

# **Manejo básico de Ubuntu**

## Sumario

1. Objeto.....	2
2. El entorno de trabajo en entorno gráfico.....	3
3. Gestión del sistema por línea de comandos .....	5
6. Gestión de procesos y memoria .....	9
7. Gestión de E/S.....	10
8. Archivos y directorios.....	12
9. Variables de entorno .....	13
10. El registro en Linux.....	14

## 1. Objeto

Este documento describe las opciones básicas de uso de Ubuntu. Se describe tanto el sistema con entorno gráfico (Desktop) como la línea de comandos relacionándolos con los cuatro servicios básicos de todo sistema operativo:

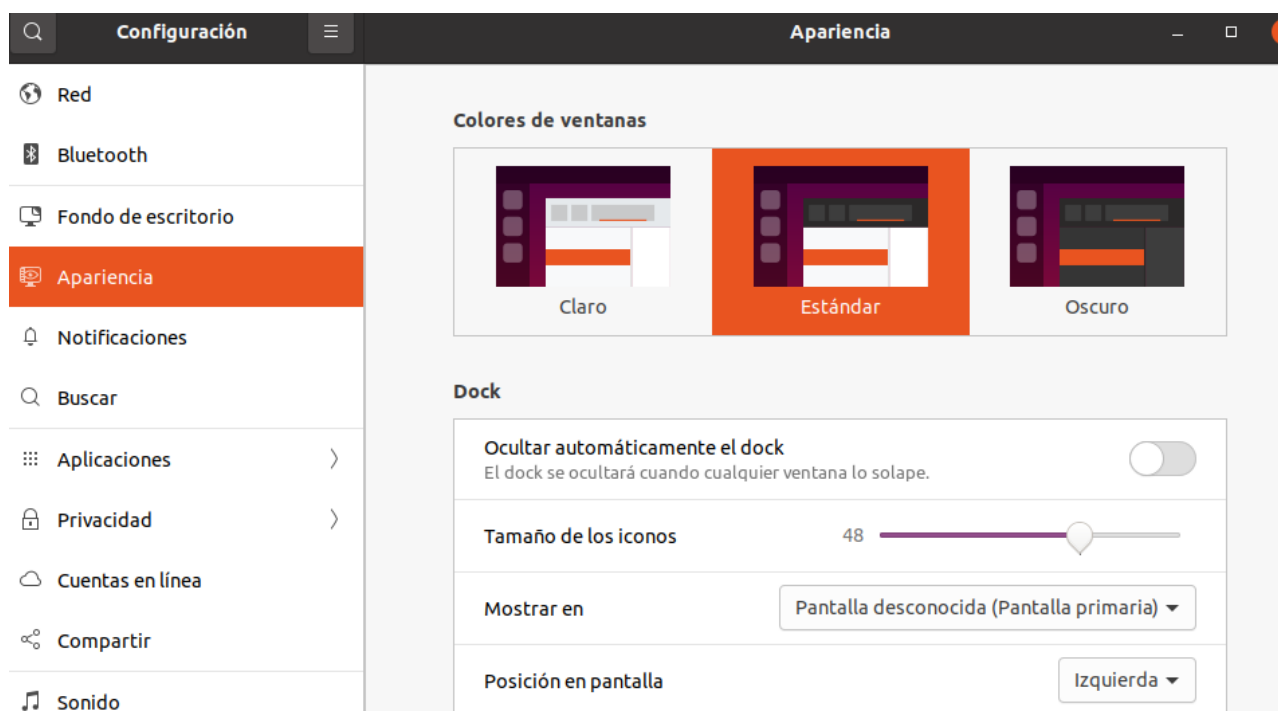
- gestión de procesos
- gestión de memoria
- gestión de dispositivos de entrada/salida
- gestión de archivos

Se trata de una visión a alto nivel, ya que se irá profundizando en unidades didácticas posteriores.

## 2. El entorno de trabajo en entorno gráfico

Los entornos de escritorio más utilizados en Linux son GNOME (usado en Ubuntu) o KDE (usado en Kubuntu, por ejemplo). En nuestro caso usamos GNOME, que es el entorno por defecto de Ubuntu. Un entorno de escritorio incluye un conjunto de aplicaciones que dotan al sistema de un entorno gráfico completo.

