

# Sistemas Informáticos

---

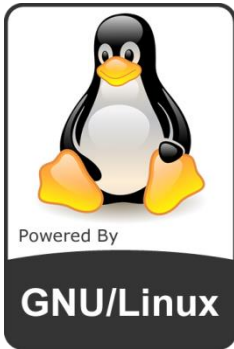
UD 7. Introducción de GNU/Linux  
Ubuntu 20.04. Tareas básicas



# ÍNDICE

1. Introducción .....	3
2. Breve repaso de la historia de los sistemas operativos GNU/Linux.....	3
3. Evolución de los sistemas operativos GNU/Linux: distribuciones .....	4
4. Características generales de los actuales sistemas operativos GNU/Linux .....	8
5. Estructura de un sistema operativo GNU/LINUX .....	9
6. Instalación de Ubuntu 20.04 LTS .....	10
6.1. Motivos por los que instalar esta distribución .....	10
6.2. Requisitos de hardware de Ubuntu 20.04 Desktop.....	10
6.3. Revisión de la compatibilidad hardware .....	10
6.4. Datos de la máquina virtual a crear.....	11
6.5. Proceso de instalación de Ubuntu Desktop .....	11
6.6. Particiones .....	15
7. Ajustes tras la instalación de Ubuntu .....	18
7.1. Habilitar el usuario root .....	18
7.2. Configurar las actualizaciones del sistema .....	18
7.3. Cambiar el nombre del equipo .....	20
7.4. Revisar los ajustes para la fecha y hora del sistema.....	20
8. Interfaz gráfica .....	21
8.1. Elementos del escritorio de Ubuntu.....	22
8.2. Personalización del escritorio.....	25
9. Sistema de archivos en Linux .....	26
9.1. Tipos de sistemas de archivos .....	26
9.2. Características generales de los sistemas de ficheros de Linux .....	27
9.3. Directorios importantes de Linux .....	28
9.4. Tipos de archivos .....	29
9.5. Ficheros y directorios ocultos.....	29
9.6. Los inodos o nodos-i:.....	30
9.7. Operaciones con ficheros y directorios en modo gráfico.....	31
10. Interfaz textual: shell .....	32
10.1. Comandos.....	32
10.2. Shell-Scripts de Linux.....	35
11. Permisos y atributos .....	40
12. Aplicaciones .....	43
12.1. Agregar o quitar programas .....	43
12.2. Asociar una aplicación a un tipo de archivo .....	45
12.3. Instalación de entornos de escritorio en Linux .....	45
13. ANEXOS .....	47
13.1. Instalar Guest Additions en Ubuntu 20.04 .....	47

## 1. INTRODUCCIÓN



A lo largo de este tema, estudiaremos las características del SO **GNU/Linux**, centrándonos especialmente en la distribución **Ubuntu**, por tratarse de una de las distribuciones más utilizadas actualmente ([www.distrowatch.org](http://www.distrowatch.org)). No obstante, todo lo que aprendamos, nos servirá para las demás distribuciones.

## 2. BREVE REPASO DE LA HISTORIA DE LOS SISTEMAS OPERATIVOS GNU/LINUX

Fue en el año 1991 cuando un estudiante, **Linus Benedict Torvalds**, creó el **núcleo de Linux** a partir de **UNIX** (SO creado por Ken Thompson y Dennis Ritchie de laboratorios Bell de AT&T en 1969) y tomando también ideas de **Minix** (versión de UNIX del creada por un profesor, Andrew S. Tanenbaum), pero con un núcleo monolítico.



Ken Thompson y Dennis Ritchie, creadores de Unix.



Andrew S. Tanenbaum, creador de Minix

Linus difundió el código fuente de esta primera versión del núcleo de Linux de manera gratuita por internet y fueron muchos los que se interesaron por él, especialmente **Richard Stallman del proyecto GNU** (GNU is Not Unix) quien propuso unir esfuerzos. De este modo, en 1992 **se unieron** las **aplicaciones** del sistema **GNU con el núcleo de Linux** creando así un SO completo y de código fuente libre, llamado **GNU/Linux**.



Linus Benedict Torvalds



Richard Stallman

GNU/Linux, como **software libre**, permite a todos los usuarios utilizar, copiar, modificar y redistribuir el código fuente libremente bajo los términos de la licencia **GNU GPL** (los trabajos derivados sólo pueden ser distribuidos bajo los términos de la misma licencia).

