



F 3: Sistemas Operativos Libres

UF 3: SISTEMAS OPERATIVOS LIBRES	142
1. Instalación de sistemas operativos libres	142
1.1. Requisitos técnicos del sistema operativo a instalar	142
1.2. Selección del sistema operativo a instalar	142
1.3. Métodos de instalación y planificación de los parámetros básicos. Particiones. Sistemas de archivos a emplear. Esquemas de partición. Clonación	143
1.4. Instalación de sistemas operativos y configuración de los parámetros básicos	144
1.5. Selección de las aplicaciones básicas a instalar	145
1.6. Creación de escenarios duales con diferentes sistemas operativos libres y propietarios instalados	146
1.7. Gestor de arranque.....	147
1.8. Normas de uso del <i>software</i>	148
1.9. Documentación del proceso de instalación e incidencias. Interpretación de la documentación técnica	148
2. Realización de tareas básicas de configuración y mantenimiento sobre sistemas operativos libres	150
2.1. Arranque y parada del sistema.....	150
2.2. Utilización del sistema operativo	151
2.3. Interfaces de usuario.....	151
2.4. Operaciones con archivos	154
2.5. Compresión y descompresión de ficheros	162
2.6. Actualización del sistema operativo	163
2.7. Agregación, configuración, eliminación y actualización del <i>software</i> del sistema operativo	164
2.8. Configuración de dispositivos periféricos en diferentes sistemas operativos	165
2.9. Inventario del <i>software</i> instalado	166
2.10. Funcionamiento correcto de las configuraciones realizadas.....	167
2.11. Documentación del proceso de configuración. Interpretación de la documentación técnica	168
3. Administración de los sistemas operativos libres	169

UF 3: Sistemas operativos libres

1. Instalación de sistemas operativos libres

1.1. Requisitos técnicos del sistema operativo a instalar

Los **requisitos mínimos** para instalar un sistema operativo libre suelen ser **menores** que los de uno propietario, y más aún si no se instala su entorno gráfico. Cada distribución de Linux tiene sus propios requisitos tanto mínimos como recomendados, los cuales dependen de sus funcionalidades.

Ubuntu Desktop 16.04 LTS es la versión con la que se va a trabajar en este módulo, y tiene los siguientes requisitos mínimos:

- Procesador 1 GHz
- 1,5 GB de memoria RAM
- 7 GB de disco duro

1.2. Selección del sistema operativo a instalar

Una **distribución Linux** es un conjunto de paquetes *software* basados en el núcleo Linux.

Existen muchas distribuciones, por lo que, a la hora de elegir una, es necesario comparar las **características** y escoger la que más se adapta a las necesidades del usuario. Algunas de estas son la estabilidad, el administrador de paquetes y la interfaz gráfica de usuario (GUI).

Trabajaremos con Linux, con la distribución de Ubuntu, que se puede descargar de su página principal:

<https://www.ubuntu.com/download/desktop>

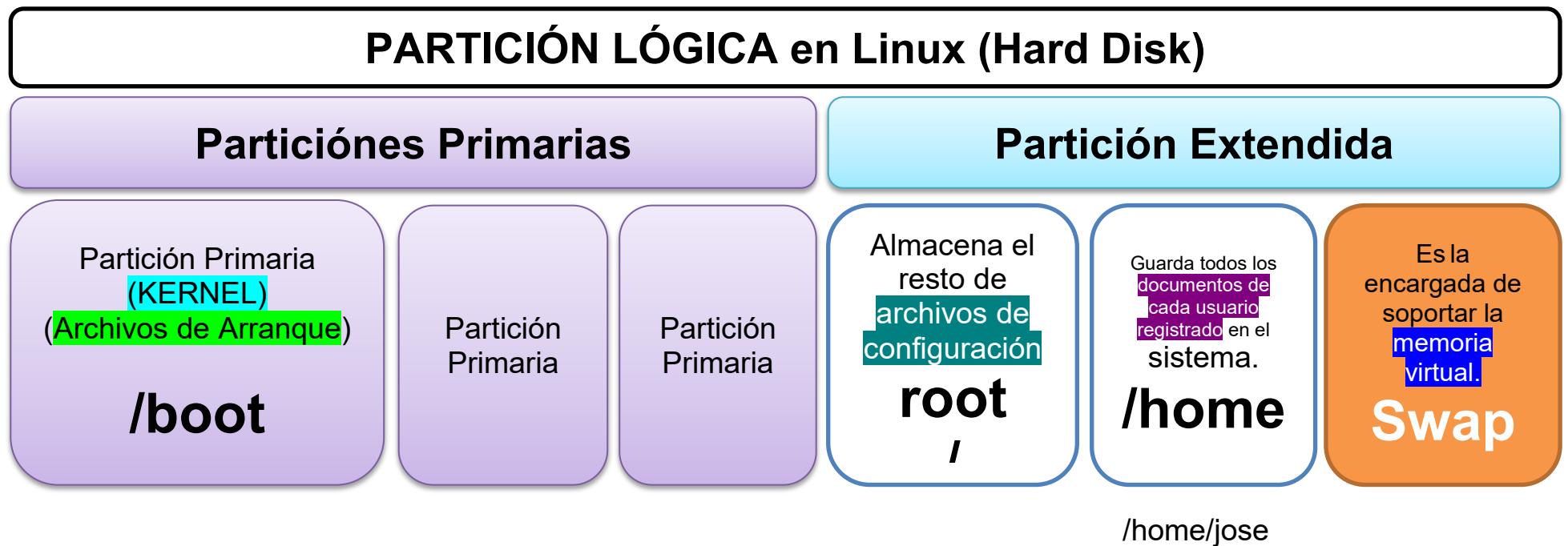
- Se trata de una distribución basada en **Debian**, que mejora la interfaz gráfica y hace que este sistema operativo sea más intuitivo y fácil de usar.
- Su entorno de escritorio es **GNOME**.

1.3. Métodos de instalación y planificación de los parámetros básicos.

Particiones. Sistemas de archivos a emplear. Esquemas de partición.

Clonación

En Ubuntu, es recomendable tener tres **particiones primarias** y una **partición extendida**. La partición primaria es en la que se instala el *kernel* y se almacenan los archivos de arranque. En este caso, se llama */boot*.



Por otro lado, dentro de las **particiones extendidas** se encuentran:

- **Partición raíz**: también denominada partición *root*. Almacena el resto de archivos de configuración. Se representa con */*.
- **Partición home**: guarda todos los documentos de cada usuario registrado en el sistema.
- Partición **swap**: también denominada **de intercambio**. Es la encargada de soportar la memoria virtual. Cuando la memoria RAM está llena, guarda todos aquellos datos que no se están necesitando en ese momento.

Cantidad de RAM en el sistema	Cantidad recomendada de espacio <i>swap</i>
Menos de 4 GB de RAM	Como mínimo 2 GB
Entre 4 GB y 16 GB de RAM	Como mínimo 4 GB
Entre 16 GB y 64 GB	Como mínimo 8 GB
Entre 64 GB y 256 GB	Como mínimo 16 GB
Entre 256 GB y 512 GB	Como mínimo 32 GB

Una vez se ha instalado el sistema operativo, al igual que en Windows, **se puede realizar la clonación del disco**, que consiste en copiar el contenido del disco duro o de la partición a otro.

Tiene las mismas **FUNCIONALIDADES** que en los sistemas operativos propietarios:

Reinicio y restauración: es la más común. Cada vez que se apaga el equipo, se instala el sistema operativo automáticamente, utilizando esta imagen que se ha creado. De esta forma, siempre se tiene un sistema limpio.

- **Equipamiento de nuevas computadoras**: cuando hay que instalar los mismos programas en varios ordenadores, se realiza la instalación en uno de ellos, después se realiza la clonación y, por último, se utiliza esa copia en el resto de equipos.

- **Actualización del disco duro**: si un usuario quiere cambiar su disco duro, puede realizar una clonación del antiguo y copiarla en el nuevo después.

- **Recuperación del sistema**: restaurar una computadora a su configuración de fábrica.

- **Copia de seguridad** de todo el sistema.

Uno de los *softwares*, también libre, que existen para realizar la clonación de discos es **Clonezilla**.

1.4. Instalación de sistemas operativos y configuración de los parámetros básicos

La instalación de un sistema operativo se compone de cinco fases:



Uno de los **fallos** que pueden aparecer es **no disponer de los dispositivos *hardware*** necesarios para ello. Por ejemplo, el dispositivo de almacenamiento que contiene el fichero de instalación del sistema operativo o una memoria USB con la ISO.

- **Planificación:** es conveniente pensar cómo se llevará a cabo y cuáles son las instrucciones por seguir. Si no se realiza este paso inicial, es posible que falle el proceso por no prever qué errores podemos tener. Por otra parte, es necesario conocer el estado actual del sistema para decidir qué acción se va a llevar a cabo. Es posible actualizar el sistema operativo o formatear, es decir, instalar todo desde cero.

- **Preparación:** se debe configurar el ordenador para recibir esta instalación. Para ello, hay que entrar en la BIOS. Dependiendo de la placa base, hay que pulsar una tecla u otra. Una vez dentro, hay que configurar la opción de arranque e indicar el dispositivo que vamos a usar.

- **Instalación:** hay que ir indicando algunas de las configuraciones, como puede ser el tipo de instalación o la partición que se va a utilizar.

- **Configuración:** una vez que el sistema operativo se ha instalado, hay que aceptar la licencia y configurar parámetros como la zona horaria, el idioma, el nombre del equipo, el usuario administrador y la administración red, entre otros.

- **Documentación:** es necesario documentar toda la información del proceso para que quede constancia de qué se ha hecho y así llevar un mejor mantenimiento del mismo.

Al final del capítulo se mostrará como documentar una instalación de un sistema operativo.

