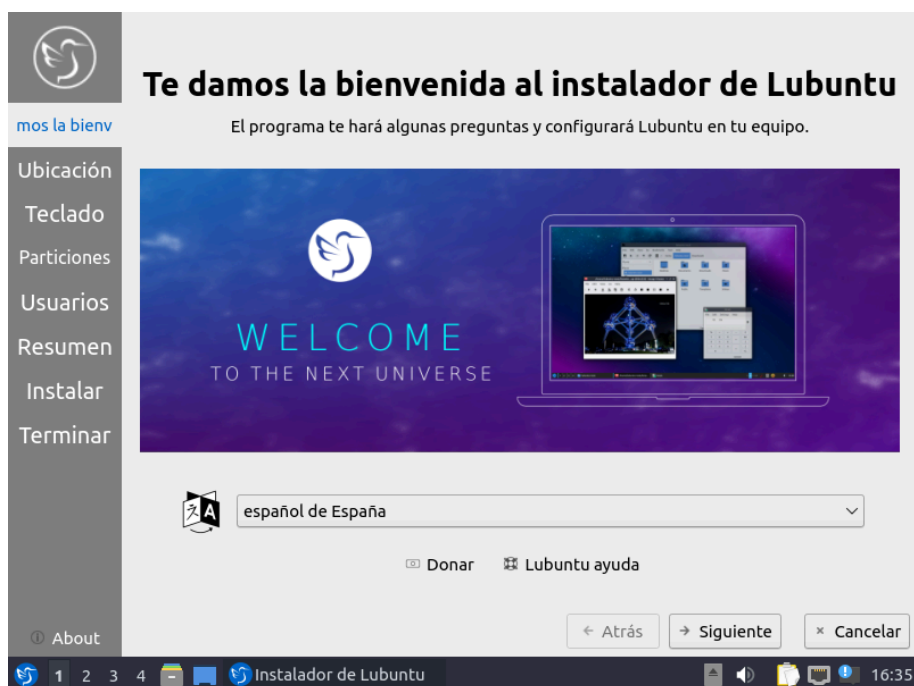


Figura 3. Lubuntu preguntando por nuestro idioma.

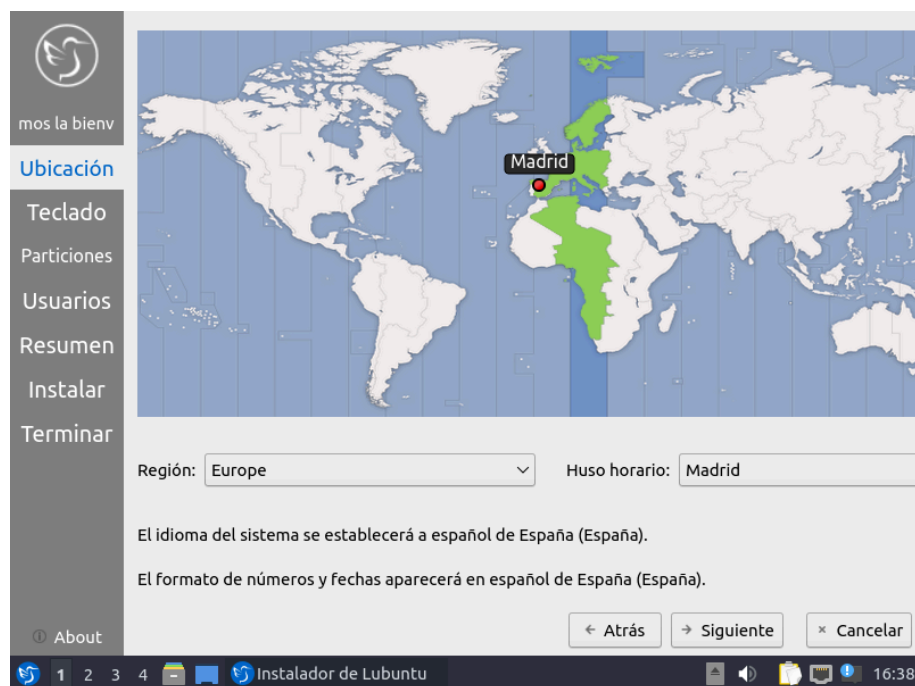


Figura 4. Ubuntu preguntando por nuestra región.

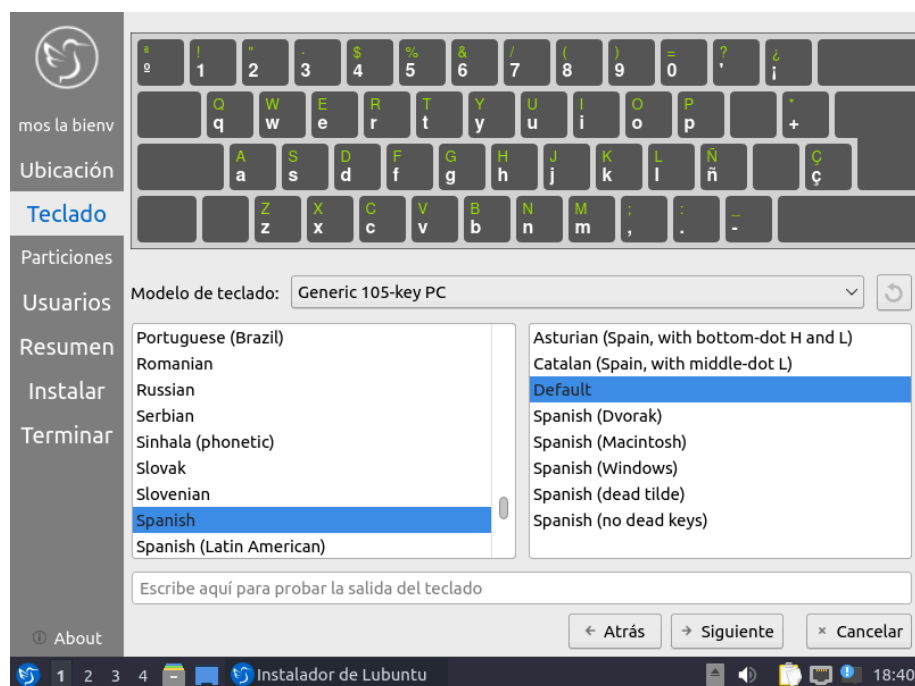


Figura 5. Ubuntu preguntando por nuestra distribución de teclado.

Cuando hayamos finalizado esos sencillos pasos, debemos seleccionar “particionamiento manual” para poder crear particiones en el disco en el que vamos a realizar la instalación, así como realizar una distribución óptima de las particiones a crear. **Este es un paso importante.**

! Atención: en la mayoría de las instalaciones, puedes usar todo el disco (por ejemplo, una instalación sola en una máquina virtual) pero en algunas situaciones, como un arranque dual con Windows, debes tener cuidado y no destruir la partición de Windows.