# GNU/Linux

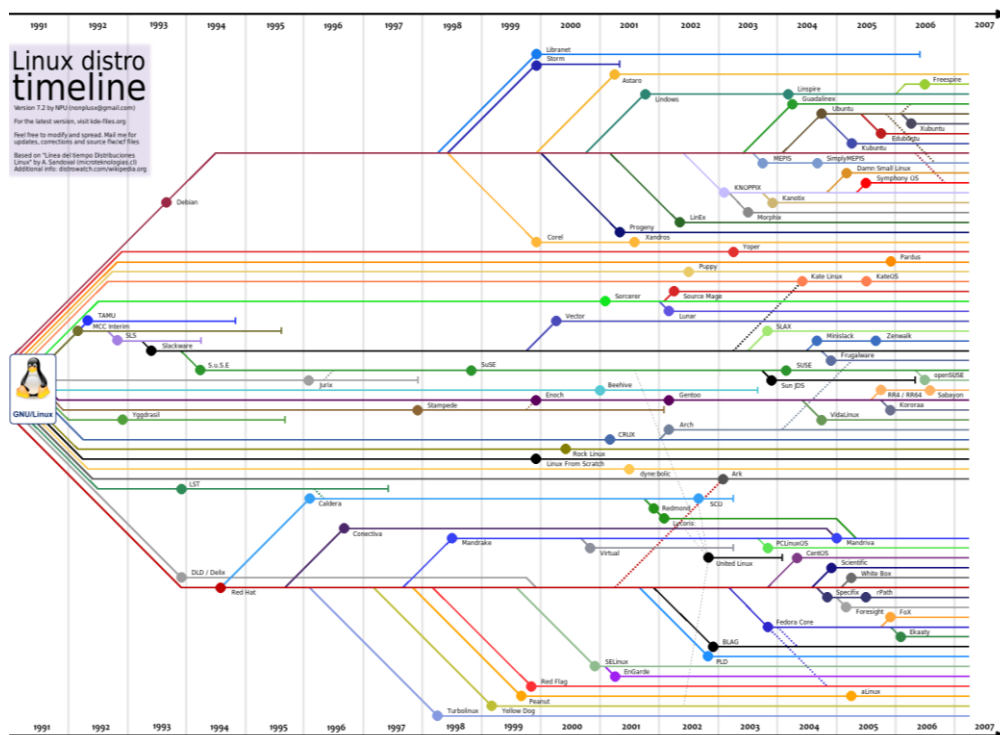


### 3. EVOLUCIÓN DE LOS SISTEMAS OPERATIVOS GNU/LINUX: DISTRIBUCIONES

Las **distribuciones GNU/Linux** son paquetes de software que incluyen:

- **El núcleo** o kernel de Linux
- **Aplicaciones GNU**
- **Instalador de la distribución**

De este modo, existen múltiples distribuciones que incluyen distintos paquetes software destinados a satisfacer las necesidades de una amplia gama de usuarios en función del uso que vaya a hacer del sistema operativo que instale en su ordenador. Algunos ejemplos son: Ubuntu, Mint, Fedora, Debian, Mageia, Mandriva, OpenSUSE... Además, cada una de estas distribuciones puede tener, a su vez, varias versiones, como en el caso de Ubuntu, donde podemos encontrar las versiones Kubuntu, Xubuntu, Lubuntu....



Puedes ver un gráfico actualizado de las distribuciones en:

[http://upload.wikimedia.org/wikipedia/commons/1/1b/Linux\\_Distribution\\_Timeline.svg](http://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg)

A continuación, se presenta algunas características de las distribuciones más utilizadas a día de hoy:

#### 1. Mint



Distribución basada en Ubuntu. Linux Mint mantiene un inventario actualizado, un SO estable para el usuario medio, con un fuerte énfasis en la usabilidad y facilidad de instalación. Es reconocido por ser fácil de usar, especialmente para los usuarios sin experiencia previa en Linux.

**Página Web de Linux Mint:**

<http://www.linuxmint.com/index.html>



## 2. Ubuntu



Distribución basada en Debian, centrada en el usuario final y fácil de usar. Muy popular y con mucho soporte en la comunidad. El entorno de escritorio por defecto es GNOME.