Utiliza por defecto el tema Yaru, pero podemos cambiarlo accediendo a *Mostrar aplicaciones* → *configuración* → *apariencia*



Los componentes del escritorio de Ubuntu son:

- **Mostrar aplicaciones:** botón en la esquina inferior izquierda. Permite seleccionar las aplicaciones que queremos ejecutar. Una vez que ejecutemos una aplicación, se añadirá a la pestaña de Frecuentes.
- **Barra superior:** proporciona acceso a ventanas y sus aplicaciones y a propiedades del sistema como el sonido, red o energía. Permite también salir de la sesión, cambiar de usuario, apagar o reiniciar.
- **Vista de actividades:** se puede acceder bien pulsando con el ratón sobre *Actividades* en la barra superior, o pulsando en el teclado el botón de Windows (de ahora en adelante **Super**). Permite acceder a las aplicaciones que estamos actualmente ejecutando y realizar operaciones con ellas.
- **Menú aplicaciones:** está junto al botón de actividades. Muestra la aplicación activa

junto con su icono. Proporciona acceso rápido a las ventanas y detalles de la aplicación.

- **Reloj, calendario y citas:** si se pulsa en reloj, se ve la fecha actual, un calendario mensual y las citas próximas. Se puede abrir el calendario con **<Super + M>**
- **Menú del sistema:** esquina superior derecha. Permite bloquear el equipo si no se usa, cambiar de usuario, apagar, reiniciar.
- **Lista de ventanas:** se puede cambiar entre ventanas mediante la combinación **<Super + Tab>**, exactamente igual que se puede hacer en Windows.

GNOME incluye Nautilus como explorador de archivos al estilo de Windows. Está accesible en la barra lateral.

Guía detallada del escritorio de Ubuntu:

<https://help.ubuntu.com/lts/ubuntu-help/index.html>

### 3. Gestión del sistema por línea de comandos

Se trata de la opción más habitual entre usuarios acostumbrados al sistema Linux. Es común para Desktop y Server.

Cada usuario trabaja con el intérprete de comandos (shell) que tiene configurado. Ya veremos esta configuración más adelante, cuando revisemos cómo gestionar usuarios.

La shell es el programa encargado de recibir las órdenes de los usuarios, interpretarlas y ejecutarlas. La más común actualmente es la *bash*, pero existen otras como *sh*, *ksh* o *csh*.

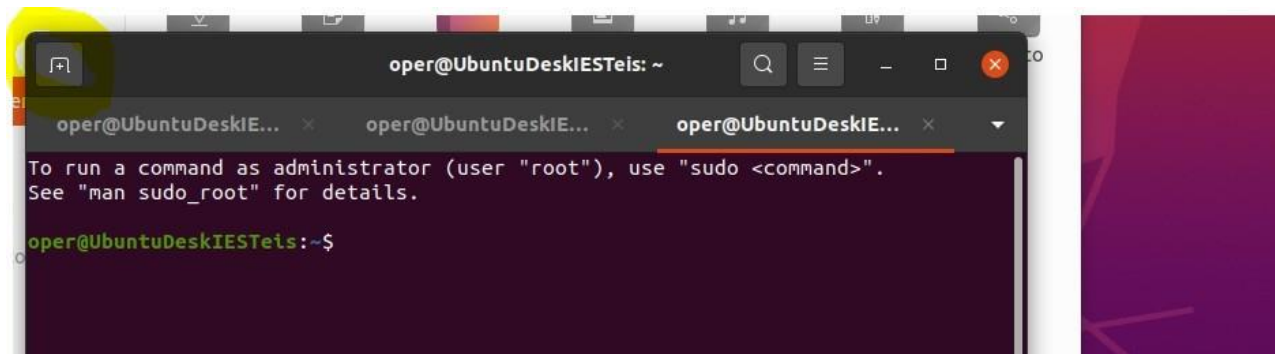
La shell proporciona un lenguaje de scripting extremadamente potente. En el siguiente enlace puedes encontrar todo el detalle:

<https://tldp.org/LDP/abs/html/>

En el Server trabajamos directamente en modo consola. En el Desktop dos maneras de abrir la consola:

- Pinchar en *Mostrar aplicaciones* y escribir *Terminal* (si ya lo hemos abierto antes, lo tendremos en Frecuentes).
- Pulsar la combinación **<Ctrl + Alt + t>**

El Desktop nos permite tener abiertos varios terminales simultáneamente, lo que es muy útil. Para ello no tenemos más que pulsar en el botón superior izquierdo de la ventana del terminal:

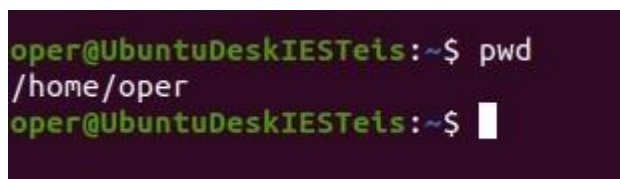


Una de las mayores diferencias en Ubuntu respecto a Windows es que nunca trabajaremos como usuario administrador (root). Es posible trabajar con él directamente, pero nada recomendable por seguridad. En su lugar, cuando queramos ejecutar algún comando con privilegios de root lo ejecutaremos poniendo delante “sudo”. Por ejemplo:

**sudo mount /dev/cdrom /cdrom**

Veremos más adelante cómo darle privilegios a un usuario normal para que pueda usar el “sudo”. Durante la instalación del sistema operativo, el usuario que cada uno ha creado tiene ya privilegios.