Para realizar cambios manuales en la tabla de particiones, debemos seleccionar el último elemento, como en la siguiente figura:

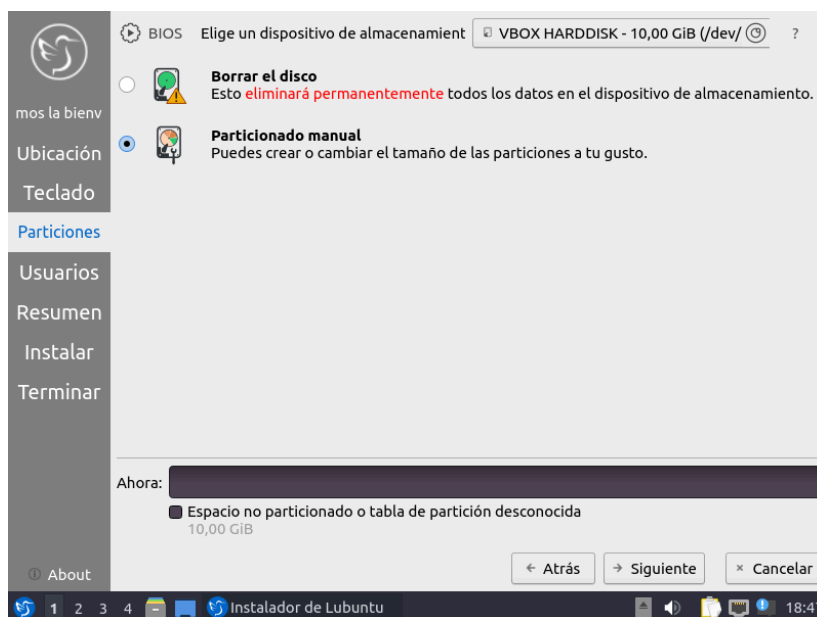


Figura 6. Partición manual de selección de Ubuntu

3.3 El disco duro

Cuando hayamos seleccionado “particionado manual” debemos seguir varios pasos.

El primer paso es elegir en qué disco queremos realizar la instalación. En Linux, los dispositivos se simulan utilizando archivos que se encuentran en el directorio de desarrollo. Hay muchos tipos, pero en cuanto a elementos conectados en los conectores PATA o SATA, la forma de llamarlos es: tipo + orden + partición. Los tipos son “**hd**” para dispositivos PATA y “**sd**” para dispositivos SATA. El orden hace referencia al elemento del número de puerto en el que están conectados. Para ello se utilizan letras comenzando por la letra “**a**”. Las particiones están numeradas con números, comenzando por el número 1. **Por ejemplo, "sdb1" se refiere a la primera partición en un dispositivo conectado al segundo puerto SATA.**

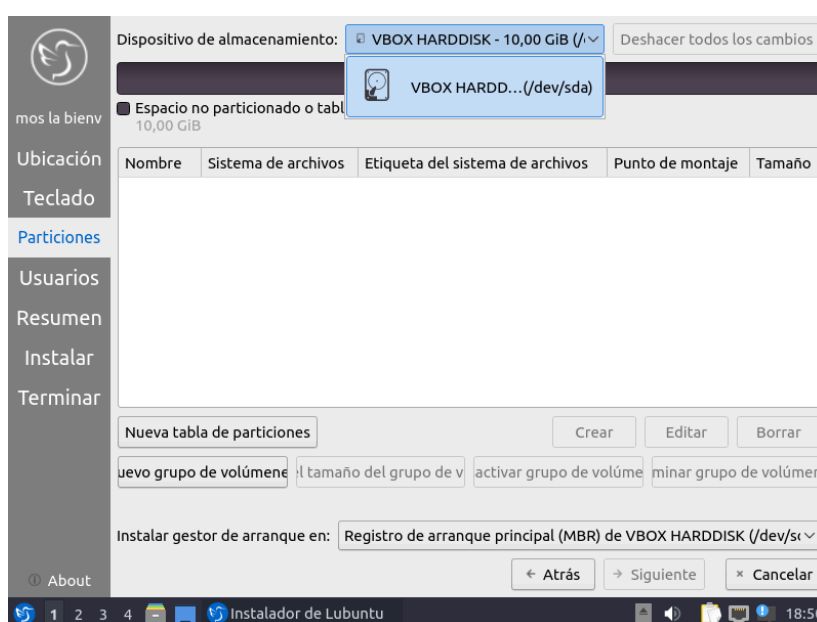


Figura 7. Ubuntu con disco duro en /dev/sda

En la última figura puedes ver que el ordenador tiene conectado únicamente un disco duro SATA ("**sda**"), por lo que es el único que puede utilizarse para realizar la instalación.

En caso de que hubiera más ("**sdb**", "**sdc**", "**sde**", etc.), estas aparecerían en la lista para que elijamos en cuál de ellas queremos instalar.

3.4 Particiones

Habitualmente hemos asociado la idea de que un programa está instalado en un disco duro. De hecho, en el apartado anterior hemos dejado entrever que es el primer paso para realizar la instalación de un SO. Sin embargo, un sistema operativo es un software especial y puede distribuirse en diferentes particiones de un disco duro o incluso en diferentes discos duros.

De hecho, esta posibilidad es una obligación para cualquier usuario con conocimientos medios. Aparte de la seguridad que proporciona el hecho de tener contenidos separados (limitada si la instalación se realiza en diferentes particiones de un mismo disco duro, ya que si el disco duro falla todas las particiones fallan), tiene la gran ventaja de que la reinstalación del sistema en caso de error (por ejemplo, al restaurar una imagen) es muy rápido y no dramático.

¿Cómo podemos distribuir las particiones? ¿Qué podemos incluir en cada uno de ellos?

Linux siempre crea por defecto al menos 2 particiones: una para el sistema y otra, llamada swap, para utilizar la memoria virtual. En el momento que decidamos realizar la instalación personalizada, debemos recordar que debemos crear una partición de intercambio. Su tamaño recomendado es el doble de la memoria RAM de la máquina, aunque pensando en posibles ampliaciones de la misma y teniendo en cuenta que la capacidad de los discos duros hoy en día es grande, una buena opción es asignar cuatro veces más.

Para ver cómo podemos distribuir el resto del sistema, es necesario entender cómo funciona el sistema de archivos de Linux. A diferencia de los sistemas de Microsoft, en los que a cada partición se le asigna una unidad (cada una de las cuales se llama usando una letra del alfabeto, "a:", "b:", "c:", "d:", "e:", etc.) en sistemas Unix el usuario dispone de una única unidad llamada "/" (barra o directorio raíz) desde la que conectar (montar) en diferentes directorios la información de todas las particiones de discos duros existentes en el ordenador,

Evidentemente, toda esta información está organizada de forma que sea sencilla su ubicación.

Por ejemplo:

- **/boot**: esta carpeta tiene archivos para la gestión de arranque.
- **/root**: contiene programas de administrador del sistema (superusuario).
- **/var**: contiene archivos variables, como registros, bases de datos, colas de correo, etc.
- **/tmp**: archivos temporales.
- **/usr**: programas y datos del sistema que pueden ser compartidos por múltiples usuarios.
- **/home**: carpetas del usuario.
- **/dev**: archivos que encapsulan diferentes dispositivos físicos conectados al sistema. Por ejemplo, **/dev/lp** es una impresora.
- **/opt**: aplicaciones de terceros (similares a archivos de programas de Windows).
- **/lib**: bibliotecas comunes a todas las aplicaciones.
- **/etc**: archivos de configuración de aplicaciones.
- **/mnt**: en esta carpeta se ubican los directorios de cada una del resto de particiones de todos los discos duros de la computadora. Por ejemplo, si tuviera un disco duro con una partición con Windows, su ubicación sería **/mnt/windows** (el nombre de la carpeta podría ser cualquiera).
- **/media**: similar al anterior, pero con dispositivos que se montan en vivo (USB o tarjeta SD).