**Página Web de Ubuntu en Español:**

<http://www.ubuntu-es.org/>



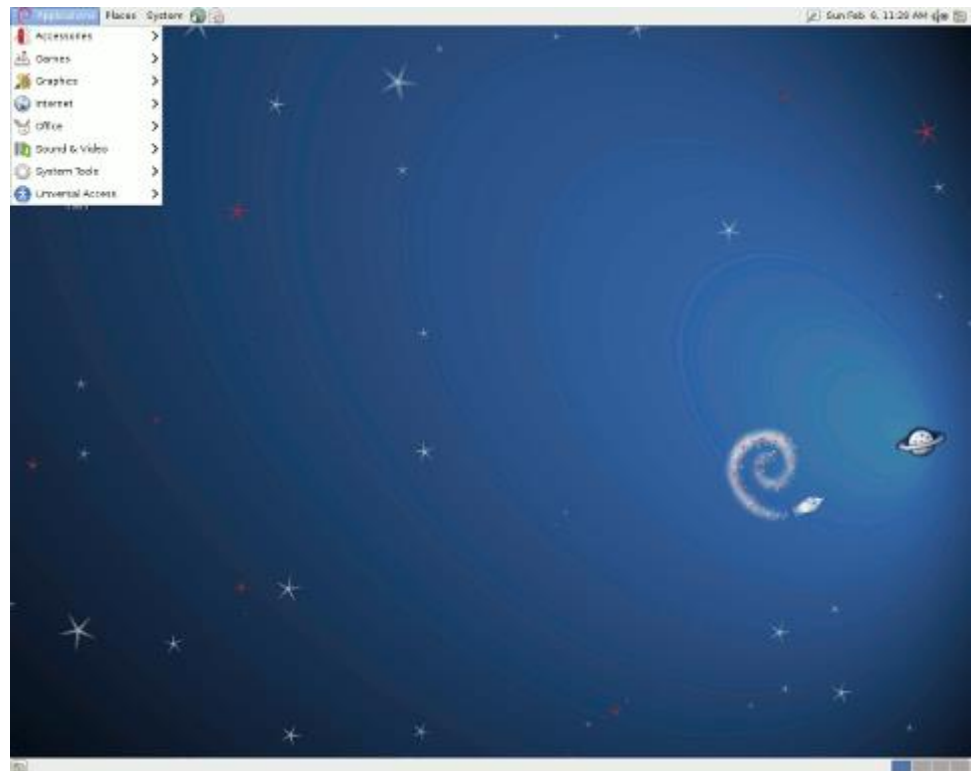
## 3. Debian



Distribución de muy buena calidad. El proceso de instalación es quizás un poco más complicado, pero sin mayores problemas. Gran estabilidad antes que últimos avances.

**Página Web de Debian en Español:**

<http://www.es.debian.org/>

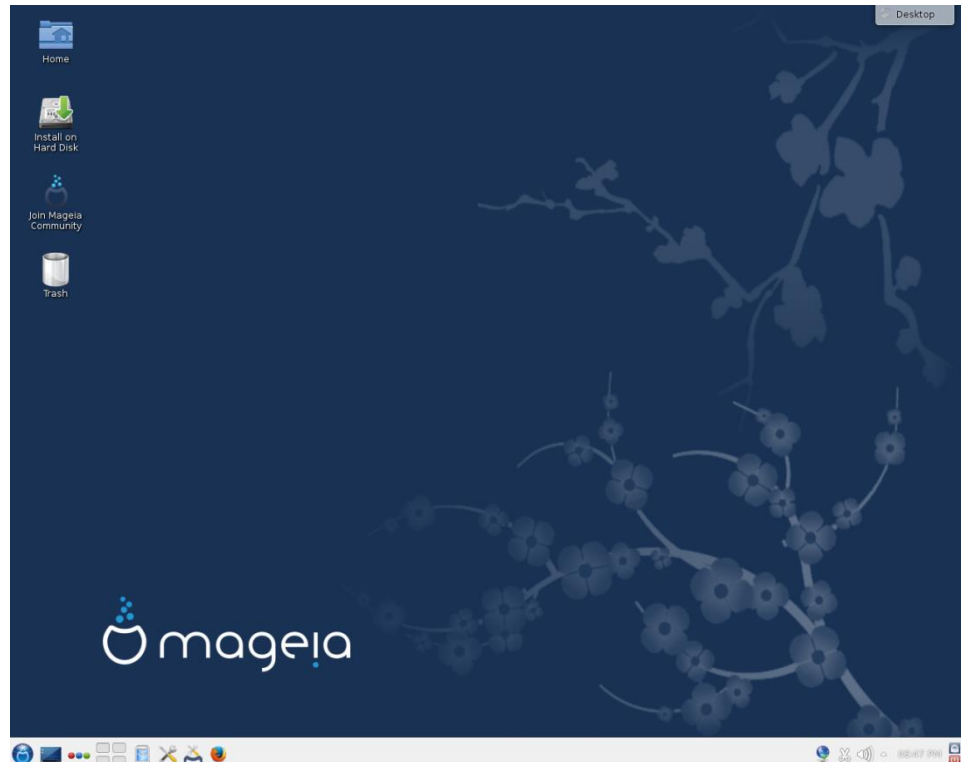


#### 4. Mageia



Mageia es un sistema operativo libre basado en GNU/Linux. Es un proyecto comunitario, soportado por una organización sin ánimo de lucro de contribuidores electos.