Por defecto, el usuario trabaja en su directorio HOME, identificado por el carácter “~”. Este home se configura al dar de alta usuarios, y veremos cómo hacerlo más adelante. Por defecto es el directorio `/home/<usuario>`



```
oper@UbuntuDeskIESTeIs:~$ pwd
/home/oper
oper@UbuntuDeskIESTeIs:~$
```

En Linux los directorios no parte de una letra de unidad, como en Windows. En su lugar, parten de “/”. Conceptos importantes (también aplicables a Windows):

- ruta absoluta: camino (path) completo para llegar a un directorio o fichero del árbol. Por ejemplo, `/home/oper/fichero.txt`
- ruta relativa: path desde el punto en que nos encontramos. En el ejemplo anterior:
  - si estuviésemos situados sobre `/home`, la ruta relativa es `oper/fichero.txt`
  - si estuviésemos situados sobre `/home/oper`, la ruta relativa es `fichero.txt`

Se listan a continuación comandos básicos que son necesarios. Iremos avanzando a lo largo del módulo, aprendiendo a usar más.

cd	Cambia al directorio especificado como parámetro. Si no se especifica nada se cambia al HOME del usuario. Se puede usar “..” para subir al directorio padre.
ls	Lista los contenidos de un directorio
mkdir	Crea un directorio.
rmdir	Elimina un directorio (debe estar vacío)
rm	Borra ficheros/directorios
mv	Mueve un fichero/directorio de un punto a otro
cp	Copia un fichero/directorio de un punto a otro
cat	Muestra el contenido de un fichero
more	Muestra el contenido del fichero, de forma paginada
tail	Muestra las últimas líneas del fichero
pwd	Nos muestra la ruta absoluta en la que estamos

En el siguiente link puedes encontrar detalle de muchos de los comandos que iremos trabajando:

<https://blog.desdelinux.net/mas-de-400-comandos-para-gnulinix-que-deberias-conocer/>

El sistema utiliza un código de colores para distinguir tipos de entradas en el árbol de directorios. Estos colores se pueden personalizar, pero por defecto son:

- Verde: ejecutables. Si intentamos ejecutar un archivo que no sea ejecutable, dará un error.
- Azul: directorio
- Rojo: comprimido
- Magenta: imágenes
- Sin color (blanco): ninguno de los anteriores

```
oper@UbuntuDeskIESTeis:~/Documentos/juan$ ls -lrt
total 8
-rw-rw-r-- 1 oper oper  0 oct  9 19:51 uno.txt
-rw-rw-r-- 1 oper oper 28 oct  9 19:51 dos.txt.gz
drwxrwxr-x 2 oper oper 4096 oct  9 19:51 subdir
-rw-rw-r-- 1 oper oper  0 oct  9 19:51 imagen.jpg
-rwxrwxr-x 1 oper oper  0 oct  9 19:52 ejecutable
```

MUY IMPORTANTE: cuando quieres ejecutar un ejecutable, lo puedes hacer de dos modos:

- Utilizando la ruta absoluta:

```
oper@UbuntuDeskIESTeis:~/Documentos/juan$ /home/oper/Documentos/juan/ejecutable
Programa ejecutado
oper@UbuntuDeskIESTeis:~/Documentos/juan$
```

- Utilizando ruta relativa respecto del punto en el que estamos:

```
oper@UbuntuDeskIESTeis:~/Documentos$ juan/ejecutable
Programa ejecutado
oper@UbuntuDeskIESTeis:~/Documentos$
```

- Ejecutar un programa que está dentro de nuestro directorio actual. En este caso, debemos especificar explícitamente que está en el directorio actual, anteponiendo “./”. Realmente estamos también especificando una ruta relativa.

```
oper@UbuntuDeskIESTeis:~/Documentos/juan$ ./ejecutable
Programa ejecutado
oper@UbuntuDeskIESTeis:~/Documentos/juan$
```

No es posible obviar ese “./” porque, si no especificamos la ruta (sea relativa o absoluta), Linux busca los comandos que tecleamos dentro de los directorios configurados en la variable PATH (la veremos más adelante. El directorio actual no está dentro del PATH.