1.5. Selección de las aplicaciones básicas a instalar

Una vez se ha instalado el sistema operativo y configurado sus parámetros, es necesario dejar el ordenador listo para que el cliente lo pueda utilizar. Al inicio del proceso, en la elección del sistema operativo, se ha hablado con él para conocer el uso que va a hacer del ordenador, por lo que se recomienda instalar aquellas aplicaciones que se vean convenientes.

El primer paso es la **instalación de los drivers necesarios**, como pueden ser los de la impresora. Además, uno de los *softwares* más importantes que se deben instalar es un antivirus. Es posible instalar diferentes *softwares* conocidos que sean multiplataforma, pero **GNU/Linux** tiene un *software* multiplataforma y *open source*, que es **ClamAV**.

1.6. Creación de escenarios duales con diferentes sistemas operativos libres y propietarios instalados

Cada vez es más habitual que haya gente interesada en tener más sistemas operativos de los que vienen instalados en el ordenador (que suelen ser Windows o Mac OS).

Al igual que se explicó en la parte anterior, donde se hablaba de los sistemas operativos propietarios, **es posible tener Linux como partición primaria y, usando *Dual-Boot*, obtener un arranque dual**. Esto es posible gracias al **gestor de arranque** (del cual veremos sus tipos y definición más adelante).

Una vez se inicia la máquina, la **BIOS** se encarga de cargar el gestor de arranque, que inicia el sistema operativo instalado. En este caso, encontrará dos y mostrará una pantalla previa al arranque del mismo permitiendo elegir cuál de ellos queremos iniciar.

Tenemos la posibilidad de **instalar tantos sistemas operativos** como unidades lógicas nos permita tener el disco duro.

En este caso, solo se podrá utilizar un sistema operativo, pues no es posible trabajar simultáneamente con los dos. Si se quiere cambiar, es necesario reiniciar la máquina y elegir el deseado en la pantalla de arranque dual.

Para más **información**, puedes consultar el siguiente **manual**:

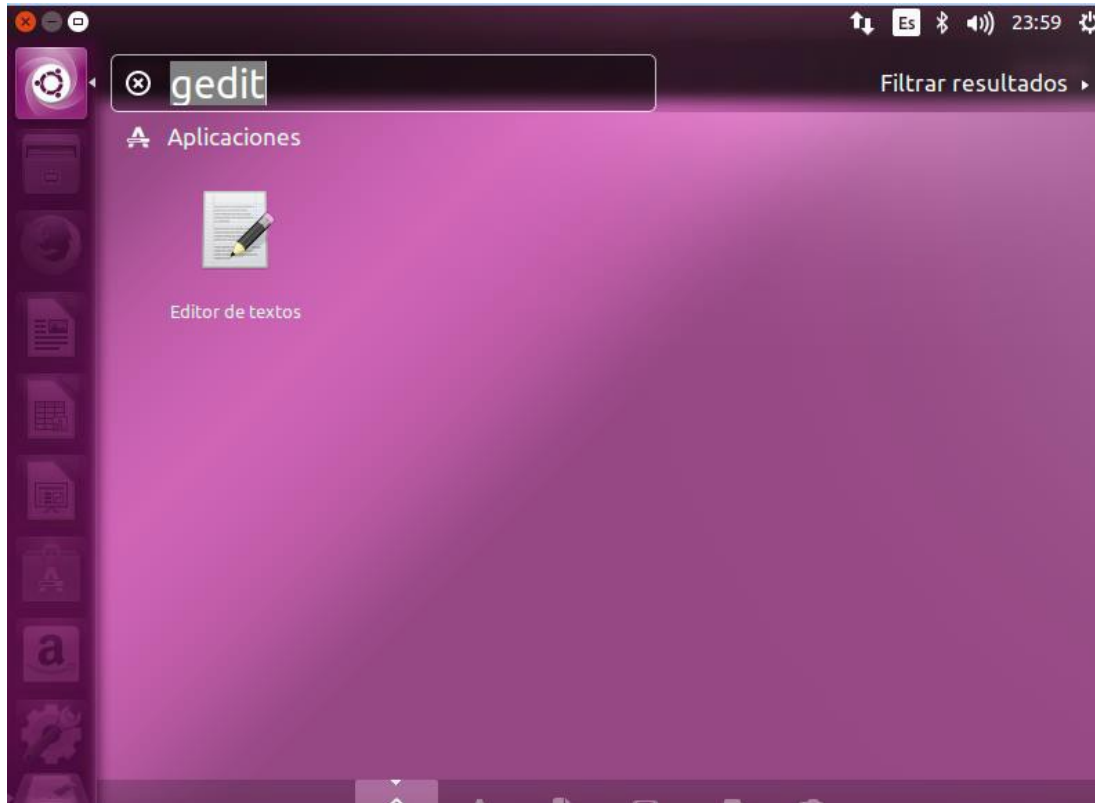
http://www.ite.educacion.es/formacion/materiales/130/cd/redesubuntu/ubuntu-SaberMas/arranque_dual_windowslinux.html

Existen varios **GESTORES DE ARRANQUE**, pero los más conocidos son:

- **GRUB**: es un gestor de arranque múltiple, por lo que se recomienda usarlo si se quieren tener dos sistemas operativos en una misma máquina sin usar una máquina virtual, es decir, cuando se desea un espacio dual en la máquina. De esta forma, permite la elección del sistema operativo. Es el que se suele usar en los sistemas GNU/Linux.
- **LILO**: aunque también permite el arranque dual, se utiliza menos que GRUB.

➤ CONFIGURACIÓN

Para establecer la configuración del gestor de arranque, hay que abrir el fichero `/etc/default/grub` con un editor de texto plano, como puede ser **gedit**



1.8. Normas de uso del software

El **software libre** se refiere a aquellos programas que pueden ser copiados, modificados y utilizados sin coste alguno, con cualquier fin y

No se debe confundir el concepto de **software libre** con **software gratuito** o **freeware**. Este último es aquel que no tiene ningún coste para el usuario, pero que no puede ser copiado, modificado o redistribuido.

Se considera **software libre** cuando cumple las siguientes **características**:

- Libertad para usarlo con cualquier **propósito**.
- Libertad para **estudiar** cómo funciona y **modificarlo** para adaptarlo a unas necesidades concretas.
- Libertad para **distribuir copias**.
- Libertad para **mejorarlo y publicar estas mejoras** para los demás.

1.9. Documentación del proceso de instalación e incidencias. Interpretación de la documentación técnica

CUADERNO DE BITÁCORA

Es necesario elaborar unos informes de incidencias siempre que se trabaja en el **mantenimiento** de un ordenador. A este tipo de documentación se le llama **cuaderno de bitácora**, pues debe tener toda la información agrupada para cada diferente equipo.

Para confeccionarlo hay que diferenciar la parte de *hardware* de la parte *software*.

HARDWARE DE LA MÁQUINA REAL

es necesario especificar las características principales:

- Referencia de equipo
- Marca y modelo
- Procesador
- Memoria RAM
- Discos de almacenaje
- Tarjeta gráfica
- Tarjeta de red

Por otro lado, del

PROCESO DE LA INSTALACIÓN DEL SISTEMA OPERATIVO PROPIETARIO,

se debe indicar:

• Fecha
• Tipo de instalación
• Nombre
• Versión
• Arquitectura
• Usuario administrador
• Contraseña
• Licencias instaladas
• Observaciones

Es importante dejar **constancia de todos los errores** que han ocurrido durante la instalación y de su solución. De esta forma, si el ordenador tiene algún problema y necesita una reparación, se podrá consultar

Además de esta documentación, también hay que indicar todo el **software que se ha instalado** durante el **proceso de la puesta a punto del equipo**, como, por ejemplo, el antivirus escogido.

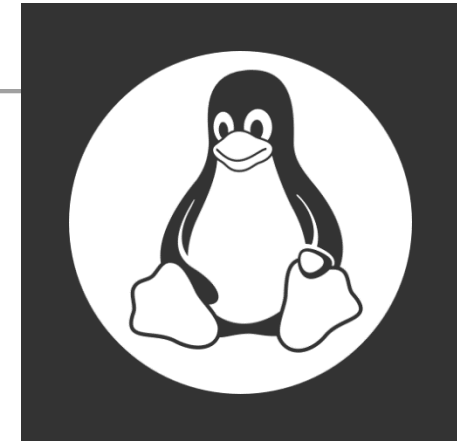
2. Realización de tareas básicas de configuración y mantenimiento sobre sistemas operativos libres

2.1. Arranque y parada del sistema

El sistema operativo Linux permite tanto arrancar como detener el sistema, aunque **no es una tarea sencilla**. La persona encargada de ello ha de comprender bien este proceso.

Para realizar dichas tareas, es necesario indicar al sistema en qué estado y en qué niveles de ejecución deberá configurarse.

Existen 7 niveles de ejecución numerados del 0 al 6, y cada uno de ellos define qué procesos ejecutar, el modo y los dispositivos del sistema operativo.



0	Detiene el sistema
1	Modo monousuario (Administración)
2	Multiusuario local con funciones de red, pero sin servicios
3	Multiusuario completo con red
4	No se utiliza
5	Multiusuario completo con red e interfaz gráfica
6	Reinicio

➤ PROCESO DE ARRANQUE DEL SISTEMA

1	PREVIAMENTE, EL SISTEMA HA DE ESTAR EN ESTADO DE PARADA , ES DECIR, EN EL NIVEL 0.
2	CUANDO SE EJECUTE EL ARRANQUE , SE CARGARÁ EN MEMORIA UNA PEQUEÑA PARTE DEL CÓDIGO, QUE DARÁ INICIO AL SISTEMA OPERATIVO.
3	UNA VEZ INICIADO ESTE, SE REALIZAN LAS DISTINTAS COMPROBACIONES DE HARDWARE Y EL MONTAJE DE TODOS LOS FICHEROS DEL SISTEMA Y DISPOSITIVOS.
4	POR ÚLTIMO, SE LANZA EL PROCESO INIT , QUE SERÁ EL PRIMERO QUE SE EJECUTE DURANTE EL ARRANQUE.
5	SUS TAREAS SON COMPROBAR DENTRO DEL FICHERO /ETC/INITTAB CUÁL ES EL NIVEL DE EJECUCIÓN A EJECUTAR.
6	O REALIZAR EL NIVEL POR DEFECTO (EL NIVEL DE EJECUCIÓN 5).