Página Web de Mageia en Español:  
<http://www.mageia.org/es/>

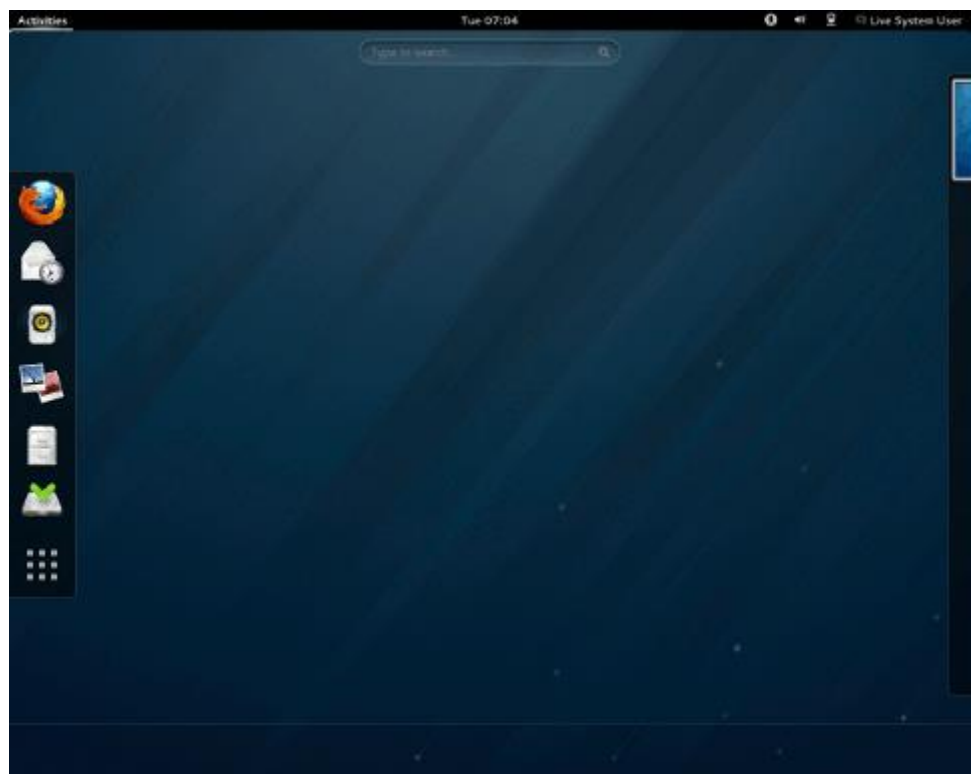


#### 5. Fedora



Esta es una distribución patrocinada por RedHat y soportada por la comunidad. Fácil de instalar y buena calidad.

Página Web de Fedora en Español:  
<http://fedoraproject.org/es/>





## 6. openSuse



Otra de las grandes.  
Fácil de instalar.  
Versión libre de la  
distribución comercial SuSE.

Página Web de OpenSuSE en Español:

<http://www.opensuse.org/>



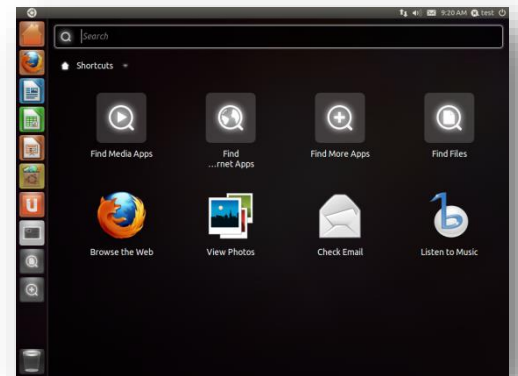
Como vemos, existen infinidad de distribuciones GNU/Linux. No obstante, observando las gráficas de evolución de las mismas, podemos observar cómo Debian, Slackware (SuSE), Red Hat (Fedora) y Ubuntu son las que han dado lugar a todas las demás.

El hecho de que existan muchas distribuciones y que sean libres no significa que sean gratis. Algunas distribuciones pueden cobrar el soporte oficial, el soporte software, documentación u otros servicios. Este es el caso de Redhat y SuSE Linux Enterprise, enfocadas a empresas y donde es necesario el pago de una licencia de soporte. De este modo, nunca pueden cobrar por el núcleo porque es libre, pero sí por algunas utilidades y aplicaciones.

## 4. CARACTERÍSTICAS GENERALES DE LOS ACTUALES SISTEMAS OPERATIVOS GNU/LINUX

A continuación, recordaremos las principales características:

- **Estructura:** Linux está dividido en **capas** y utiliza un **núcleo monolítico** (el núcleo se basa en el SO Unix y la última versión estable es el Kernel 5.15.7)
- **Software libre y código abierto:** Es un SO de libre distribución, por lo que se puede instalar y actualizar de forma gratuita. Además, también es de código abierto, por lo que podemos utilizar su código fuente para introducir modificaciones o mejoras.
- **Multiusuario, multitarea, multiprocesador y de tiempo compartido.**
- **Gestión de la memoria:** Utiliza **paginación** y como memoria virtual utiliza una **partición de intercambio** llamada **swap**.
- **Interfaz gráfica:** Con el paso de los años ha ido introduciendo notables mejoras en los interfaces gráficos de usuario. Además, cuenta con muchas distribuciones y gestores de ventanas para el entorno gráfico, por lo que el usuario puede elegir la distribución que más le convenga para sus necesidades o la que más le guste, donde se puede elegir qué gestor de ventanas se quiere utilizar.
- **Compatibilidad de Hardware:** Linux ha **mejorado** bastante en cuanto al reconocimiento del hardware sobre el que se instala y en cuanto a mejoras y novedades en el software que se puede instalar.
- **Hoy en día Linux es el SO más usado en servidores de alto rendimiento.**
- El administrador del sistema con “poderes” para hacer cualquier cosa en Linux es el usuario **root** (aunque en la distribución de **Ubuntu**, **root** está **deshabilitado** y el **usuario que se crea durante la instalación** es el que tiene privilegios de **administrador**).