La distribución de tabiques se podrá realizar siguiendo esta estructura. Así, por ejemplo, si una partición tiene asignado “/”, en esa partición instalaremos todo el sistema (que es lo que se hace en una instalación por defecto).

Una distribución más consistente es crear una partición para todo el sistema y otra donde se almacenan las carpetas del usuario (/home). De esta forma, el sistema queda separado de los datos, facilitando la reinstalación. Otra opción interesante es, además de estas dos, crear una partición donde almacenar /var, pensando en separar otros tipos de datos como bases de datos.

Interesante: las carpetas indicadas anteriormente son solo algunas de ellas. Y no todos pueden estar aislados en una partición. Por ejemplo, “/mnt” y “/media” no tiene ningún sentido que estén en una partición separada, ya que ya son en sí mismos un sitio donde se van a incorporar otras particiones.

Cada una de estas particiones puede estar en el disco duro que queramos. En nuestro caso usaremos un único disco duro y crearemos 3 particiones: “/”, “/home” y, por supuesto, “swap”.

Para ello simplemente selecciona el espacio vacío del disco duro donde desees crear la partición y presiona el botón +. Esto abre una ventana donde puedes asignar un tamaño (en MB), el tipo, el tipo de sistema de archivos (generalmente “ext4”) y el punto de montaje que desees incluir en esa partición.

Interesante: el acto de incluir una partición en nuestro sistema de archivos general se llama montaje. Así, por ejemplo, cuando conectamos un USB, el sistema lo que hace es que automáticamente monta esa unidad en la carpeta “/media”.

Aunque el tipo de partición es un tema que hoy en día no es tan importante como hace unos años, al menos es interesante saber que existen dos tipos de particiones: las primarias y las extendidas. Un disco duro solo puede contener 4 particiones primarias. Antiguamente, era más que suficiente, ya que el tamaño de los discos duros no alcanzaba para mucho más, pero con el aumento de capacidad surgió la necesidad de crear más particiones.

Ante la imposibilidad de aumentar el número de particiones primarias, se creó el concepto de partición extendida, de tal forma que en un disco duro se pueden crear como máximo 3 primarias y una extendida. La gran ventaja del ampliado es que en su interior podrían existir tantos tabiques como se desee. Estas particiones incluidas dentro de la partición extendida se denominan particiones lógicas. A efectos prácticos, el tipo de partición no tiene mucha importancia, aunque suele ser recomendable que las particiones principales y el sistema de arranque estén en particiones primarias.

En el resto de figuras podéis ver cómo se distribuyen las nuevas particiones en nuestro disco duro.

Atención: si quieres saber más sobre cómo funciona y por qué es importante tener una partición de intercambio, puedes consultar:

<https://haydenjames.io/linux-performance-almost-always-add-swap-space/>

<https://haydenjames.io/linux-performance-almost-always-add-swap-part2-zram/>

Atención: obviamente, tenemos que tener en cuenta que cuando aceptemos la distribución, toda la información que haya almacenado previamente en ese disco duro será destruida.

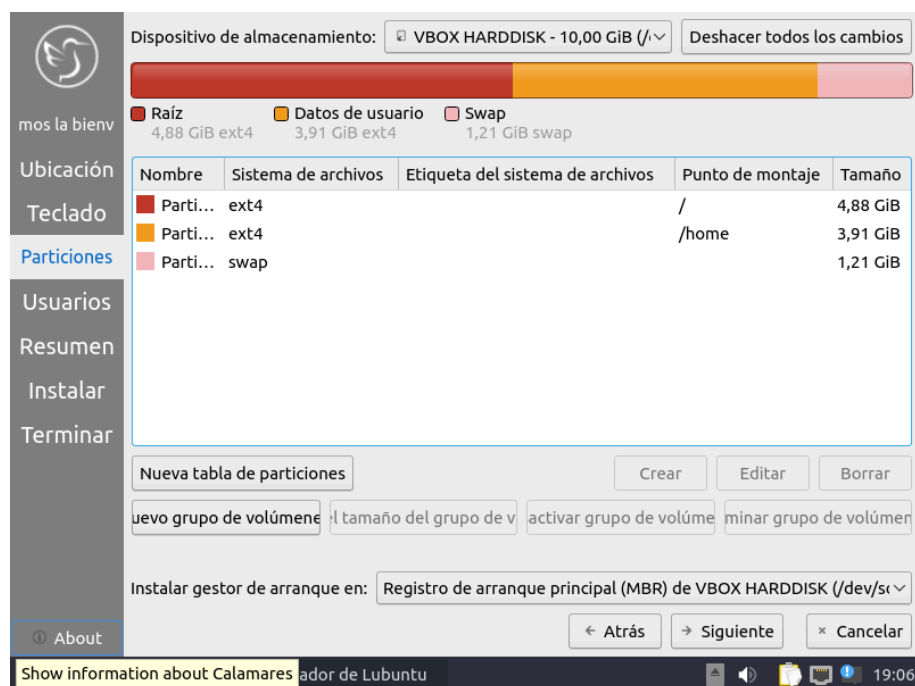


Figura 8. Disco “/dev/sda” particionado en “/”, “/home” y partición de intercambio

3.5 El sector de arranque

El último paso es seleccionar en qué disco duro queremos instalar el gestor de arranque. Como vimos en lecciones anteriores, en la BIOS se configura un orden para buscar, en los diferentes sistemas de almacenamiento, un sector (MBR) que permita ejecutar la carga de un sistema operativo. El programa BIOS, encargado de esa tarea, busca esa lista de dispositivos hasta encontrar uno que permita arrancar. Es por eso que cada sistema operativo tiene que instalar y configurar ese sector, independientemente de la instalación del propio sistema operativo. Esto se consigue fácilmente cuando en el ordenador solo hay un sistema operativo, pero ¿qué pasa cuando se va a instalar más de uno?

Con el primer sistema a instalar no habrá problema, pero cuando instalemos el resto, cada uno instalará el sector de arranque sobrescribiendo el anterior. De esta forma, aunque los sistemas operativos estén instalados, solo podrás arrancar el último que se instaló. Afortunadamente, las distribuciones Linux suelen ser muy respetuosas en ese aspecto, por lo que si al instalar su sector de arranque detectan la existencia de otro, no lo sobrescriban, sino que modifiquen un pequeño programa (normalmente llamado GRUB) para que muestre un menú que permita seleccionar el sistema operativo que desea iniciar.

Interesante: sin embargo, si el sistema instalado posteriormente es uno de la familia Microsoft, solo toma en cuenta los sistemas que pertenecen a la misma familia. Entonces, si desea tener varios sistemas operativos en la misma computadora, instale primero los de la familia Microsoft.