➤ PARADA DEL SISTEMA

1	ESTE PROCESO SE ENCARGA DE FINALIZAR LOS SERVICIOS Y LOS PROGRAMAS QUE ESTÁN EN EJECUCIÓN.
2	DESMONTA TODAS LAS UNIDADES Y DISPOSITIVOS ACTIVOS Y EL SISTEMA DE ARCHIVOS DEL SISTEMA.
	ES NECESARIO EJECUTARLO SIEMPRE QUE SE DESEE CAMBIAR DE NIVEL O CUANDO EXISTA UNA EJECUCIÓN DE PROCESOS QUE SOBRECARGUEN EL SISTEMA.
	ADEMÁS, PERMITE AL ADMINISTRADOR TENER EL CONTROL DEL MISMO.

➤ SESIONES

La utilización de los recursos, permisos e instalación de aplicaciones son dependientes del tipo de usuario. Linux cuenta con un **sistema de gestión de cuentas** por medio de **sesiones**. Cada una de ellas se corresponde con un escritorio y una configuración que permite a ese usuario realizar unas determinadas tareas. De esta forma, todos pueden acceder al mismo equipo y utilizar cada uno su propio entorno de trabajo.

2.2. Utilización del sistema operativo

Cualquier sistema operativo se puede utilizar tanto en **modo gráfico** (con una interfaz intuitiva) como en **modo consola** (con comandos). No obstante, en Linux se suele trabajar sin entorno gráfico por razones de seguridad.

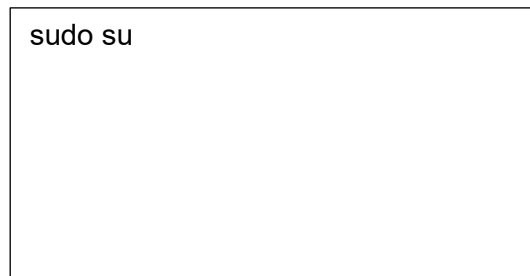
➤ Modo **GRÁFICO**



La principal diferencia reside en la **visualización de los elementos**. Se distribuye en ventanas y la interacción se realiza con un ratón (o directamente con las manos, como en los dispositivos táctiles).

Ubuntu utiliza el entorno de escritorio GNOME, el cual es necesario instalar en algunas ocasiones, además del sistema operativo.

➤ Modo **CONSOLA**



El modo consola consume muchos menos recursos (por ejemplo, memoria RAM) que el modo gráfico, lo que se considera una gran ventaja. Como hemos visto anteriormente, hay opciones de arranque que no muestran la interfaz gráfica.

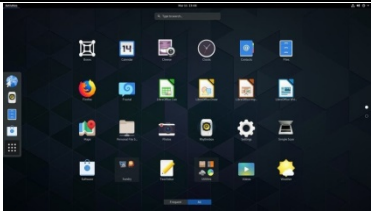
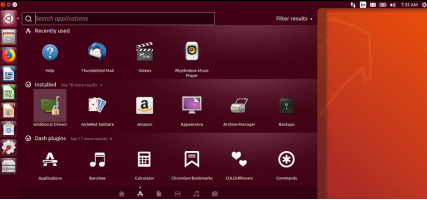



2.3. Interfaces de usuario

► Tipos, propietarios y usos

Hay una **gran variedad de entornos gráficos** en Linux, lo que permite que los usuarios menos expertos en el uso de comandos de gestión en una consola puedan hacer uso de un sistema operativo libre más fácilmente. La mayor parte de ellos están basados en lo que se conoce como **Sistema de Ventanas x**, que ofrece la gestión y el uso del sistema operativo por medio de ventanas.

Los diferentes tipos de entornos de escritorio de *software* libre que nos podemos encontrar son GNOME, KDE, Unity, XFCE y LXDE.

INTERFACES DE USUARIO en LINUX

				
				
GNOME	KDE	Unity	XFCE	LXDE

Configuración de las preferencias de escritorio

Linux permite personalizar cualquier parte del sistema. Una de ellas es el escritorio, puesto que existen distintos tipos:

- **GNOME**: permite un acceso más intuitivo y rápido a las diferentes aplicaciones. Además, incluye la personalización de apariencias, resoluciones, temas, etcétera.

- **KDE**: es un tipo de comunidad que desarrolla entornos de escritorio y que busca que el usuario sea capaz de modificar todos los componentes (menús, diálogos de configuración, botones en las ventanas) y definir su apariencia.

- **Unity**: desarrollada para la distribución de Ubuntu. Se ejecuta por encima de GNOME y ofrece una escalabilidad del sistema para dispositivos móviles.

- **XFCE**: es, básicamente, un gestor de ventanas que permite modificar aspectos como la resolución y la rotación de la pantalla y modificar parámetros de configuración de la misma (como la velocidad de refresco, entre otras).

- **LXDE**: además de las mejoras en la apariencia, destaca por su bajo consumo de energía y recursos.

Accesibilidad para las personas discapacitadas

Hay multitud de herramientas que se pueden utilizar por aquellas personas que tienen algún tipo de discapacidad.

Dentro del entorno gráfico GNOME, uno de los más utilizados, existen las siguientes:

- **Orca**: es una herramienta que permite a personas con discapacidades visuales aumentar y mejorar la visibilidad de los textos. También es posible mediante comandos de voz para el braille la lectura de texto hablado.

- **Control de la velocidad del ratón y uso de repetición de teclas**: de la misma forma, se puede encontrar para el teclado para personas con problemas de movilidad.

- **Otras herramientas** integradas dentro de la apariencia del sistema permiten modificar el contraste o el brillo de la pantalla.

- También tiene integrados todos los **comandos de voz** de Linux para gestionar órdenes en consola.

Dentro de cada una de las distribuciones, es posible acceder al menú de **Accesibilidad**, donde se podrán configurar las tareas para cada una de las necesidades de las personas.

2.4. Operaciones con archivos

Nombres y extensiones

➤ Sistemas de archivos

Archivo:

Nombre Archivo.(Extensión) Tamaño

Todos los archivos tienen un **nombre** (que es su identificador) y una **extensión**.

Ambos pueden cambiarse, aunque es poco recomendable hacerlo con el segundo.

No obstante, se puede cambiar un archivo con extensión **.txt** (un archivo de texto)

a un ejecutable de Windows (con extensión **.bat**).

Para realizar esto mediante comandos, hay que utilizar el comando **rename**.

A la hora de definir los filtros para realizar una búsqueda, podemos utilizar caracteres comodines:

- **Símbolo “?”**: sirve para indicar que puede ser cualquier caracter, pero es un único caracter.
 - **Símbolo “*”**: sirve para indicar que pueden ser varios caracteres.
-

Operaciones más comunes

Las operaciones más usuales entre los usuarios son aquellas que se refieren al trabajo con archivos y con directorios.

OPERACIONES CON DIRECTORIOS

Operaciones con directorios

Crear un directorio:	<i>mkdir</i>	(nombre de directorio)
Cambiar de directorio:	<i>cd</i>	(nombre de directorio)
		(acompañado de ruta que puede ser relativa o ruta)
Listar directorio:	<i>ls -l</i>	(para ver en lista)
	<i>ls -a</i>	(para ver en oculto)
Copiar directorio:	<i>cp</i>	(nombre de directorio)
Mover directorio:	<i>mv</i>	(nombre de directorio)
Eliminar directorio:	<i>rm</i>	(nombre de directorio)

- **Crear un directorio**: para crear un directorio se utiliza el comando **mkdir**. Si el parámetro es solo un nombre, crea un directorio con dicho nombre en el directorio actual. También es posible introducir una ruta como parámetro.

```
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$ mkdir SMIR
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$ ls -l
total 4
drwxrwxr-x 2 ilernaonline ilernaonline 4096 jul 12 00:00 SMIR
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$
```

- **Cambiar de directorio**: para que un usuario pueda moverse por los directorios del sistema de archivos debe utilizar el comando **cd**, el cual tiene que ir acompañado de la ruta a la que quiere ir, que puede ser tanto relativa como absoluta.

```
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$ cd SMIR
ilernaonline@ilernaonline-virtual-machine:~/Escritorio/SMIR$
```

- **Listar directorio**: para conocer el contenido de un directorio se debe introducir el comando **ls**; si se quiere ver en modo lista, junto con el parámetro **-l**; pero si se quieren ver también los archivos ocultos, se introducirá el parámetro **-a**.

```
ilernaonline@ilernaonline-virtual-machine:~/Escritorio/SMIR$ ls -l
total 0
-rw-rw-r-- 1 ilernaonline ilernaonline 0 jul 12 00:02 PAC1.txt
```

Como se puede observar, existen diferentes bloques de información: permisos, número de i-nodos, usuario propietario, grupo propietario, fecha y hora de creación y nombre del fichero o directorio.

Un **i-nodo** es un bloque que almacena información de un fichero.