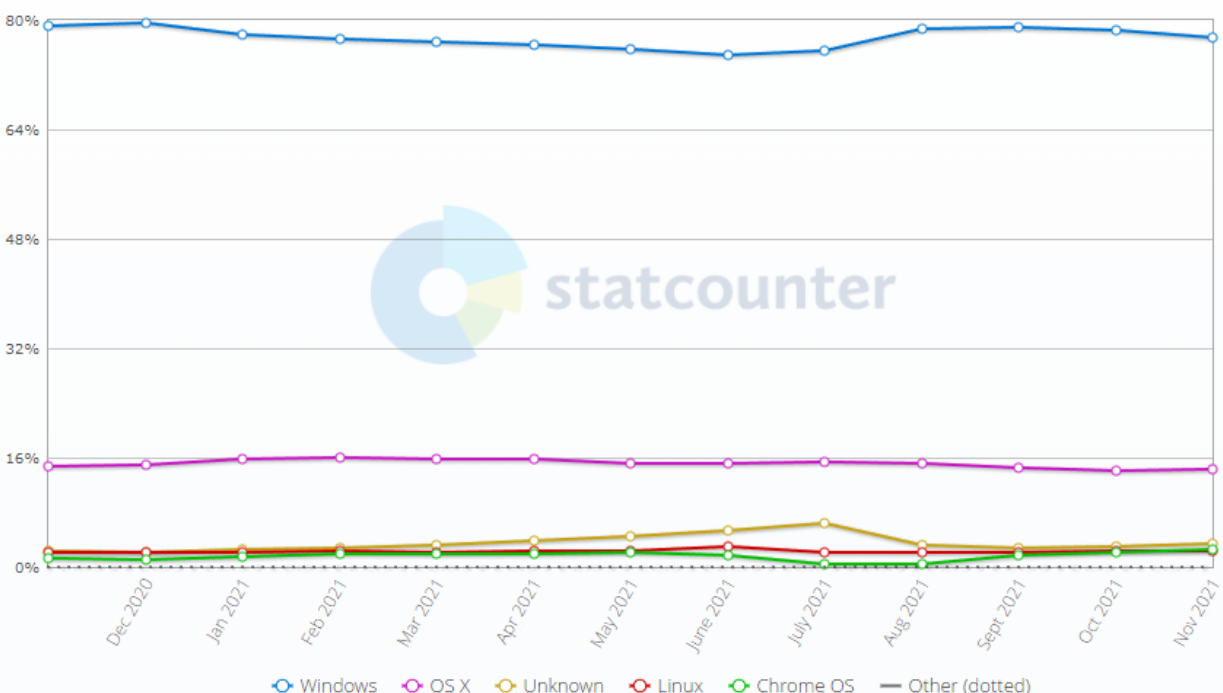


Estadísticas de uso de los SO en España en los últimos 12 meses

<https://gs.statcounter.com/os-market-share/desktop/spain/#monthly-202011-202111>

### Desktop Operating System Market Share Spain

Nov 2020 - Nov 2021



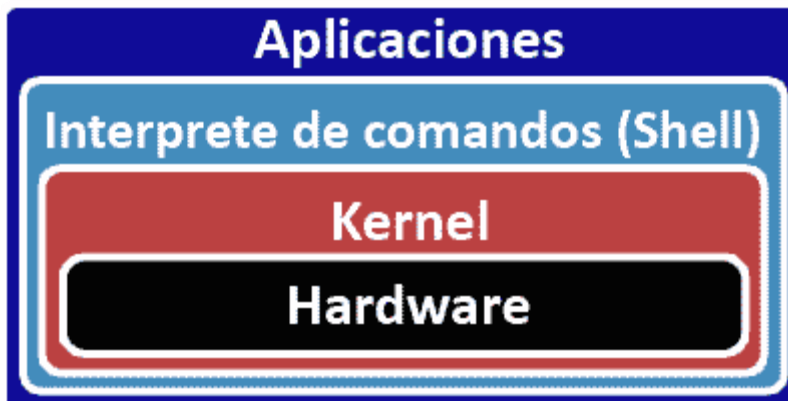


## 5. ESTRUCTURA DE UN SISTEMA OPERATIVO GNU/LINUX

Linux, además del núcleo que ejecuta los servicios más básicos del sistema operativo, tiene un proceso llamado Shell que aísla al usuario del núcleo e interpreta las órdenes o aplicaciones del usuario, bien desde una terminal en modo texto o desde un entorno gráfico, traduciéndolas a instrucciones que el SO entienda.

De este modo, podríamos decir, que Linux consta de los siguientes componentes:

- **Núcleo o Kernel:** es la parte del sistema que interactúa con el HW. Administra todos los recursos HW del sistema, como la memoria, el procesador, los periféricos... Actualmente, la última versión estable de kernel es la 5.15.7  
Donde:
  - o 5 es la versión del núcleo (varía cuando hay una gran modificación en el código del núcleo)
  - o 15 es la principal revisión del núcleo
  - o 7 hace referencia al número de revisión menor (varía cuando se incluye algunos nuevos drivers o algunas características nuevas)
- **Shell:** es un programa que se ejecuta al arrancar el sistema y que sirve para introducir órdenes, ejecutar programas. Es el equivalente al intérprete de comandos de Windows
- **Aplicaciones de los usuarios**



## 6. INSTALACIÓN DE UBUNTU 20.04 LTS

### 6.1. MOTIVOS POR LOS QUE INSTALAR ESTA DISTRIBUCIÓN

Actualmente, Ubuntu, como ya se comentó anteriormente, es una de las distribuciones más descargadas y además, es una de las que más derivadas o “distribuciones hijas” está desarrollando.