Interesante: Si tienes problemas con el sector de arranque, **recuerda que el sistema operativo está ahí, el único problema es acceder a él.** Para eso existen varias utilidades para realizar una copia de seguridad, restaurarlo o repararlo en caso de problemas. Por ejemplo, “Rescatux” es un buen “Live DVD/USB” para ayudar con ese tipo de problema <https://www.supergrubdisk.org/rescatux/>

3.6 Creando nombre de usuario de administrador

El último paso importante, antes de empezar a copiar archivos en disco, es crear un usuario con perfil de administrador. Es muy fácil, como puedes ver en la siguiente figura:

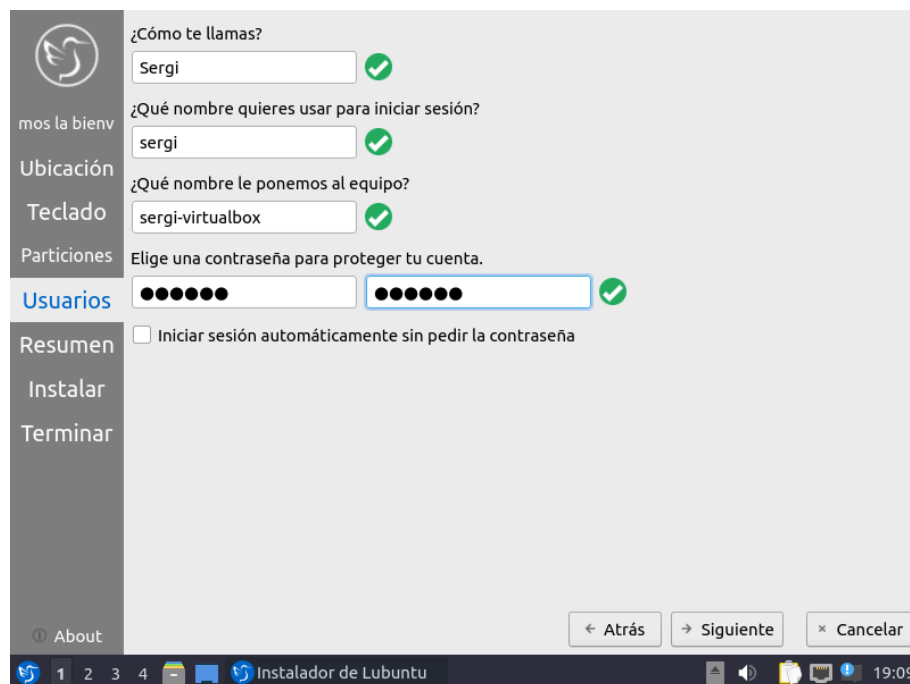


Figura 9. Creación de usuario con perfil de administrador

Cuando hayamos finalizado este paso, podremos comenzar a copiar el sistema operativo al disco y cuando finalice la instalación, nuestro sistema Ubuntu estará listo para funcionar :)

4. APLICACIONES

4.1 Aplicaciones básicas

Aunque cada distribución incluye las aplicaciones que considera que cumplen sus objetivos, hay tres que son muy comunes y pueden ayudarte en casi cualquier distribución de escritorio.

4.1.1 Explorador de archivos

Viene a ser algo así como el “Explorador de archivos de Windows” o el “Buscador de macOS”. Puedes ejecutarlo desde Accesorios o buscar en la barra de búsqueda del sistema, pero la forma más común es ejecutarlo desde el ícono que existe en la barra de herramientas. La gran ventaja de Linux es su heterogeneidad, ya que no existe un único explorador de archivos. Dependiendo del escritorio que haya instalado, las opciones pueden variar. Uno de los exploradores de archivos más conocidos para Linux es Nautilus. En “Ubuntu” podemos usar “PCManFM”.

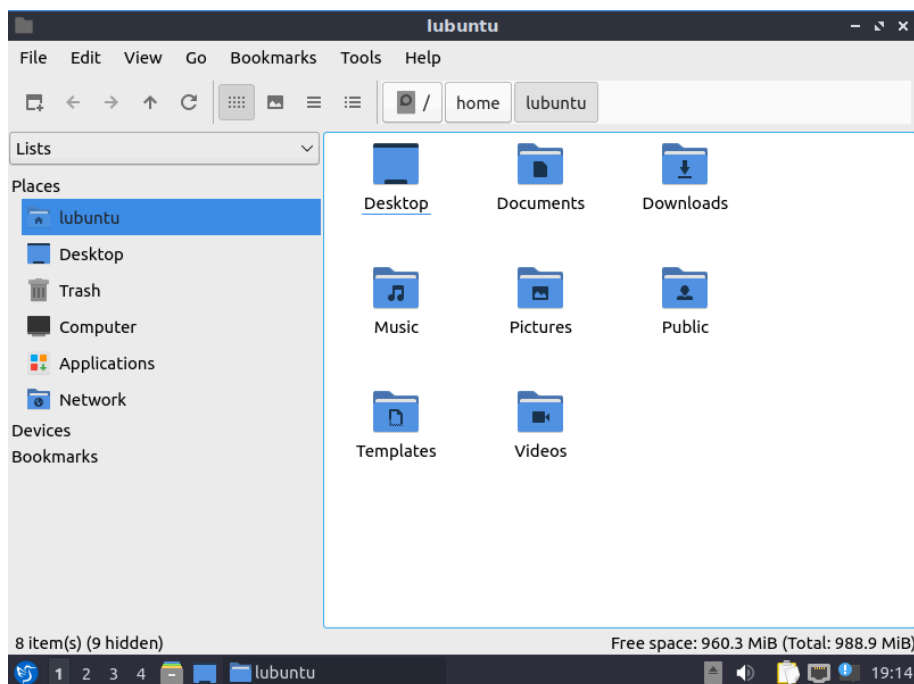


Figura 10. Lubuntu con el explorador de archivos "PCManFM"

4.1.2 Editor de texto

Es similar a herramientas como el "Bloc de notas" del sistema operativo Windows. Como siempre, las opciones son múltiples, pero la más utilizada y que viene de serie en la mayoría de escritorios es "Gedit". En "Lubuntu" podemos usar "FeatherPad". Se puede acceder desde Accesorios.