- **Copiar directorio**: se utiliza el comando *cp*.

```
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$ cp -r PACs SMIR
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$ cd SMIR
ilernaonline@ilernaonline-virtual-machine:~/Escritorio/SMIR$ ls -l
total 4
drwxrwxr-x 2 ilernaonline ilernaonline 4096 jul 12 00:18 PACs
ilernaonline@ilernaonline-virtual-machine:~/Escritorio/SMIR$
```

- **Mover directorio**: se utiliza el comando *mv*. Este mueve un directorio junto con todos los archivos y directorios que contiene.

```
ilernaonline@ilernaonline-virtual-machine:~/Escritorio/SMIR$ mv PACs UF1
ilernaonline@ilernaonline-virtual-machine:~/Escritorio/SMIR$ cd UF1
ilernaonline@ilernaonline-virtual-machine:~/Escritorio/SMIR/UF1$ ls -l
total 4
drwxrwxr-x 2 ilernaonline ilernaonline 4096 jul 12 00:18 PACs
ilernaonline@ilernaonline-virtual-machine:~/Escritorio/SMIR/UF1$
```

- **Eliminar directorio**: se utiliza el comando *rm*.

```
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$ rm -r SMIR
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$ ls -l
total 0
```

OPERACIONES CON FICHEROS

Operaciones con FICHEROS

Crear un fichero:

- se utiliza el comando

touch. (nombre de directorio)

Si el parámetro es solo un nombre, creará un directorio con dicho nombre en el directorio actual. También es posible introducir una ruta como parámetro.

Abrir un fichero:

cat o ***more*** (nombre de directorio)

Copiar un fichero:

cp (nombre de directorio)

Mover un fichero:

mv (nombre de directorio)

Eliminar un fichero:

rm (nombre de directorio)

- **Crear un fichero:** se utiliza el comando ***touch***. Si el parámetro es solo un nombre, creará un directorio con dicho nombre en el directorio actual. También es posible introducir una ruta como parámetro.

```
ilernaonline@ilernaonline-virtual-machine:~/Escritorio/SMIR$ touch PAC1.txt
ilernaonline@ilernaonline-virtual-machine:~/Escritorio/SMIR$ ls -l
total 0
-rw-rw-r-- 1 ilernaonline ilernaonline 0 jul 12 00:02 PAC1.txt
ilernaonline@ilernaonline-virtual-machine:~/Escritorio/SMIR$
```

-
- **Abrir un fichero:** es posible ver el contenido de un fichero de texto en el *Símbolo de sistema* mediante una línea de comando. Para ello, se utiliza el comando ***cat*** o ***more***.

```
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$ more documento.txt
Hola, buenos días a todos. Estamos estudiando la UF3.
```

qq

```
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$ cat documento.txt
Hola, buenos días a todos. Estamos estudiando la UF3.
```

-
- **Copiar un fichero:** se usa el comando ***cp***.

```
ilernaonline@ilernaonline-virtual-machine:~/Escritorio/SMIR$ cp PAC1.txt ..
ilernaonline@ilernaonline-virtual-machine:~/Escritorio/SMIR$ cd ..
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$ ls -l
total 4
-rw-rw-r-- 1 ilernaonline ilernaonline 0 jul 12 00:07 PAC1.txt
drwxrwxr-x 2 ilernaonline ilernaonline 4096 jul 12 00:02 SMIR
```


-
- **Mover un fichero:** se utiliza el mismo comando que para mover un directorio, *mv*.

```
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$ mv PAC1.txt ..
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$ cd ..
ilernaonline@ilernaonline-virtual-machine:~$ ls -l
total 44
drwxr-xr-x 2 ilernaonline ilernaonline 4096 dic 12 2016 Descargas
drwxr-xr-x 2 ilernaonline ilernaonline 4096 dic 12 2016 Documentos
drwxr-xr-x 3 ilernaonline ilernaonline 4096 jul 12 00:08 Escritorio
-rw-r--r-- 1 ilernaonline ilernaonline 8980 dic 12 2016 examples.desktop
drwxr-xr-x 2 ilernaonline ilernaonline 4096 dic 12 2016 Imágenes
drwxr-xr-x 2 ilernaonline ilernaonline 4096 dic 12 2016 Música
-rw-rw-r-- 1 ilernaonline ilernaonline 0 jul 12 00:07 PAC1.txt
drwxr-xr-x 2 ilernaonline ilernaonline 4096 dic 12 2016 Plantillas
drwxr-xr-x 2 ilernaonline ilernaonline 4096 dic 12 2016 Público
drwxr-xr-x 2 ilernaonline ilernaonline 4096 dic 12 2016 Vídeos
```

-
- **Eliminar un fichero:** se utiliza el comando *rm*.

```
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$ rm documento.txt
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$ ls -l
total 0
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$
```


PERMISOS Y ATRIBUTOS DE LOS ARCHIVOS Y DIRECTORIOS

Todos los archivos y directorios tienen una serie de **permisos** para cada uno de los usuarios del sistema. Estos se pueden establecer tanto para el propietario, como para el grupo al que pertenece y el resto de usuarios.

Los **permisos** que tienen tanto un fichero como un directorio son:

R -> lectura

W -> escritura

X -> ejecución

Comando chmod: ¿Qué es? y ¿Cómo usarlo?



Ayuda
Linux

Para cambiar los **permisos** se utiliza el comando **chmod**, seguido de los **permisos** para los diferentes tipos de usuarios y, después, el **nombre del archivo o directorio**.

Anteriormente, se ha expuesto la información que se muestra al utilizar el comando **ls -l** y, como se ha visto, el primer bloque de la misma corresponde a los permisos de dicho archivo y directorio.

```
ilernaonline@ilernaonline-virtual-machine:~/Escritorio/SMIR/UF1/PACs$ ls -l
total 0
-rw-rw-r-- 1 ilernaonline ilernaonline 0 jul 12 00:22 PAC1.txt
-rw-rw-r-- 1 ilernaonline ilernaonline 0 jul 12 00:22 PAC2.txt
-rw-rw-r-- 1 ilernaonline ilernaonline 0 jul 12 00:23 PAC3.txt
-rw-rw-r-- 1 ilernaonline ilernaonline 0 jul 12 00:23 PAC4.txt
-rw-rw-r-- 1 ilernaonline ilernaonline 0 jul 12 00:23 PAC5.txt
ilernaonline@ilernaonline-virtual-machine:~/Escritorio/SMIR/UF1/PACs$
```

En este bloque existen **diez dígitos**, que pueden tener

valor “**—**” o una letra asociada (**r**, **w** o **x**), _____ es decir, el tipo de **permiso**.

Pero también hay una letra **d**, _____ la cual indica qué es un directorio.

Los siguientes nueve dígitos se dividen en

3 (bloques) de 3 (elementos).

1er Grupo) **Muestra los permisos para el usuario propietario del archivo;**

2er Grupo) **para el grupo al que pertenece el propietario;**

Último Grupo) **muestra los permisos del resto de usuarios.**

Si un permiso es “—”, significa que ese grupo de usuarios no tiene permisos sobre el fichero.

Para cambiar los permisos de los archivos, usamos el comando **chmod**.
Existen dos formas de utilizarlo:

- **Modo octal**: se utiliza el comando **chmod** con los permisos traducidos a octal. Los grupos de tres elementos se convierten en una cifra octal y se tienen en cuenta los siguientes valores:

R	W	X
4	2	1

Debido a esto, se pueden traducir los permisos de un archivo a valor octal, con un 1 si aparece la letra y con un 0 si no aparece.

r	w	x	Nº BINARIO	Nº OCTAL
0	0	0	000	0
0	0	1	001	1
0	1	0	010	2
0	1	1	011	3
1	0	0	100	4
1	0	1	101	5
1	1	0	110	6
1	1	1	111	7

- **Modo carácter:** se indican los permisos para cada grupo de usuarios mediante letras.
 - Sintaxis del comando:

chmod grupo operación permisos.

Tipo y sintaxis de grupo:

- Usuario propietario: *u*
- Grupo del usuario propietario: *g*
- Resto de usuarios: *o*

Operaciones:

- Añadir: *+*
- Eliminar: *-*
- Sobreescribir al modo anterior: *=*

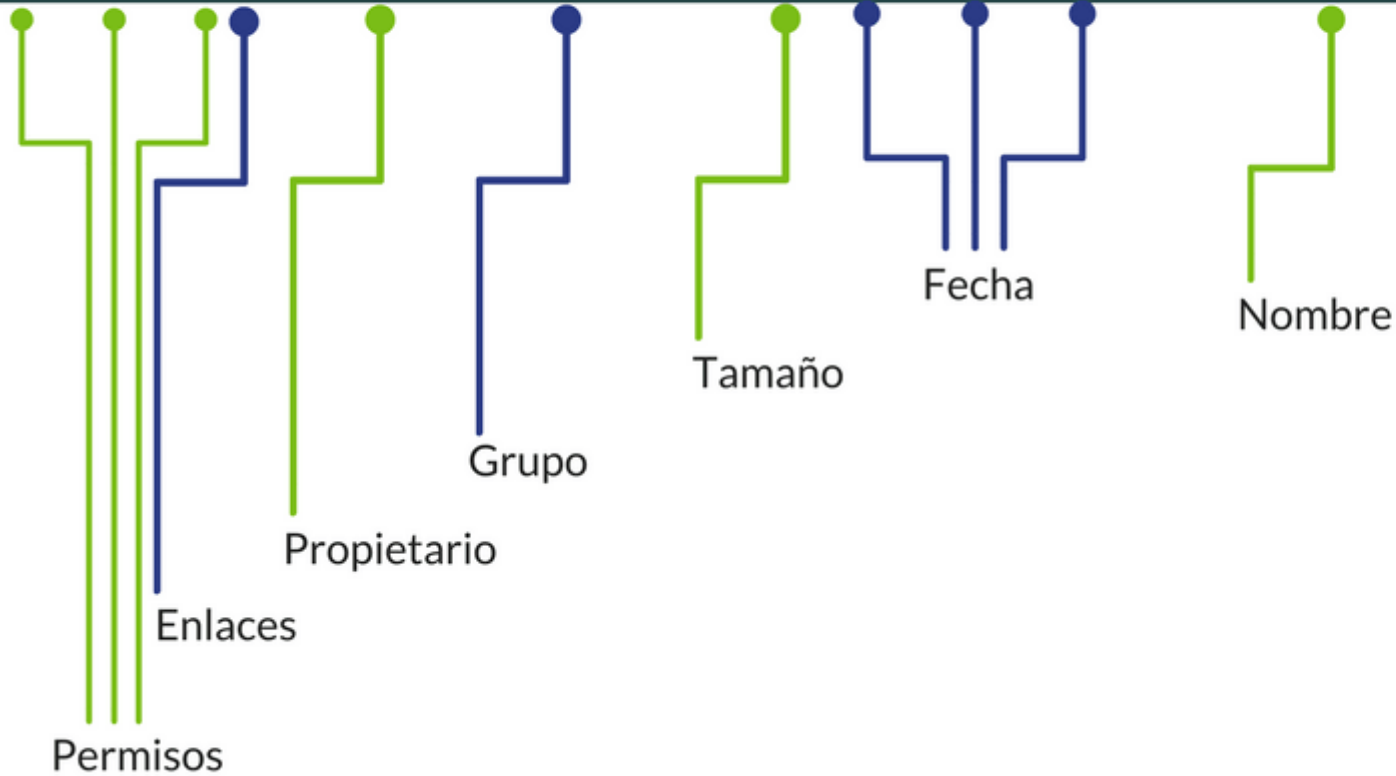
Uso del comando "CHMOD"