Además, el motivo de utilizar esta versión en concreto, es decir, la **20.04 LTS** es porque la 20.04 es la **última versión LTS**. Las versiones LTS (Long Term Support), se liberan cada dos años y reciben **soporte durante cinco años**. Sin embargo el soporte podría llegar hasta los 10 años a través del servicio ESM (Extended Security Maintenance) como parte de la suscripción UA-I (Ubuntu Advantage for Infrastructure), poniéndose a la par con los que ofrece Red Hat y SUSE.

Página de descarga de las ISOs de Ubuntu: <https://ubuntu.com/download/desktop>

Enlace directo a la descarga de la versión 20.04 Desktop: <https://ubuntu.com/download/desktop/thank-you?version=20.04.3&architecture=amd64> (todas las versiones Server y Desktop en <http://releases.ubuntu.com/20.04/> )

### 6.2. REQUISITOS DE HARDWARE DE UBUNTU 20.04 DESKTOP

Según <https://help.ubuntu.com/community/Installation/SystemRequirements/> los **requisitos mínimos “recomendados” para la edición “desktop”**, teniendo en cuenta los efectos de escritorio, son:

- Procesador 2 GHz (64 bits).
- Memoria RAM de 4 GB.
- Disco Duro de 25 GB (swap incluida).
- Tarjeta gráfica y monitor con resolución de 1024x768. Además de lector de CD, puerto USB o tarjeta de red.

Por otro lado, según <https://ubuntu.com/server/docs/installation> los **requisitos mínimos “recomendados” para la Ubuntu 20.04 Server Edition** (sin entorno de escritorio) son:

CPU	RAM	Espacio en disco duro
1 GHz	1GB	2.5 GB

### 6.3. REVISIÓN DE LA COMPATIBILIDAD HARDWARE

Disponer de controladores para todos los dispositivos HW supone cada vez un menor problema para las distribuciones Linux. Cada vez son más los drivers soportados por el kernel oficial de Linux.

Concretamente, Ubuntu, dispone de un programa de certificación de HW, de modo que para saber si un equipo o dispositivo es compatible podemos consultar las siguientes listas:



- Lista de equipos por fabricante y versión de la distribución para los que se certifica que funciona correctamente: [www.ubuntu.com/certification](http://www.ubuntu.com/certification)
- Lista de dispositivos SoC (System on a Chip ) que funcionan en el sistema: <https://certification.ubuntu.com/soc>

De todos modos, aunque nuestro equipo o componente HW no aparezca en estas listas, no significa que no haya un controlador para Ubuntu, ya que la comunidad desarrolla drivers por su cuenta. Por otro lado, también existe HW que no ha sido enviado a Canonical por el fabricante para ser certificado, pero sí que ha sido más que testeado por los miembros de la comunidad. Por ejemplo, en los foros de [ubuntuforums.org](http://ubuntuforums.org) puede ser de gran ayuda en caso de duda con algún HW.

## 6.4. DATOS DE LA MÁQUINA VIRTUAL A CREAR

Datos de la máquina virtual a crear:

- **RAM:** 1,5-2GB
- **Disco duro:** 20 GB

## 6.5. PROCESO DE INSTALACIÓN DE UBUNTU DESKTOP

Puedes ver el proceso de creación de una MV e instalación de Ubuntu 20.04 en: <https://youtu.be/OJI7hRH0HTs>

La instalación de Ubuntu Desktop no tiene misterio: cogemos un CD/DVD de instalación de la versión Desktop que queremos instalar y seguimos los pasos de instalación (como los que se muestran más adelante).

Empezamos entonces el proceso de instalación:

Iniciamos el equipo con el CD de Ubuntu en la unidad CD/DVD-ROM (previamente la BIOS debe estar configurada para que arranque en 1er lugar desde el CD). *Nota: También podemos instalar Ubuntu desde un pendrive, por ejemplo.*

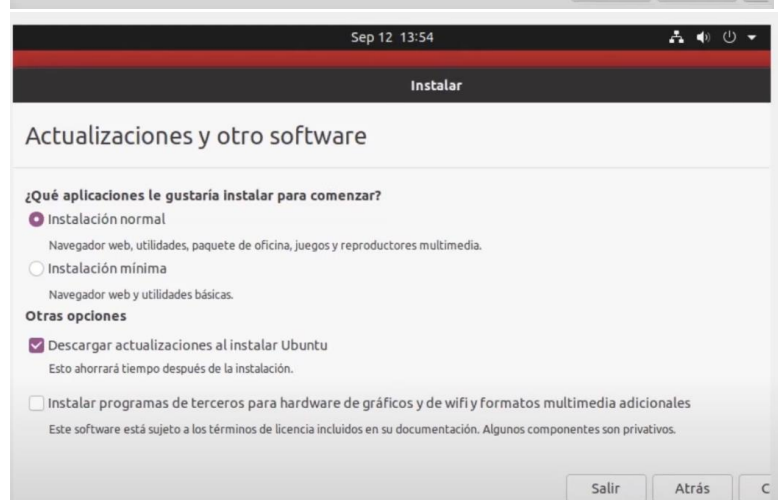
Esperamos a que el CD cargue...



Elegimos nuestro idioma y hacemos clic en “instalar Ubuntu” para continuar. Elegimos la disposición del teclado en español.