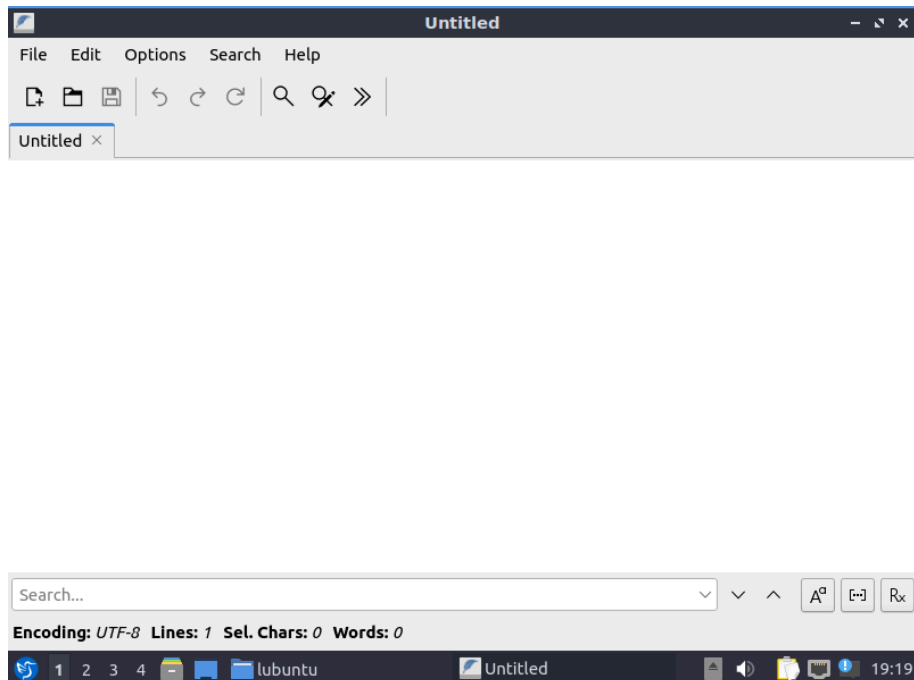


Figura 11. Lubuntu con "FeatherPad" como aplicación de bloc de notas

Interesante: en Linux, para ayudarte a programar o simplemente editar archivos de texto, Visual Studio Code es una gran opción. Puedes instalarlo desde <https://code.visualstudio.com/>.

4.1.3 La terminal

Quizás la app más odiada por los usuarios básicos, pero la mejor amiga de aquellos que quieren sacarle el máximo partido al sistema. A través de él podremos iterar con el sistema operativo en modo texto. A partir de él desarrollaremos todo nuestro trabajo. Podemos acceder a la aplicación de terminal predeterminada en “Lubuntu” accediendo a través de “Herramientas del sistema” y luego “QTerminal”.

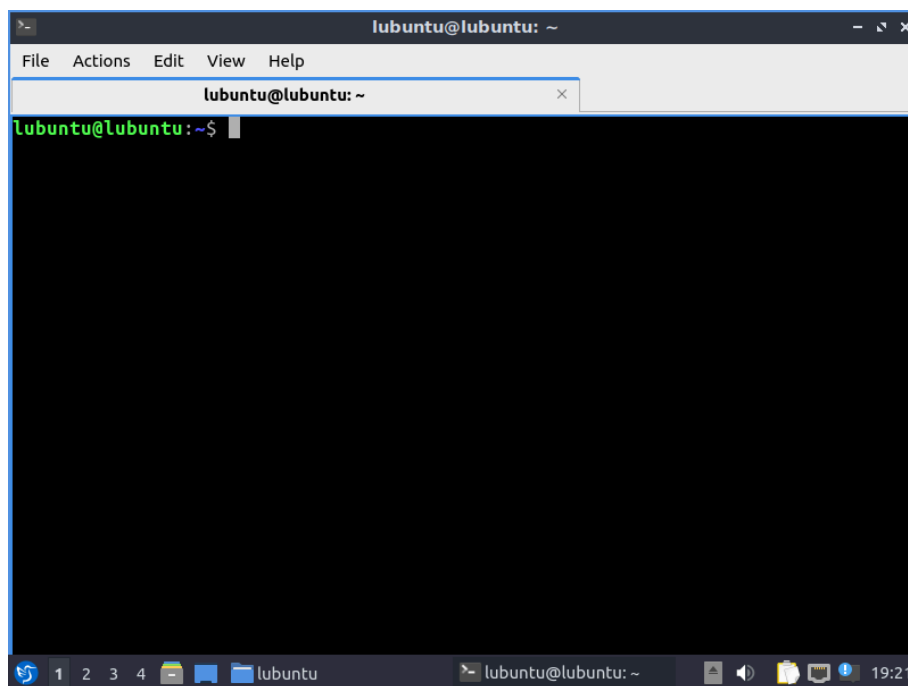


Figura 11. Lubuntu con “QTerminal” como aplicación terminal

4.2 Cómo instalar programas en Linux

4.2.1 Instalación usando paquetes

Hay tres formas fundamentales de instalar programas en Linux:

- De un paquete.
- De un archivo comprimido.
- Del código fuente.

Evidentemente, esta última opción es la más compleja y requiere los conocimientos necesarios para compilar la aplicación, conocimientos que están fuera del alcance de este módulo.

La forma más sencilla es utilizar paquetes, algo similar al “MSI” de Windows. Estos archivos incluyen toda la información para la instalación y la configuración de la aplicación, notificando al sistema qué dependencias son obligatorias para poder ejecutarla correctamente.

Interesante: el problema es que no existe un formato de paquete único. En el mercado existen dos formatos, “.deb” (usado por distribuciones como Ubuntu o Debian) y “.rpm” (usado por distribuciones como OpenSuse o Fedora). Eso hace que los desarrolladores deban generar los dos tipos de paquetes para distribuir sus aplicaciones.

En realidad la instalación de estos paquetes es muy sencilla, simplemente se descargan, se hace clic en ellos y automáticamente se lanza el programa de instalación.

El problema es que los sistemas Linux son sistemas muy abiertos y muy dinámicos. Las versiones de las aplicaciones pasan muy rápido y los lugares desde donde descargar son muy variados. Por eso conviene utilizar nombres de gestores de aplicaciones, una idea que existe en Linux desde hace muchos años y que viene a ser algo así como las tiendas de aplicaciones actuales (App Store, Google Play, etc.).

Por supuesto, cada tipo de paquete está asociado a su propio gestor (trabajando en modo terminal). Hoy en día el formato más común es el de tienda (en modo gráfico), pero posiblemente el más versátil sea el clásico (el antecesor de la tienda).

Por ejemplo, en distribuciones con paquetes “.deb”, el administrador es un comando de consola llamado “**apt**”. Su versión clásica de escritorio es “Synaptic” y el modo de almacenamiento se llama “Centro de aplicaciones”. En caso de conocer el nombre del paquete, el método más sencillo es el primero. Por ejemplo, si queremos instalar el reproductor VLC, la forma más sencilla es abrir la terminal y escribir:

```
sudo apt install vlc
```

Dónde:

- **sudo**: nos permite ejecutar aplicaciones en modo superusuario. Evidentemente, la instalación de programas no es algo que pueda hacer cualquier usuario, por lo que para poder hacerlo es necesario solicitar credenciales de superusuario.
- **apt**: nombre de la aplicación del administrador de paquetes.
- **install**: opción del programa administrador que permite la instalación. Evidentemente, existen otras para eliminar, actualizar, etc.
- **vlc**: nombre del paquete.

Si ejecutamos, el sistema nos solicita la contraseña y procede a la instalación.

! Atención: Linux distingue entre mayúsculas y minúsculas, por lo que no es lo mismo escribir en mayúsculas (por ejemplo “Vlc”) que en minúsculas (por ejemplo “vlc”).

Si no sabemos el nombre exacto del paquete, es mejor utilizar “Muon”. Podemos acceder desde el menú “Herramientas del sistema”. La aplicación dispone de un buscador para localizar el paquete que nos interesa y luego seleccionarlo. Las dependencias necesarias para la instalación se seleccionarán automáticamente.