SoloLinux



u	g	o	+ -	r	w	x
(user) dueño del fichero	(group) usuarios que pertenecen al mismo grupo	(others) el resto de usuarios	dar permiso quitar permiso	(read) lectura	(write) escritura	(execution) ejecución

```
-rw-r--r-- 1 gospelidea gospelidea 92 Oct 30 09:36 html5-player.php  
-rw-r--r-- 1 gospelidea gospelidea 6472 Sep 20 2017 icecast.php  
-rw-r--r-- 1 gospelidea gospelidea 14934 Sep 20 2017 reproductores-aac-gratis.php
```

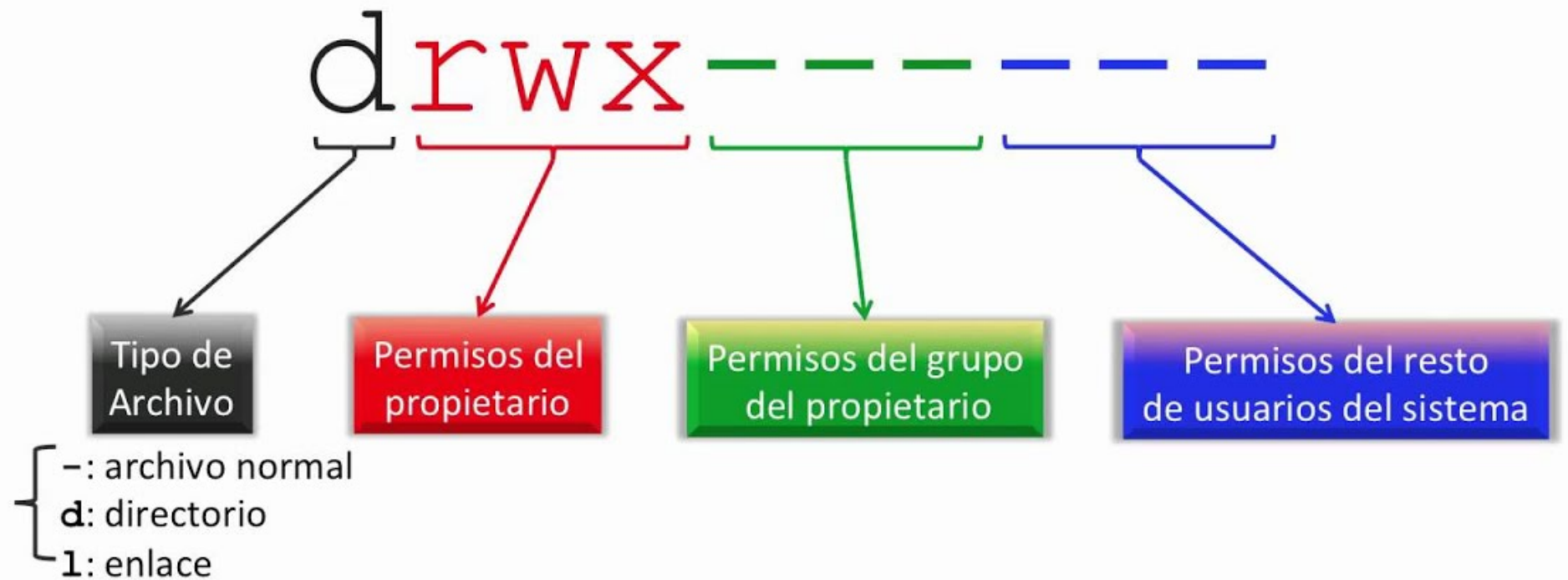


drwxr-xr-x	3	raul	raul	4096	2005-02-16	14:47	Desktop
drwxr-xr-x	5	raul	raul	4096	2005-02-16	14:47	GNUstep
-rw-r--r--	1	raul	raul	246417	2005-03-03	14:47	foto1.png
-rw-r--r--	1	raul	raul	232505	2005-03-03	16:26	carta2.abw
-rw-r--r--	1	raul	raul	239618	2005-03-03	18:15	informe.abw
drwxr-xr-x	2	raul	raul	4096	2005-02-16	14:47	tmp

Annotations:

- Tipo de Archivo
- PERMISOS DE PROPIETARIO
- PERMISOS DE GRUPO
- PERMISOS DE OTROS
- Enlaces
- Propietario
- Grupo
- Tamaño
- Fecha
- Hora
- Nombre

Gestión de Permisos



Octal	Decimal	Permission	Representation
000	0 (0+0+0)	No Permission	---
001	1 (0+0+1)	Execute	--x
010	2 (0+2+0)	Write	-w-
011	3 (0+2+1)	Write + Execute	-wx
100	4 (4+0+0)	Read	r--
101	5 (4+0+1)	Read + Execute	r-x
110	6 (4+2+0)	Read + Write	rw-
111	7 (4+2+1)	Read + Write + Execute	rwx

Tipo	Valor	=	Lectura	Escritura	Ejecución
Ningún permiso	0	=	0	0	0
Permiso de Ejecución	1	=	0	0	1
Permiso de Escritura	2	=	0	1	0
Permiso de Escritura y Ejecución	3	=	0	1	1
Permiso de Lectura	4	=	1	0	0
Permiso de Lectura y Ejecución	5	=	1	0	1
Permiso de Lectura y Escritura	6	=	1	1	0
Permiso de Lectura, Escritura y Ejecución	7	=	1	1	1

Propietario
puede escribir
(w)

Grupo
puede escribir
(w)

Resto no puede
escribir
(-)



Permisos
Propietario

Permisos
Grupo

Permisos
Resto

ESTRUCTURA DEL ÁRBOL DE DIRECTORIOS

Una de las **diferencias principales** entre Linux y los sistemas Windows radica en el uso de un solo directorio raíz: (/).

Los siguientes directorios son los principales que derivan del directorio raíz:	
bin:	Incluye los comandos básicos del SISTEMA OPERATIVO (ejecutables como cd o cat). Tenemos un directorio similar <i>/usr/bin/</i> con ejecutables relacionados con aplicaciones instaladas.
sbin:	Incluye comandos esenciales para la administración del sistema operativo. Igual que con <i>bin</i> , encontramos un directorio <i>sbin</i> en <i>usr</i> .
boot:	Contiene los ficheros binarios de arranque del sistema.
dev:	Ficheros de dispositivos.
etc:	Incluye ficheros de configuración del sistema.
home:	Directorio donde se almacenan las carpetas personales de los usuarios que pueden acceder al sistema.

-
- **root:** directorio personal del usuario *root*.
 - **lib:** librerías básicas para trabajar con Linux. Estas son usadas por los programas ubicados en */bin* y */sbin*.
 - **mnt:** directorio destinado al montaje de dispositivos como CDs, DVDs o cualquier otro sistema de archivos después del arranque.
 - **proc:** lugar destinado a mostrar información del *kernel*. Se constituye un sistema de ficheros virtual donde se puede localizar información de interés sobre módulos del *kernel*/cargados, etcétera.
 - **var:** directorio usado para ubicar ficheros cuyo tamaño varía en el tiempo. Un ejemplo de estos son los archivos *log*, los cuales registran información del sistema en función de las operaciones que se van realizando.
 - **usr:** directorio dedicado a aplicaciones que no tienen relación directa con el sistema operativo, como los programas de aplicación.
 - **tmp:** almacena la información temporal de la sesión de usuario que se haya iniciado. Puede ser usado por todos los usuarios del sistema, ya que tiene permisos de lectura y escritura para todos, además del permiso *sticky bit*, que obliga a que cada usuario solo pueda modificar y eliminar lo que él crea conveniente.
 - **lost+found:** este es un directorio que usa la utilidad *fsck*. Este programa es llamado por el sistema cuando se producen fallos en el mismo y se quiere preservar la integridad del sistema de archivos. Por ejemplo, cuando se produce un apagón y no nos ha dado tiempo a cerrar la sesión adecuadamente ni a apagar de forma correcta el equipo. Cuando volvemos a encender el PC, tras la intervención de *fsck*, podemos acceder a *lost+found* y averiguar qué ficheros se guardaron bien.
 - **media:** Usado para el montaje de sistemas de ficheros temporales como *mnt*, siendo *media* el que se usa con más frecuencia en la actualidad.
 - **opt:** Almacena paquetes adicionales de aplicaciones.
 - **srv:** Incluye ficheros con información del sistema sobre ciertos ser vicos como *FTP*, *http*, etc.
 - **sys:** Muestra información sobre los dispositivos tal y como los ve el kernel.
-

Una **ruta** es la **referencia a la localización** de un fichero dentro del sistema de archivos del sistema operativo.