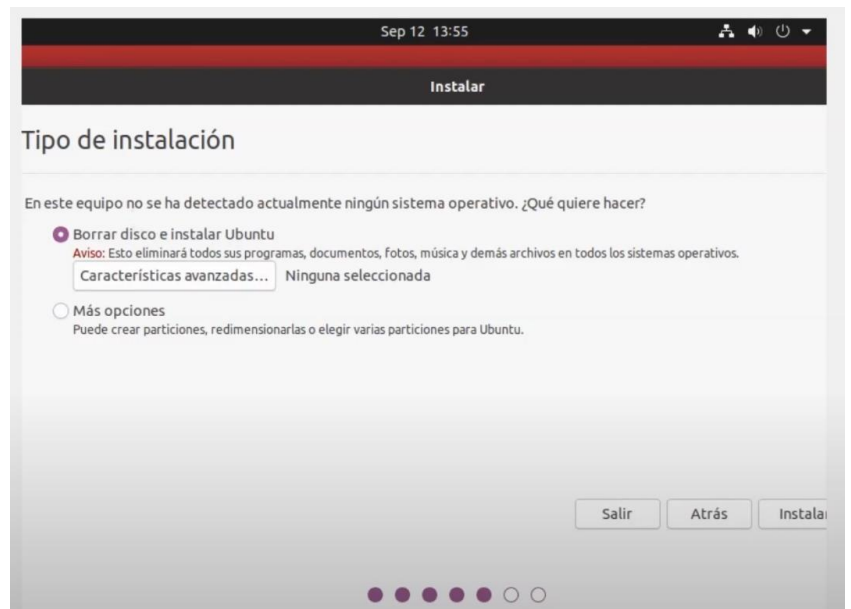


Preparando la instalación de Ubuntu. Podemos optar por descargar actualizaciones durante la instalación o por instalar aplicaciones de terceros, como codecs MP3. Sin embargo, hay que tener en cuenta que si seleccionamos esas opciones, todo el proceso de instalación durará más tiempo. Hacemos clic en Continuar.

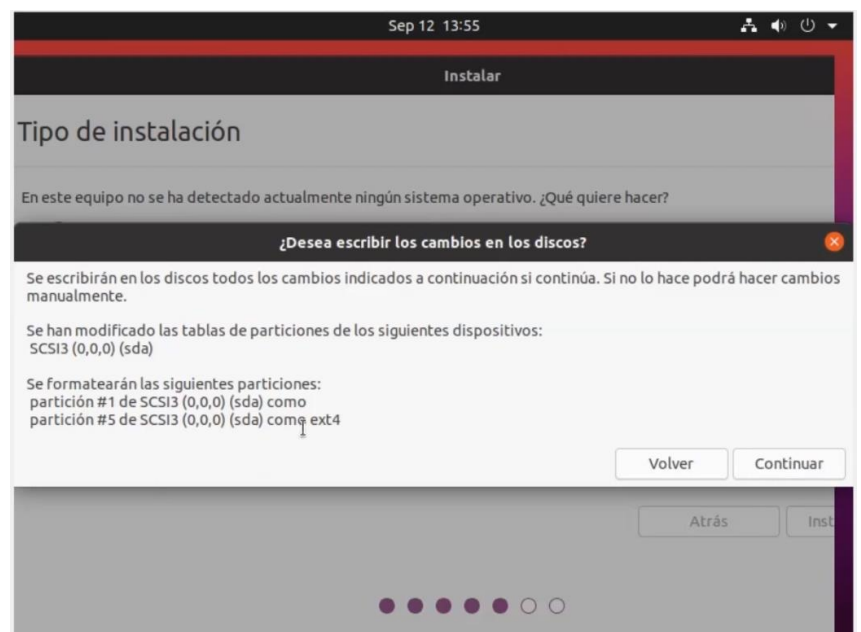


Elegimos el tipo de instalación:

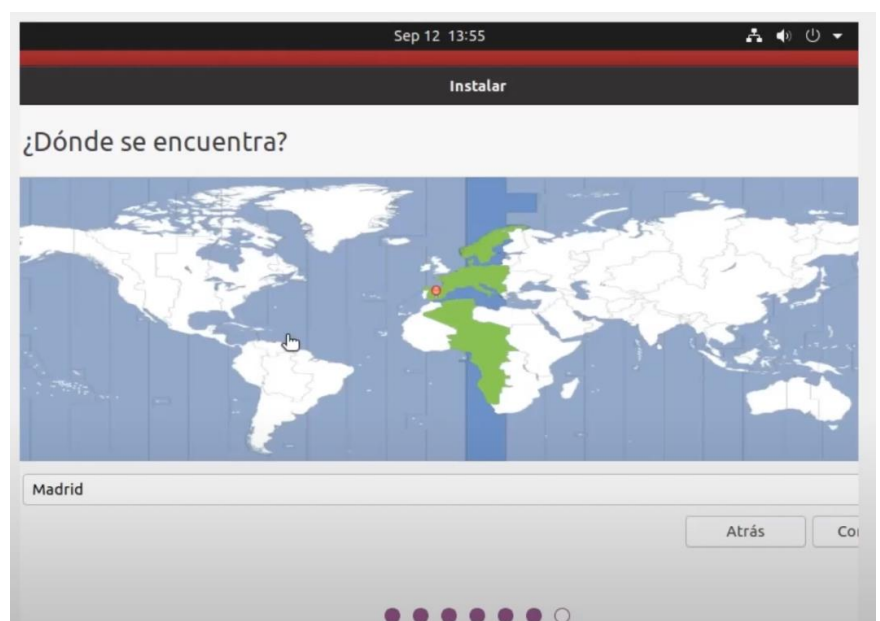
- Borrar disco e instalar Ubuntu: Elige esta opción si quieres eliminar tu actual sistema operativo, o si el disco duro ya está vacío y quieres que el instalador particione automáticamente el disco duro para ti. Esta es la opción recomendada especialmente para aquellos que quieran un equipo con un solo sistema operativo. En nuestro caso, elegimos esta opción y hacemos clic en continuar.
- Más opciones: permite crear particiones especiales o formatear el disco duro con otro sistema de archivo diferente del predeterminado.