Una solución podría parecer incluir “./” en el PATH, pero no se recomienda por seguridad.

Podemos obtener ayuda de cualquier comando ejecutando:

**man <comando>**

## 6. Gestión de procesos y memoria

Veremos la gestión de procesos y memoria en detalle más adelante, cuando revisemos la monitorización.

De momento, en caso de que quieras ver cómo está el sistema de carga, puedes usar el comando “top”.

```
top - 20:04:55 up 49 min, 1 user, load average: 0,00, 0,02, 0,05
Tareas: 180 total, 1 ejecutar, 179 hibernar, 0 detener, 0 zombie
%Cpu(s): 0,7 usuario, 0,0 sist, 0,0 adecuado, 99,3 inact, 0,0 en espera, 0,
MiB Mem : 1985,3 total, 324,9 libre, 737,8 usado, 922,5 búfer/caché
MiB Intercambio: 448,5 total, 448,5 libre, 0,0 usado. 1079,3 dispon
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
1363	oper	20	0	546608	73664	46400	S	1,3	3,6	0:13.26	Xorg
1555	oper	20	0	3756968	381912	129884	S	1,0	18,8	0:56.28	gnome-s+
2465	oper	20	0	825464	53204	39508	S	0,7	2,6	0:03.40	gnome-t+
1	root	20	0	167684	11420	8296	S	0,0	0,6	0:01.42	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par+
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
9	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_perc+
10	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_tas+
11	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_tas+
12	root	20	0	0	0	0	S	0,0	0,0	0:00.12	ksoftir+
13	root	20	0	0	0	0	I	0,0	0,0	0:00.45	rcu_sch+
14	root	rt	0	0	0	0	S	0,0	0,0	0:00.04	migrati+
15	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	idle_in+
16	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/0

Para salir de **top**: pulsa la tecla “q”.

<https://geekytheory.com/funcionamiento-del-comando-top-en-linux>

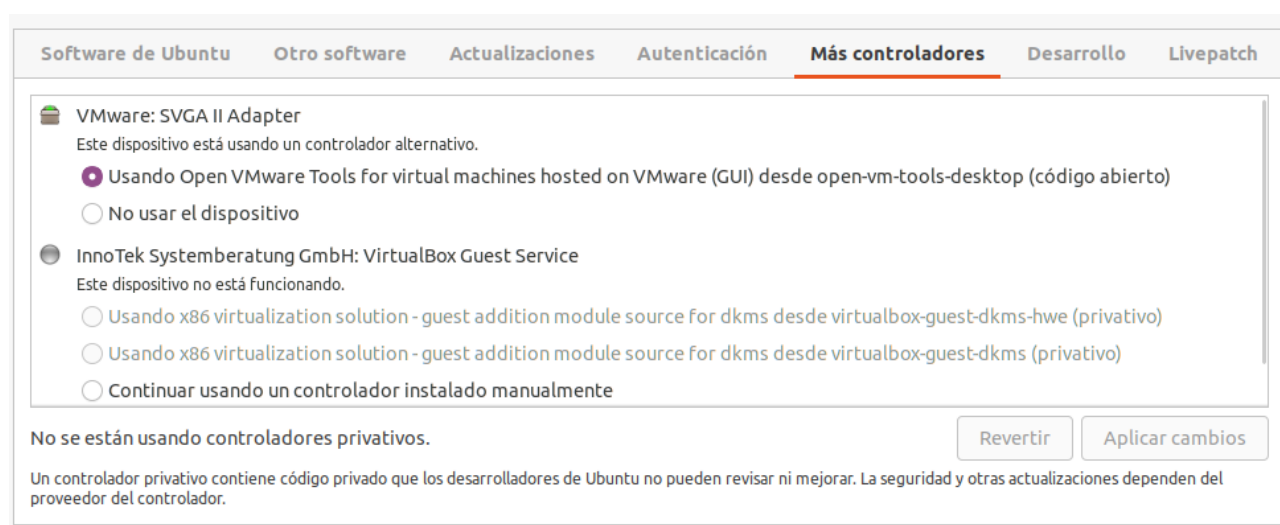
<https://www.flopy.es/como-interpretar-correctamente-la-carga-de-la-cpu-load-average-en-equipos-linux/>

## 7. Gestión de E/S

Al igual que Windows, Linux gestiona el acceso a la entrada/salida mediante los drivers.

Trata a los drivers como si fuese una aplicación más, descargándolos desde los mismos repositorios que el resto del software. De este modo, cuando comprobemos las actualizaciones del sistema mediante **Actualización de software**, ya estaremos comprobando las actualizaciones de los drivers.

Además, en **Software y actualizaciones** → **Más controladores**, podremos ver si se está usando algún driver especial:



Es importante tener actualizados los repositorios de paquetes, de modo que nuestros controladores estén actualizados, tal y como vimos en la actividad de actualización de software.

Para instalar un controlador nuevo, podemos usar los métodos vistos en la actividad de instalación de software:

- mediante apt-get install
- manualmente a partir de los fuentes con
  - ./config
  - make
  - make install

Para comprobar el hardware que hay instalado tenemos varias opciones:

Listado general del hardware	<b>lshw [ &gt; hardware.txt ]</b>
Información de CPU	<i>cat /proc/cpuinfo</i> <i>lscpu</i>
Información de memoria	<i>lshw -c memory</i>
Información de discos	<i>fdisk -l</i>
Información de buses PCI y dispositivos conectados	<i>lspci</i>
Información de dispositivos USB en el sistema	<i>lsusb</i>