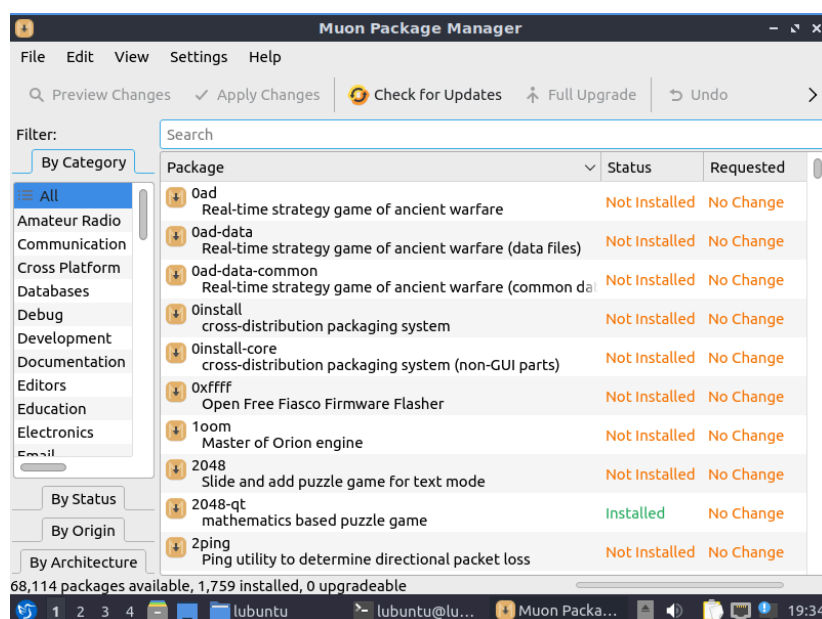


Figura 12. Ubuntu con “Muon” como administrador de paquetes

Estos gestores son configurables para poder soportar cualquier fuente de datos. Por lo general se configura contra los servidores de la distribución que se encarga de mantenerlos lo más actualizados posible. Aun así, no contienen todos los programas ni las últimas versiones, por lo que en muchos casos suele ser conveniente añadir nuevas fuentes que permitan estar lo más actualizados posible.

Puedes administrar esas fuentes desde “Muon” en la opción “Configuración” y luego presionando “Configurar fuentes de software”.

4.2.2 Instalación usando paquetes completos

Hay una cuarta forma: paquetes completos (incluidas en un paquete la aplicación + dependencias) como <https://snapcraft.io> o <https://flatpak.org/>. Son más fáciles de configurar, pero consumen mucho espacio en disco en comparación con los paquetes normales, porque incluyen no solo el programa, sino todas sus dependencias.

Para poder utilizarlos, es necesario instalar un administrador de software especial. Por ejemplo, para instalar “snap” para usar paquetes de <https://snapcraft.io> debes usar este comando:

```
apt install snapd
```

Cuando haya instalado “snapd”, puede instalar el programa desde <https://snapcraft.io> siguiendo las instrucciones. Por ejemplo, si desea instalar “Visual Studio Code”, puede seguir las instrucciones de <https://snapcraft.io/code>. En este caso, las instrucciones dicen que ejecute este comando:

```
sudo snap install code --classic
```

5. USUARIOS EN LINUX

Linux es un sistema operativo multiusuario.

Los usuarios de Linux tienen un nombre asociado, pero internamente se identifican con un número. Este identificador se llama UID. Si dos usuarios tienen nombres diferentes, pero el mismo UID, internamente son el mismo usuario.

Más información en https://en.wikipedia.org/wiki/User_identifier

Básicamente, existen dos tipos de usuarios: usuarios normales y root.

- Un usuario normal es un usuario con UID mayor que 0 y puede realizar operaciones limitadas y solo acceder/modificar los recursos a los que tiene permiso para acceder.
- El usuario raíz es un usuario con UID=0. Es el administrador principal del sistema y prácticamente puede hacer casi todo (cambiar configuración, instalar programas, instalar controladores, ejecutar servidores, leer/eliminar cualquier archivo, etc.).

! Atención: realizar operaciones siendo usuario root es muy peligroso (puedes cometer un error y dañar tu sistema). Si entras en un sistema siendo root tienes que saber muy bien lo que estás haciendo.

5.1 Archivos “/etc/passwd” y “/etc/shadow”

La lista de usuarios se almacena en un archivo llamado “/etc/passwd”. Almacena varios atributos como UID, directorio de inicio, si el usuario está habilitado o no, etc.


Si ejecutamos “cat /etc/passwd” podremos ver su contenido.

Puede encontrar más información sobre el archivo “/etc/passwd” y su formato en

<https://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/>

La contraseña cifrada también se puede almacenar en `"/etc/passwd"`, pero no se recomienda por razones de seguridad (`"/etc/passwd"` puede ser leída por todos).

Por este motivo existe otro archivo para almacenar contraseñas llamado `"/etc/shadow"` que solo el usuario root puede leer y modificar. Puede encontrar más información sobre el archivo `"/etc/shadow"` en <https://www.cyberciti.biz/faq/understanding-etcshadow-file/>

 **Interesante:** En resumen, `"/etc/passwd"` almacena información general de los usuarios y `"/etc/shadow"` almacena contraseñas cifradas.

5.2 Comando "sudo" y lista de sudoers

Hace unas líneas hemos dicho que existen 2 tipos de usuarios: root y usuarios normales. Es una forma ineficiente e insegura de administrar cuentas de administrador.

Por este motivo, las distribuciones modernas de Linux como Ubuntu o Mint:

- De forma predeterminada, la cuenta raíz está desactivada (no puede iniciar sesión como usuario root).
- Hay una lista llamada "sudoers". En esta lista, puede otorgar varios privilegios a usuarios normales.
- El privilegio más común (y útil) es "convertirse en root" temporalmente usando un comando llamado "sudo" antes de la instrucción a ejecutar.


Con esta herramienta y esta configuración, el sistema puede tener más de un administrador (cada usuario que esté en la "lista de sudoers" puede realizar operaciones de root).

Además, es obligatorio utilizar el comando "sudo" antes de ejecutar el comando como root. Aumenta la seguridad porque se supone que si usas "sudo" sabes lo que estás haciendo.

Ejemplo:

Si el usuario pepe (UID=1001) está en "sudoer list" y ejecuta "sudo cat fichero.txt"

Ejecuta el comando "cat fichero.txt" siendo root (UID=0).

 **Interesante:** Cuando ejecuta por primera vez en su sesión un comando sudo (o su último comando sudo fue hace mucho tiempo), el sistema le solicita su propio inicio de sesión por razones de seguridad.

Más información en <https://en.wikipedia.org/wiki/Sudo>

5.3 Comando "su"

El comando "su" es una abreviatura de "Cambiar de usuario".

Este comando se puede llamar:

- **Sin parámetros:** en este caso intenta iniciar sesión como root (UID=0). Funciona incluso si la cuenta raíz está deshabilitada.
- **Con parámetros:** tiene un parámetro que es el nombre de usuario con el que deseas iniciar sesión.
- Si ejecuta el comando siendo root, automáticamente inicia sesión como usuario. Si eres un usuario normal te pide la contraseña de usuario.