Se compone de los diferentes directorios que constituyen el camino que hay que recorrer hasta llegar a él. Cada uno de estos directorios se separa por el carácter “\”.

Hay **dos tipos** de rutas:

- **Absoluta**: señala la ubicación del fichero desde el directorio raíz del sistema de archivos.
- **Relativa**: señala la ubicación del fichero desde el punto en el que se encuentra el usuario.

➤ HERRAMIENTAS GRÁFICAS

Siempre que se trabaja con archivos, es posible hacerlo tanto a través de la interfaz gráfica como por comandos. La mayor parte de las operaciones vistas anteriormente se conocen en el entorno gráfico, puesto que solemos encontrarlas en el menú que desplegamos con el botón derecho del ratón.

2.5. Compresión y descompresión de ficheros

Como ya se ha visto anteriormente, hay distintos diferentes formatos de compresión, por lo que existen **distintos comandos para comprimir y descomprimir** archivos en Linux.

Comando

gzip/ gunzip

Permite comprimir y descomprimir archivos en formato *gz*, pero, a diferencia de otros, **NO permite la compresión** de directorios y subdirectorios.

```
~$ gzip archivo.gz fichero1 fichero2 fichero3 /directorio
```

Con la opción **-9**, se indica que utilice el mayor factor de compresión posible. Para descomprimir, se añade la opción **-d**:

Comando

tar

Es uno de los más utilizados en Linux. Se usa para **empaquetar varios archivos en uno solo**, sin comprimirlos, en formato *tar*.

Las posibles opciones son:

tar	-f:	Crea el archivo con el nombre del archivo dado.
	-t:	Testea el contenido .
	-u:	Añade a un archivo existente.
	-v:	Muestra en pantalla las operaciones que va realizando, archivo por archivo.
	-c:	Crea un nuevo archivo .
	-x:	Extrae el contenido .

Para extraer los contenidos de un archivo *tar*:

```
tar -xvf archivo.tar
```

```
~$ tar -xvf archivo.tar
```

2.6. Actualización del sistema operativo

A través de la terminal

Haciendo uso de la aplicación
integrada

Una de las tareas de mantenimiento más importantes dentro de un sistema operativo es **mantenerlo actualizado**, pues esto ayudará a corregir posibles errores derivados de las aplicaciones que tenemos instaladas y algunos de los errores que puede tener el propio sistema operativo.

En las distintas distribuciones Linux, esto se puede realizar:

- **a través de la terminal**
- **o haciendo uso de la aplicación integrada** en las últimas versiones, como, por ejemplo, de Ubuntu.

➤ Consola

Primero se abre la terminal (hay que tener en cuenta que, puesto que es una tarea que requiere permisos de administrador, es necesario estar como usuario **root**), y después se escribe lo siguiente:

sudo update -manager -devl-release

Este comando permite comprobar si existe una versión estable disponible para nuestro sistema operativo. Para ello, lista aquellas versiones adecuadas a nuestra distribución.

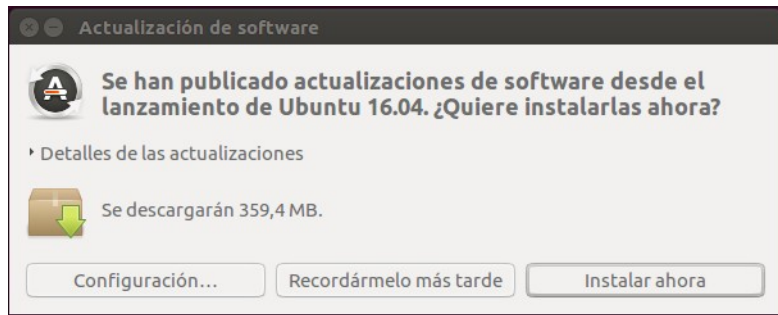
Por otro lado, para aplicar una actualización también hay que utilizar la terminal, donde se ingresa el siguiente comando:

sudo do-release-upgrade -d

Esto aplica la última actualización disponible en nuestro sistema operativo.

➤ Aplicación

La aplicación integrada en Linux provee de una interfaz que permite realizar estas actualizaciones de *software* de una manera más sencilla. Para ello, se elige la aplicación **Actualización de Software**, lo que abre un asistente que muestra si existe alguna actualización para instalar.






Es importante tener en cuenta que, para realizar estos procesos, es necesario tener una conexión estable a internet. Además, al finalizar es muy posible que el sistema pida reiniciar el equipo.

2.7. Agregación, configuración, eliminación y actualización del software del sistema operativo

En las distribuciones **GNU/Linux**, el *software* se distribuye en forma de **paquetes**, por lo que su manejo se realiza a través de un *gestor de paquetes*.

El **gestor de paquetes** es una colección de herramientas de administración que se utiliza para automatizar el proceso de instalación, actualización y configuración del *software*

Gestor de paquetes de GNU/Linux

		
RPM PACKAGE MANAGEMENT	DEBIAN / UBUNTU PACKAGE MANAGEMENT	
RMP	APT	SYNAPTIC

- **RPM**: es una herramienta de administración de paquetes desarrollada por **Red Hat**, aunque actualmente la usan otras distribuciones como Fedora, Mandriva y openSUSE, entre otras.

Sus principales características son:

- **Las dependencias entre paquetes** se resuelven automáticamente por el gestor de paquetes.
 - **La seguridad es importante**, los paquetes se cifran y se verifican con diferentes algoritmos de cifrado.
-

- **APT**: sistema de gestión de paquetes creado por Debian. Para descargar un paquete *software* se utiliza el comando `apt`, para instalarlo se utiliza `apt install` y para desinstalarlo es necesario introducir el comando `apt remove`.

Es el gestor de paquetes utilizado en distribuciones Debian y Ubuntu.

- **Synaptic**: es un **gestor de paquetes gráfico**. Se instala a través del comando `apt`, y ofrece una mayor cantidad de información y gestión sobre los paquetes.

2.8. Configuración de dispositivos periféricos en diferentes sistemas operativos

Linux tiene varios ficheros de carácter especial en la ruta **/dev** que reciben el nombre de **ficheros de dispositivos**, los cuales permiten la comunicación con el controlador correspondiente a cada dispositivo y establecer una configuración para poder usarlo correctamente.

No obstante, esta configuración se realiza automáticamente en las últimas versiones de Ubuntu. Una vez el sistema detecta la conexión del dispositivo, muestra la ventana del asistente que presentará las instrucciones a seguir durante el proceso. Después de la instalación del controlador correcto, es recomendable comprobar su funcionamiento verificando que en la ruta correspondiente existe la carpeta creada para el tipo de dispositivo instalado.

Tanto la instalación como la configuración de un dispositivo se puede realizar a través de la terminal de Linux. En este caso, es necesario,

1. en primer lugar, **MONTAR EL DISPOSITIVO**, es decir, crear en el fichero de dispositivos el directorio asociado a dicho periférico.

2. Después, **DESDE LA TERMINAL SE ESCRIBE:**

```
sudo mkdir /media/usb.
```

Es necesario conocer el nombre que tiene el dispositivo, para lo que hay que listar todos los dispositivos conectados al equipo mediante la orden:

```
ls -l /dev/sd*
```

Una vez identificado, lo montamos:

- (Formato FAT): `mount -t vfat /dev/sde1 /media/usb`
- (Formato NTFS): `mount -t ntfs-3g /dev/sde1 /media/usb`

Donde **sde1** es el nombre que el sistema operativo utiliza para identificar el dispositivo.

Ahora, dentro de la carpeta **/media/usb** estarán todos los archivos y directorios que estén el dispositivo.

También es posible realizar la operación contraria, es decir, desmontar el dispositivo. Para ello se utiliza el comando:

```
umount /media/usb
```

Donde **/usb** es el nombre del directorio creado para el USB. En caso de tener conectados varios USB por ejemplo, se introduce lo siguiente:

```
mount -t vfat /dev/sde1 /media/usb2
```

Y para desmontar:

```
umount /media/usb2.
```

2.9. Inventario del software instalado

Durante el uso y vida de un equipo, se realizan de manera continuada una **gran cantidad de instalaciones y desinstalaciones** de *software*. Por eso, es importante revisar periódicamente la cantidad de programas que hay instalados.

En Linux, se puede consultar un listado que detalla todos los elementos *software* existentes usando el siguiente comando en la terminal:

```
dpkg --get-selections
```

Cada una de las filas que aparecen en ese listado es un programa. En la mayor parte de los casos, la cantidad de aplicaciones instaladas será mucho más extensa que la mostrada como resultado en la terminal. Por eso, si se quiere ver la lista completa, hay que exportar el contenido a un fichero de texto de la siguiente manera:

```
dpkg --get-selections > nombre_fichero
```

De esta forma, se creará un fichero dentro de la ruta escogida en la terminal con el nombre especificado.

Este se puede abrir con un editor de texto (como puede ser Gedit) para visualizar el listado completo. Después, se puede buscar con mayor eficiencia cualquier programa dentro del equipo y realizar la tarea que se desee con él.

Por otra parte, si lo que se quiere hacer es desinstalar un programa, una vez obtenido su nombre se ejecuta la siguiente orden en la terminal:

```
sudo apt-get remove Programa
```

Teniendo en cuenta que ***Programa*** es el nombre del programa elegido.

2.10. Funcionamiento correcto de las configuraciones realizadas

Tras realizar una instalación, conviene comprobar que la configuración asignada es la adecuada para asegurar el correcto funcionamiento.

En Linux, existen algunas órdenes que permiten comprobar de una manera rápida el estado de los distintos dispositivos conectados al equipo.