En Linux, los dispositivos de entrada/salida se manejan con un fichero en */dev*. Esto permite que se lea/escriba en un dispositivo externo como si fuesen ficheros. En ese directorio encontraréis ficheros:

- con una “b” delante: dispositivos de bloque. Por ejemplo, discos duros.
- con una “c” delante: dispositivos de carácter. Por ejemplo monitores o impresoras.

Por ejemplo:

- */dev/sda*: Primer disco duro
- */dev/tty*: terminales

En caso de estar trabajando en entorno gráfico hay dos paquetes que podemos instalar y nos permitirán visualizar la información:

- *hardinfo*
- *lshw-gtk*

## 8. Archivos y directorios

Si estamos trabajando con GNOME, a la derecha tenemos el acceso al explorador de archivos que nos permitirá trabajar de un modo similar a Windows.



En caso de trabajar con el terminal, opcional en Desktop y necesario en Server, los comandos son los indicados en el apartado 3.

Simplificando mucho, en Linux el raíz del árbol comienza en “/”. Veremos la estructura de filesystems en una unidad posterior. En Linux el filesystem sería el equivalente al volumen (C:, D:) en Windows.

Por línea de comandos podemos ver los filesystems existentes ejecutando el comando:

**df -k**

La salida de este comando la comprenderás mejor cuando trabajemos la unidad de gestión de información.

## 9. Variables de entorno

Al igual que en Windows, Linux trabaja con variables de entorno. Se trata de variables que almacenan información usada por los diferentes programas.

Podemos ver las diferentes variables definidas ejecutando el comando **env**.

Para definir una nueva variable de entorno, o modificar una existente podemos usar el siguiente comando:

**export <VAR>=<VALOR>**

Accedemos a las variables definidas anteponiendo un "\$".

Por ejemplo, para incluir el directorio actual de trabajo en el PATH (no recomendado), ejecutaríamos:

**export PATH="./:\$PATH"**

OJO: con este comando no hacemos definitivo el cambio. Cuando abramos un nuevo terminal no lo mantendremos.

Si queremos hacerlo permanente, y trabajamos con la shell bash, podemos añadir la línea al final del fichero **~/bashrc**

Si lo queremos configurar para cualquiera de nuestras shells, lo añadimos en **~/profile**

Variables de entorno importantes:

COMANDO	DESCRIPCIÓN
<b>\$PATH</b>	Camino de búsqueda de órdenes.
<b>\$HOME</b>	Directorio de trabajo del usuario.
<b>\$USER</b>	Usuario que estableció la sesión.
<b>\$PWD</b>	Ruta completa del directorio de trabajo actual.
<b>\$TERM</b>	Tipo de terminal.
<b>\$SHELL</b>	Shell que está ejecutándose.
<b>\$PS1</b>	Prompt.

Para ver el contenido de una variable de entorno, podemos ejecutar:

**echo \$<VAR>**

## 10. El registro en Linux

En Linux no existe una base de datos de registro del estilo de Windows. Las diferentes configuraciones se almacenan en ficheros, en la mayor parte de las ocasiones en el directorio */etc*.

Los diferentes ficheros se editan con editores de texto (vi, nano, etc.).

Los mensajes del sistema y de las aplicaciones se vuelcan en ficheros de log, frente a los registros de sucesos de Windows que veremos en unidades posteriores. Es habitual que los diferentes ficheros de log se generen en el directorio */var*. Por ejemplo, los logs del sistema se guardan en */var/log*.