Ejemplo:

`"su pepe"`

El sistema intentará iniciar sesión como el usuario "pepe".

`"sudo su"`

El sistema intentará iniciar sesión como root (UID=0).

Más información en [https://en.wikipedia.org/wiki/Su_\(Unix\)](https://en.wikipedia.org/wiki/Su_(Unix))

5.4 Creando usuarios en Linux

En esta página puedes leer información de cómo crear usuarios (por línea de comando) y si lo deseas darles privilegios “sudo”:

<https://www.digitalocean.com/community/tutorials/how-to-add-and-delete-users-on-ubuntu-16-04>


Además, puedes ver un ejemplo con interfaz gráfica en este vídeo.

<https://www.youtube.com/watch?v=DQHS1tQ2Xt8>

Cuando crea un usuario en Linux, el contenido predeterminado de su nuevo directorio de inicio se obtiene del directorio “/etc/skel”. Funciona como una “plantilla”. Más información en <http://linuxg.net/the-unix-and-linux-skeleton-directory-etcskel/>

6. GRUPOS EN LINUX

Linux te permite crear grupos de usuarios. Es útil otorgar permisos o privilegios (como “lista de sudoers”) a un grupo completo (por ejemplo, puede otorgar privilegios “sudo” a un grupo y cada miembro de este grupo podría ejecutar el comando sudo para convertirse en root).

 **Interesante:** normalmente, Ubuntu y distribuciones similares (Lubuntu, Mint, etc.) están configuradas de manera que ser miembro del grupo "sudo" te permite realizar "operaciones sudo".

Un usuario puede ser miembro de varios grupos a la vez.

Al igual que los usuarios, los grupos tienen un nombre, pero internamente se identifican mediante un GID entero. Si dos grupos comparten el mismo GID, internamente son el mismo grupo.

6.1 Fichero “/etc/grupo”

Hay un archivo “/etc/group” donde se enumeran los grupos. Cada línea es un grupo y almacena información como “nombre”, “GID” y el valor más importante: la lista completa de usuarios que son miembros de ese grupo.

Más información sobre “/etc/group” en

<https://www.cyberciti.biz/faq/understanding-etgroup-file/>

6.2 Creando grupos en Linux

En este enlace, puedes ver cómo crear un grupo y agregar un nombre de usuario existente a ese grupo usando la consola:

<http://www.omniseccu.com/gnu-linux/redhat-certified-engineer-rhce/how-to-create-a-new-group-in-linux-using-groupadd-command.php>

Además, puedes ver cómo hacerlo gráficamente en este vídeo.

<https://www.youtube.com/watch?v=ZNeWntArcOg>

7. FICHEROS Y DIRECTORIOS EN LINUX

7.1 Tipos de archivos

En Linux, existen esos tipos de archivos:

- **Ficheros regulares:** contiene información. Son archivos normales, como los que usamos todos los días.

- **Directorios:** son archivos especiales con referencias a otros directorios y archivos.
- **Enlaces:**
 - **Enlaces simbólicos:** es un archivo que contiene la ruta a otro archivo. Es similar a los atajos de Windows. Si elimina el archivo original, el enlace simbólico permanece, pero apunta a un archivo inexistente.
 - **Enlaces duros:** no es un tipo de archivo, es un segundo nombre de un archivo. Si crea un enlace físico de un archivo, para el sistema de archivos son el mismo archivo y no hay forma de saber cuál es el original. Si un archivo tiene más de una referencia, solo se elimina cuando se eliminan todas las referencias.
 - **Archivos especiales:** son archivos que suelen representar dispositivos físicos, como unidades de almacenamiento, impresoras, etc.

7.2 Ficheros ocultos

En Linux, los archivos ocultos son archivos que comienzan con "." como ".bash". Cuando enumera un directorio, no aparece, a menos que use el parámetro "-a". Puedes verlos usando "ls -a".

8. PERMISOS EN LINUX

En Linux, utilizando el comando de línea de comando "ls -l" puede ver información detallada sobre archivos y directorios. Esta información contiene permisos de cada archivo o directorio.

Los principales tipos de permisos en Linux son:

- **Permiso de lectura:**
 - **En un fichero:** puede leer su contenido.
 - **En un directorio:** puede enumerar sus archivos, nombres de directorios y atributos (comando ls).
- **Permiso de escritura:**
 - **En un archivo:** puede modificar el contenido del archivo.
 - **En un directorio:** puede eliminar o crear archivos y directorios en ese directorio.
- **Permiso de ejecución:**
 - **En un fichero:** puede ejecutar el archivo (como Windows ".exe").
 - **En un directorio:** puede ingresar al directorio (comando "cd").

Estos permisos principales deben definirse en 3 grupos: propietario (afecta al propietario del archivo), grupo (afecta a los miembros del grupo) y otros (afecta a otros usuarios).

Un ejemplo del comando "ls -l" aplicado a los permisos:

```

shum@sol:~$ ls -l
total 20
drwx----- 2 shum  staff    4096 Jan 16 22:04 Mail
drwx----- 3 shum  staff    4096 Jan 16 14:15 csc128
drwxr-xr-x  2 shum  staff    4096 Jan 13 16:42 public
drwxr-xr-x  2 shum  staff    4096 Jan 16 14:07 public_html
-rw-r--r--  1 shum  staff     628 Jan 15 20:04 verse
  
```

El diagrama muestra las siguientes anotaciones:

- file type:** Indica el tipo de archivo (d para directorio, - para archivo ordinario).
- number of hard links:** El primer número (2, 3, 2, 2, 1).
- user (owner) name:** El nombre del propietario (shum).
- group name:** El nombre del grupo (staff).
- size:** El tamaño en bytes (4096, 4096, 4096, 4096, 628).
- date/time last modified:** La fecha y hora de la última modificación (Jan 16 22:04, Jan 16 14:15, Jan 13 16:42, Jan 16 14:07, Jan 15 20:04).
- filename:** El nombre del archivo o directorio (Mail, csc128, public, public_html, verse).
- Permissions:** Se detallan los permisos para el propietario (rwx), el grupo (---) y otros (---). Las letras r, w y x representan:
 - r (readable):** legible
 - w (writeable):** escribible
 - x (executable):** ejecutable

8.1 Algoritmo de concesión de permisos

Para determinar si se concede o no un permiso, se sigue el siguiente algoritmo:

1. Primero, verifique si el usuario es root (UID=0). Si es cierto, se concede el permiso.
2. En segundo lugar, compruebe si el usuario es el propietario. Si es el propietario, se aplican "permisos de propietario".
3. En tercer lugar, si el usuario no es root ni propietario, pero es miembro del grupo asociado al archivo, se aplican "permisos de grupo".
4. Por último, si el usuario no es root, ni el propietario ni miembro del grupo, se aplican "otros permisos".

Es posible encontrar contradicciones como que "otros" tienen más permisos que "propietario". Si "otros" pueden escribir y el propietario no, aunque resulte extraño, es una configuración válida.