En el **hardware**:

Listar los dispositivos PCI / PCIe	<i>lspci</i>	
Listar todos los dispositivos USB	<i>lsusb</i>	
Listar el hardware (información resumida)	<i>sudo lshw -short</i>	
Listar el hardware (información completa)	<i>sudo lshw less</i>	Requiere realizar previamente la instalación de <i>lshw</i>
Listar el hardware (información completa):		
Listar las particiones	<i>sudo fdisk -l</i>	
Listar los dispositivos de audio	<i>aplay -l grep -i tarjeta</i>	
Mostrar el espacio usado en disco:	<i>df -h</i>	

Cuando se ejecuta desde la terminal, la información se encuentra resumida. Por ello, Linux permite la instalación de *software* externo que muestra a través de una interfaz gráfica toda la información relacionada con el equipo. Las herramientas más conocidas que permiten tener una visión global de dichas configuraciones son ***hardinfo*** y ***sysinfo***.

Para instalarlos, basta con ejecutar los siguientes comandos:

```
sudo apt-get install hardinfo sudo apt-get install sysinfo
```

También es posible realizarlo mediante el uso del ***Gestor de paquetes***

Tan solo es necesario buscarlo e instalarlo siguiendo los pasos que determina el asistente.

Finalmente, si los programas están correctamente instalados, al realizar el listado aparecerá en la columna consecutiva el estado ***instalado***. Si esto no es así, significa que el programa o el paquete no se ha instalado.

2.11. Documentación del proceso de configuración. Interpretación de la documentación técnica

En cualquier versión del sistema operativo Linux es importante, antes de instalar un programa o aplicación, revisar la **documentación técnica del fabricante** relacionada con todos los requisitos y pasos a seguir durante el proceso.

Esta documentación debe tener una guía de la configuración del procedimiento de instalación y los parámetros de configuración recomendados que garanticen que el funcionamiento será correcto en el sistema. Es posible encontrar diferentes escenarios de configuración basados en los intereses del usuario. Además, puede realizarse una instalación con la configuración recomendada o bien una instalación personalizada, en la que se seleccionan aquellos componentes que uno desea instalar en el equipo.

3. Administración de los sistemas operativos libres

3.1. Creación y gestión de los usuarios y grupos

Gestión de perfiles de usuarios y grupos locales

Las tareas de administración en Linux solo las puede realizar el usuario *root*, por medio de los comandos de gestión de usuarios. Por ello, para cualquiera de estos comandos es necesario estar identificado como tal o poner *sudo* delante de cada uno de los comandos.

- **Usuarios**: de esta forma se consigue que las cuentas de usuario creadas controlen los archivos y programas a los que tendrán permiso de acceso y qué tipo de acciones podrán realizar. El comando utilizado para añadir un **nuevo usuario** es ***useradd*** y su sintaxis es:

useradd [opciones] nombreUsuario

Las opciones que se pueden establecer son:

-g	identificador del grupo al que va a pertenecer el usuario. Debe existir previamente.
-d:	carpeta <i>home</i> del usuario. Normalmente se encuentra en la ruta <i>/home/nombreUsuario</i> .
-m	crea la carpeta <i>home</i> , si es que no existe
-s	- referencia a la <i>Shell</i> del sistema operativo del usuario. Suele ser

	<code>/bin/bash.</code>
--	-------------------------

No obstante, existen otros comandos para la gestión de los usuarios:

- ***usermod***: permite la modificación de la información del usuario.
- ***userdel***: permite la eliminación de un usuario.

- **Grupos**: de la misma forma que se gestionan los usuarios, es posible gestionar los grupos asociados a ellos. Para añadir un **nuevo grupo** se utiliza el comando **groupadd** y su sintaxis es:

```
groupadd nombreGrupo
```

Para la **modificación** de un grupo se utiliza el comando **groupmod**, que sigue la siguiente sintaxis:

```
groupmod -g identificador nombre  
Grupo
```

La modificación es realizada mediante el identificador (**GID**) del grupo, aunque también es posible hacerla mediante su propio nombre. Con el comando anterior se cambia el nombre de ese grupo.

Para la **eliminación** de un determinado grupo se utiliza el comando:

```
groupdel nombreGrupo
```

Por medio del nombre del grupo es posible eliminarlo. En caso de que exista dentro del grupo un usuario que tenga asignado este grupo como principal, no será posible hacerlo.

Una vez se han creado los diferentes grupos y usuarios del sistema, es posible **establecer una relación entre ellos**, es decir, indicar a qué grupo pertenece cada usuario.

Finalmente, para **añadir un usuario** a un grupo se utiliza el comando **adduser**, con la siguiente sintaxis:

```
adduser nombreUsuario nombreGrupo
```

- **Contraseñas:** cuando hay distintos usuarios en un mismo equipo, cada uno de ellos puede tener cierta privacidad en su sesión si establece una contraseña. Para ello, se utiliza el comando *passwd*, seguido del nombre de usuario al que se le quiere poner.

Hay que tener en cuenta que un usuario puede establecer su propia contraseña, pero no la del resto de usuarios, por lo que es necesario utilizar este comando siendo **root**.

La sintaxis es la siguiente:

passwd nombreUsuario

Cabe mencionar que hay que establecerla dos veces por motivos de seguridad, ya que será necesaria para el acceso y los permisos de instalaciones en la cuenta de usuario.

- **Permisos:** cuando ya se han definido todos los usuarios y grupos, se realiza la gestión de los permisos. En este momento se definen los recursos que cada uno de los usuarios o grupos podrá ver, modificar o eliminar.

3.2. Gestión del sistema de archivos

➤ Herramientas gráficas

Los sistemas operativos cuentan con una **cantidad elevada de archivos o ficheros** que son utilizados por diversas fuentes. Esto implica que exista una organización que permita separar por carpetas o directorios unos de otros.

En Linux existen **tres tipos** de ficheros:

- **Carpetas o directorios:** permiten la organización jerárquica del sistema de ficheros.
- **Ficheros regulares u ordinarios:** son los archivos con los que se trabaja. Pueden ser de cualquier tipo y según su formato indican el modo en que se gestionan los ficheros dentro de las particiones.
 - Linux contiene sistemas de ficheros basados en discos como ext2, ext3, ext4, FAT, FAT32 o NTFS. Sistemas de ficheros para la comunicación en red, NFS, para compartir recursos entre equipos Linux.
 - Existe un tipo denominado SWAP que es una especie de partición dentro del disco duro que se utiliza como punto de carga de aplicaciones o programas y para reducir la carga en memoria RAM.
- **Ficheros especiales:** son los usados para la comunicación con los distintos periféricos.

➤ Consola

En Linux podemos gestionar y navegar a través del sistema de archivos desde la consola integrada. El término por el que se conoce al intérprete de comandos en Linux es *Shell*.

La Shell permite **ejecutar programas** que forman parte de nuestro sistema operativo mediante comandos que se escriben en una consola en modo texto. El Shell de Linux es *bash*.

3.3. Gestión de los procesos del sistema y del usuario

➤ Activación y desactivación de los servicios

Los servicios de Linux son **procesos que se ejecutan de forma continua** una vez se ejecuta el sistema operativo. En Linux, reciben el nombre de **demonios**.

Los **demonios** son **scripts** que ejecutan una serie de porciones de código que suelen localizarse en el directorio */etc/init.d*.

Para activarlos y desactivarlos solo se tiene que añadir *start* o *stop* como parámetro en el comando de ejecución:

root ruta / nombreServicio estado

• Iniciar un servicio:	<i>root /etc /init.d/demonio start</i>
• Detener un servicio:	<i>root /etc /init.d/demonio stop</i>
• Reiniciar un servicio:	<i>root /etc /init.d/demonio restart</i>

Este tipo de servicios se ejecutan en segundo plano, por lo que no forman parte del control del usuario y no tienen interfaz gráfica.

3.4.

3.5. Optimización de la memoria y del funcionamiento de los dispositivos de almacenamiento

Optimizar la memoria es una de las tareas de mejora del rendimiento necesarias durante la ejecución de un sistema operativo. La más importante ya ha sido comentada en un apartado anterior, es la que se conoce como **SWAP**.

Linux no cuenta con un sistema de paginación definido, por lo que, para evitar que la memoria RAM colapse y se sobrecargue, se emplea este mecanismo. Consiste en la **creación de un espacio de intercambio dentro del disco duro**, donde se almacenarán todos los datos que no se pueden alojar dentro de la

memoria RAM. Esta asignación de espacio en Linux se realiza durante la instalación del sistema operativo.

A pesar de que es algo que para el sistema Linux supone una mejora notable en el rendimiento, hay que tener en cuenta que este uso de memoria se debe minimizar. El equipo tiene unos recursos *hardware* limitados porque esto necesita una serie de librerías que intercambien información con la memoria RAM.

Para consultar el espacio asignado en memoria se utiliza el comando **free**.

```
ilernaonline@ilernaonline-virtual-machine:~/Escritorio$ free
              total        used         free       shared  buff/cache   available
Memoria:    2030628      833952      619444         9780     577232     992336
Swap:      2094076           0      2094076
```

Este es el espacio de memoria que mantiene los estados de los archivos en ejecución también cuando se pone el equipo en modo de hibernación, por lo que la memoria SWAP se podría definir como una memoria caché.

3.6. Rendimiento del sistema

Al igual que otros sistemas operativos, Linux tiene equipado un **monitor** en alguna de sus distribuciones que permite consultar de forma gráfica y en tiempo real el rendimiento del equipo.

Se puede consultar en la **consola del sistema** mediante el uso de algunos comandos:

• Monitorizar la carga del sistema:	<i>uptime</i>
• Tiempo de ejecución de un programa:	<i>time</i>
• Actividad de los procesos:	<i>top</i>
• Actividad de la memoria:	<i>vmstat</i>

Herramientas del sistema de seguimiento y monitorización

En Ubuntu existe una aplicación denominada **Monitor del sistema** que permite observar el rendimiento de los procesos en ejecución y el porcentaje de CPU utilizado por cada uno de ellos y conocer el uso de los recursos *hardware* del equipo (tanto de la memoria RAM como de CPU). También se puede observar el espacio ocupado en los discos duros.