8.2 Usando el comando "chmod" para establecer permisos

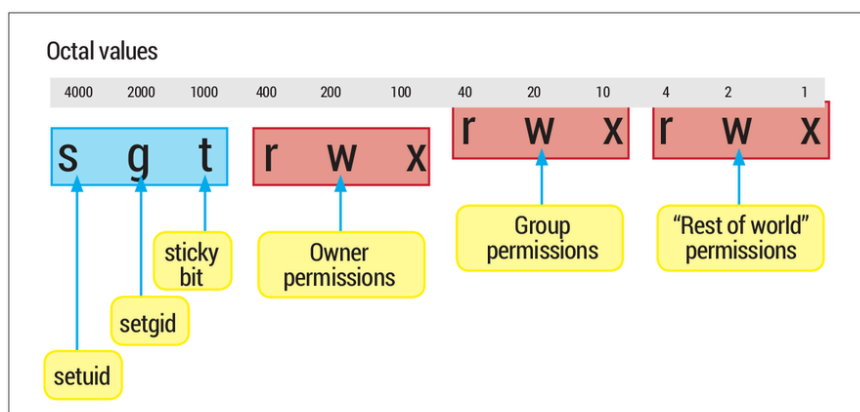
El comando "chmod" se utiliza para establecer permisos. Solo el root y el propietario del recurso pueden cambiar los permisos.

El comando "chmod" tiene principalmente dos notaciones:

- **Notación alfa:**
 - **Ejemplo:** `chmod u=rwx, g=rx, o=- myFile.txt` Pone todos los permisos al propietario, lectura y ejecución al grupo y nada a los demás.
- **Notación octal:**
 - Utiliza un "valor binario" de un valor octal para establecer permisos. Por ejemplo, 5 es 101 en binario y equivale en "rwx" a permisos de lectura y ejecución.
 - **Ejemplo:** `chmod 750 myFile.txt` Pone los mismos permisos que el último ejemplo.
 - Más información al respecto en <http://www.perlfect.com/articles/chmod.shtml>

8.3 Permisos especiales

Hemos hablado de 9 bits de permisos ("rwx" para propietario, "rwx" para grupos y "rwx" para otros). Pero hay 3 bits más: "setUID", "setGID" y "Sticky bit":



- **setUID:** <https://en.wikipedia.org/wiki/Setuid>
 - **En ficheros:** Si el permiso setUID está activado, cuando ejecuta ese archivo, no lo ejecuta con su propio UID, lo ejecuta con el UID del propietario.
 - **En directorios:** si el permiso setUID está activado, si creas un archivo o un directorio, el propietario no eres tú, es el propietario del directorio donde estás.
- **setGID:** <https://en.wikipedia.org/wiki/Setuid>
 - Lo mismo que setUID, pero con ID de grupo en lugar de ID de usuario.
- **Sticky bit:** https://en.wikipedia.org/wiki/Sticky_bit


- Hoy en día se utiliza principalmente en directorios. Si alguien tiene permiso de escritura en un directorio, puede crear archivos y directorios, pero también puede eliminar cualquier archivo o directorio. Si se activa el bit adhesivo en un directorio, cualquier persona con permisos de escritura puede crear archivos y directorios, pero solo puede eliminar archivos y directorios que sean de su propiedad.
 - Las únicas excepciones son el root y el propietario del directorio principal.


Más información sobre esos permisos en

<http://www.unixrock.com/2013/09/how-to-use-setuid-setgid-and-stickybit.html>

9. PRINCIPALES COMANDOS DE LINUX

En esta sección, describiremos los principales comandos de la consola en sistemas Linux. Si deseas obtener información detallada sobre cada uno de ellos, puedes utilizar el comando “man”.

 **Interesante:** “man” es un comando que muestra el manual/ayuda de otros comandos. Es muy útil y está disponible en varios idiomas (inglés, español, etc.).

 **Interesante:** es útil tener una hoja de referencia. Hay muchos de ellos. Por ejemplo, esto es muy interesante. <https://linuxopsys.com/wp-content/uploads/2022/06/linux-cheat-sheet.pdf>

Comando	Que hace	Ejemplo
Comandos para gestionar la interfaz.		
man	Muestra la ayuda de un comando.	<i>man ls</i>
clear	Pantalla clara.	<i>Clear</i>
echo	Mostrar un texto literal en pantalla.	<i>echo "Hello World"</i>
exit	Cierra la sesión en consola.	<i>exit</i>

Comando	Que hace	Ejemplo
Comandos para configurar el sistema.		
date	Establece fecha del sistema.	<i>date #Shows date</i> <i>date -s #Sets date</i>
cal	Muestra el calendario.	<i>cal</i>
shutdown	Apaga el sistema.	<i>shutdown</i>
reboot	Reinicie el sistema.	<i>reboot</i>

Comando	Que hace	Ejemplo
Comandos para obtener información sobre discos.		

du	Muestra el uso del disco para cada archivo.	<i>du -h</i> <i>#Human readable format</i>
df	Muestra información sobre sistemas de archivos.	<i>df -h</i> <i>#Human readable format</i>

Comando	Que hace	Ejemplo
Comandos para gestionar archivos y directorios.		
touch	Crea un archivo vacío.	<i>touch myfile.txt</i>
vi / nano	Crea/edita un archivo de texto.	<i>nano myfile.txt</i> <i>vi myfile.txt</i>
mkdir	Crea un directorio.	<i>make mydir</i>
cat more	Muestra el contenido de un archivo de texto.	<i>cat myfile.txt</i> <i>more myfile.txt</i>
grep	Busca un usuario de texto en un archivo de texto.	<i>grep root /etc/password</i>
ls	Muestra el contenido de un directorio.	<i>ls</i> <i>ls -la</i>
cd	Cambia de directorios.	<i>cd /home #Absolute route</i> <i>cd ../myDir #Relative route</i>
pwd	Muestra la ruta actual.	<i>pwd</i>
rm	"rm" borra ficheros/files. "rm -r" borra directorios y sus contenidos de forma recursiva.	<i>rm myfile</i> <i>rm -r myDirectory</i>
cp	"cp" copia un fichero. "cp -r" copia un directorio de forma recursiva.	<i>cp myFile /home/admin</i> <i>cp -r myDir /home/admin</i>
mv	Mueve/cambia el nombre de un archivo o directorio.	<i>mv myFileOldName /home/myNewName</i>
ln	"ln" crea un enlace duro. "ln -s" crea un enlace simbólico (como los accesos directos de Windows).	<i>ln myFile hardLinkMyFile</i> <i>ln -s myFile shortcutMyFile</i>
mount	Montar un dispositivo en una carpeta.	<i>mount /dev/sda1 /media/myDisk</i>

Comando	Que hace	Ejemplo
Comandos relacionados con permisos		
chmod	Cambia los permisos de un archivo o directorio.	<i>chmod 750 myFile</i>
chown	Cambia el propietario/grupo de un archivo o directorio.	<i>chown newuser:newgroup my file</i>

10. BIBLIOGRAFÍA

[1] Organización de los directorios en Linux

<http://www.linux-es.org/node/112>

[2] Install Lubuntu

<https://www.tecmint.com/install-lubuntu/>

[3] "The Linux command line" Creative Commons book <http://linuxcommand.org/tlcl.php>

[4] "Linux commands Handbook"

<https://bjpcjp.github.io/pdfs/devops/linux-commands-handbook.pdf>