Por otra parte, no solo es posible medir el rendimiento de los recursos y procesos, sino también monitorizar el tráfico que hay en la red. El analizador de paquetes **TCPDUMD** permite capturar todos los paquetes **TCP** que circulan por una interfaz específica, mediante la orden **tcpdump**.

Si lo que se desea es mostrar las estadísticas de red entrantes y salientes, presentará el estado de todas las conexiones que se hayan registrado en el sistema e indicará aquellas que permanecen activas. Además, existen otras que no están integradas, tanto de código abierto como propietarias. Estas ofrecen un mayor catálogo de opciones y mejoras en la interfaz gráfica para recoger estadísticas del sistema.

3.7. Compartición de recursos

En la actualidad, es necesario disponer de unos determinados recursos que muchas veces es imposible conseguir físicamente. Además, se necesita que un mismo ordenador pueda ser usado por diferentes usuarios.

Algunas de las opciones que existen para **gestionar los recursos compartidos** son:

- **SAMBA**: esta implementación del protocolo **SMB** de Microsoft libre en Linux permite configurar una serie de directorios para compartirlos a través de la red. Estos tienen unos permisos de acceso (lectura y escritura) ya definidos.

- **NFS**: es necesario instalar el paquete *nfs-utils* en aquellos equipos que necesitan acceder a la carpeta compartida. Se habilita el servicio en cada uno de ellos y se establecen los permisos de acceso desde el ordenador que crea el recurso compartido.

- **Transferir de un ordenador** a otro los archivos deseados por medio de órdenes como **SSH** o **SCP**.

3.8. Interpretación de datos de configuración y comportamiento del sistema operativo

➤ Hardware instalado

Existe una herramienta ya integrada en Linux que permite **conocer con detalle el listado de *hardware*** que tiene instalado el equipo.

Para que el equipo devuelva un listado completo del sistema se ejecuta desde la terminal del sistema la orden ***lshw***.

Es posible copiar el resultado directamente en un archivo generado:

```
sudo lshw -html > nombreArchivo.extension
```

Mediante la opción **-C** es posible conocer un mayor detalle de los dispositivos:

```
sudo lshw -C disk
```

Si se quiere visualizar esta información mediante una interfaz gráfica, entonces es necesaria la opción **-gtk**:

```
sudo lshw-gtk
```

➤ Aplicaciones

En el apartado anterior se ha visto cómo listar todos los componentes *hardware*. No obstante, para ver el listado de los componentes *software* se hace uso de la orden

dpkg -l

De esta forma, se muestra el listado completo de paquetes instalados en el sistema. Si se desea conocer toda la información relativa a la versión del sistema operativo instalado se realiza la siguiente operación: desde la terminal localizamos el archivo que lo define e incluimos el comando

cat/etc/issue.net

, que mostrará el nombre de la distribución de Linux instalada. No obstante, si deseamos un mayor detalle introduciremos

lsb_release -a

, que mostrará los módulos y la descripción del sistema operativo.

➤ Técnicas de mantenimiento del software de aplicación

Todos los sistemas operativos necesitan realizar continuamente comprobaciones de posibles **paquetes de actualización**. Estos corrigen algunos de los fallos que existen en el sistema y aumentan la fiabilidad de uso. Para acceder a ellos, se ingresan determinadas órdenes en la terminal: ***sudo apt-get update***

sudo apt-get dist-upgrade

Con esto, se descargan e instalan todas las actualizaciones disponibles para el sistema, tanto a nivel de sistema operativo como de aplicaciones.

➤ Copias de seguridad

Realizar copias de seguridad de los archivos más importantes es una buena práctica dentro del mantenimiento.

➤ Archivos temporales

Otra de las tareas es el **borrado de los archivos temporales**. Para ello, es necesario acceder a la carpeta ***/tmp***. Linux realiza esta limpieza de manera automática.

➤ Programas que no se utilizan

También es útil desinstalar programas de los que no se hace uso. Se pueden desinstalar, junto con sus configuraciones, a través del comando

sudo apt-get purge [package],

aunque si estas últimas quieren dejarse almacenadas se usa ***sudo apt-get remove [package]***.

Desfragmentación de discos

En Linux no es necesario desfragmentar el sistema de archivos, pues es posible instalar el sistema operativo con las configuraciones básicas de funcionamiento. Sin embargo, existe una herramienta de gran utilidad que permite la reparación del sistema de archivos por medio de la orden ***FCK***.

3.9. Automatización de tareas

Linux permite la **automatización de algunos procesos** o servicios. Uno de los más usados e importantes es ***cron***, que se encuentra en la ruta ***/etc/cron.d*** y cuya configuración y permisos son editables.

Dentro de esta ruta, encontramos **tres ficheros**:

```
ilernaonline@ilernaonline-virtual-machine:/etc$ gedit crontab
ilernaonline@ilernaonline-virtual-machine:/etc$
```

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --
report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --
report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --
report /etc/cron.monthly )
#
```

- ***crontab***: especificación de todas las tareas.
- ***cron.allow***: usuarios con permisos para ejecutar cada una de las tareas.
- ***cron.deny***: usuarios que no tienen permisos para ejecutar alguna tarea.

Las tareas especificadas dentro del archivo ***crontab*** se ejecutan en el tiempo y frecuencia especificadas. Además, para modificar los ***cron*** del usuario en uso se utiliza el comando

crontab -e.

Por otra parte, para especificar una tarea es necesario indicar la frecuencia con la que se ejecutará. Para ello, hay que escribir la siguiente sintaxis:

[minuto] [hora] [día] [mes] [día_de_la semana] [rutaScript]

Hay que tener en cuenta que especificar los **días de la semana** se realiza de **forma numérica**, teniendo en cuenta que el domingo es el 0, el lunes el 1 y así sucesivamente.

Por ejemplo:

```
0 12 * * 1 /Escritorio/script.sh  
#
```

Con esta línea ejecutamos el *script* todos los lunes a las doce del mediodía.

3.10. Ejecución de programas y guiones administrativos

En Linux **no siempre resulta fácil** ejecutar un determinado archivo o programa. En primer lugar, se van a definir algunos de los tipos de archivos ejecutables que existen en este sistema operativo.

- **.bin y .run**: suelen ser los archivos de instalación de las aplicaciones.
- **.sh**: son los *scripts* que se ejecutan directamente desde la consola.

Como parte de la seguridad del sistema operativo, Linux bloquea la ejecución de programas y aplicaciones que no tienen los permisos necesarios.

Por eso, desde la terminal se navega a la carpeta o directorio donde está contenido el archivo a ejecutar. Para ejecutarlo simplemente se escribe

/nombreArchivo.

Además, es conveniente instalar los programas como administrador, lo cual se consigue añadiendo *sudo* en la orden anterior

sudo /nombreArchivo.

3.11. Métodos para la recuperación del sistema operativo

La opción más fácil y rápida para realizar la recuperación de un sistema operativo es el uso del **sistema de copias de seguridad** creado de nuestro disco duro y archivos más importantes. En caso de no tener realizadas estas copias, sería necesario arrancar en el **modo rescate** de Linux, para lo cual se deberá utilizar uno de los siguientes métodos:

- Arrancar desde un CD-ROM o DVD.
- Arrancar desde otros medios de arranque, como dispositivos *flash* USB.
- Arrancar desde el DVD de Red Hat Enterprise Linux.

Una vez se haya iniciado el sistema operativo en la terminal, se escribe:

Linux rescue dd

De esta forma, el sistema arranca descargando, en caso de ser necesario, el controlador del disco. Además, realizará automáticamente la búsqueda dentro del disco de una imagen estable para montarla en él y restaurar el sistema operativo. Este proceso normalmente se realiza en la ruta ***/mnt/sysimage***.

Otra de las opciones para recuperar el sistema operativo es el uso del *LIVECD*, que arranca el sistema operativo desde el CD para realizar una reparación o instalación nueva.

3.12. Comprobación del correcto funcionamiento del sistema

Mantenimiento del inventario del *software* utilizado y seguimiento de cambios

Es posible acceder al inventario *software* instalado, como ya se ha comentado, por medio del comando

dpkg -l

el cual muestra el listado completo de paquetes instalados en el sistema.

La mejor forma de llevar a cabo un mantenimiento óptimo es usando herramientas externas.

Linux Mint: es, básicamente, un administrador de actualizaciones. Gestiona y lleva a cabo las actualizaciones de forma automática, aunque también es capaz de realizar copias de seguridad de los

3.13. Documentación de las tareas de administración y las incidencias aparecidas con sus soluciones. Interpretación de la documentación técnica

Todo sistema operativo Linux tiene incluido **un apartado con la documentación** necesaria para cualquier tipo de gestión. Es, por tanto, una buena práctica realizar las tareas siguiendo los pasos que se indican en ella.

Además, contiene apartados relacionados con algunas de las incidencias que se han encontrado otros usuarios, junto con su solución. En caso de que la incidencia ocurrida en el equipo no aparezca solventada, será necesario ponerse en contacto con el fabricante y explicar con detalle el problema.

Bibliografía

Unix y linux: guía practica. Sebastian Sanchez Prieto; Óscar García Población, 2004. McMillan

<https://www.xatakawindows.com/>

[https://technet.microsoft.com/es-es/library/cc709667\(v=ws.10\).aspx](https://technet.microsoft.com/es-es/library/cc709667(v=ws.10).aspx)

http://www.ite.educacion.es/formacion/materiales/130/cd/indice_ub_untu.htm

<https://www.microsoft.com/es-es/evalcenter/evaluate-windows-10-enterprise>

<https://ubuntu.com/download/desktop>

```
5 function updatePhotoDescription() {  
6     if (descriptions.length > (page * 9) + (currentimage.substring(0, 1)))  
7         document.getElementById("bigimageDesc").innerHTML = descriptions[page * 9 + currentimage.substring(0, 1)]  
8     }  
9 }
```