Tras la elección del tipo de instalación, haremos clic en “Instalar ahora” y en “Continuar”.



Elegimos la zona en la que nos encontramos. En el caso de España, debemos elegir Madrid.



Introducimos nuestro nombre, el nombre del ordenador (generado automáticamente, pero puede modificarse), el nombre que deseamos usar para iniciar sesión en Ubuntu (nombre de usuario) y la contraseña.

En este paso, también hay una opción llamada “Iniciar sesión automáticamente”. Si marcamos esta casilla, el sistema operativo accederá por sí sólo al escritorio de Ubuntu.

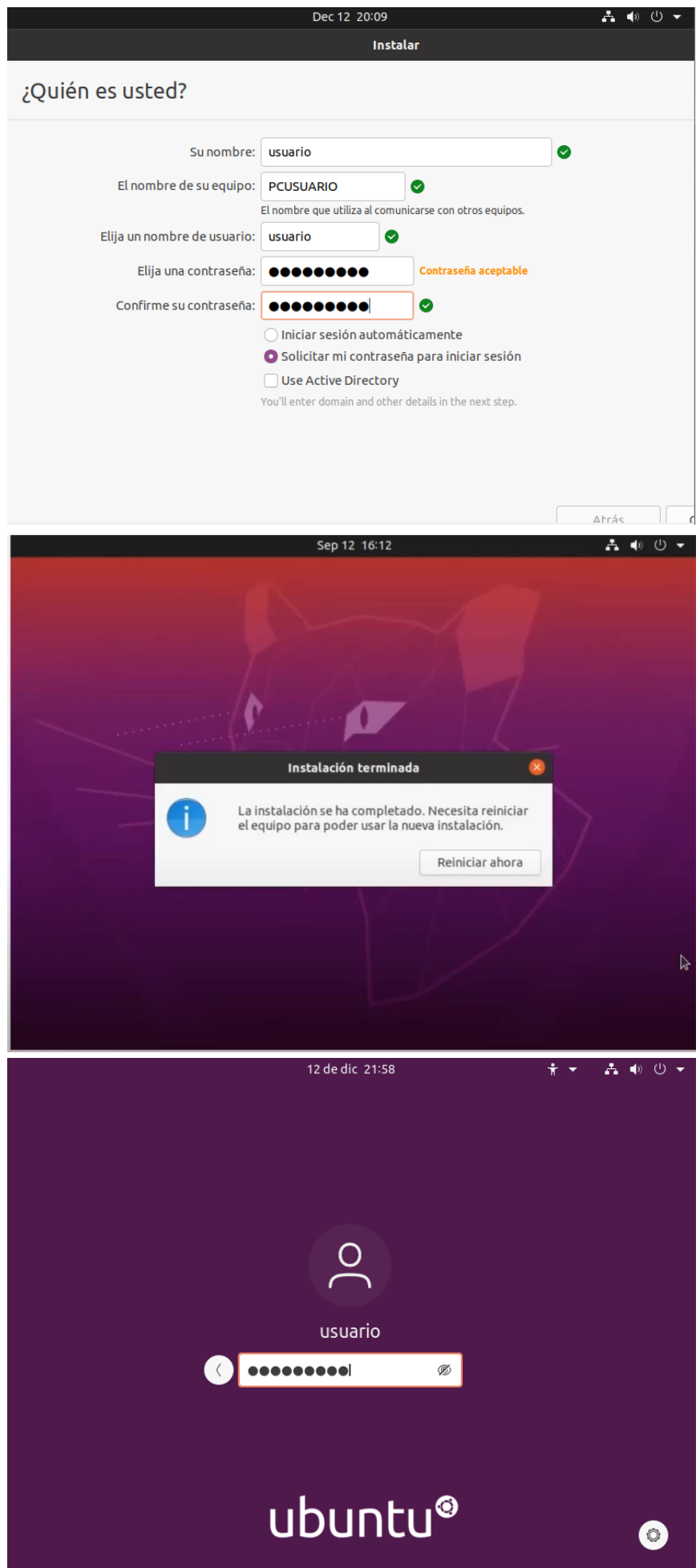
Hacemos clic sobre el botón “Continuar” para seguir...

En este momento, se instala el sistema, se copiarán los archivos y se configurará el sistema y el hardware.

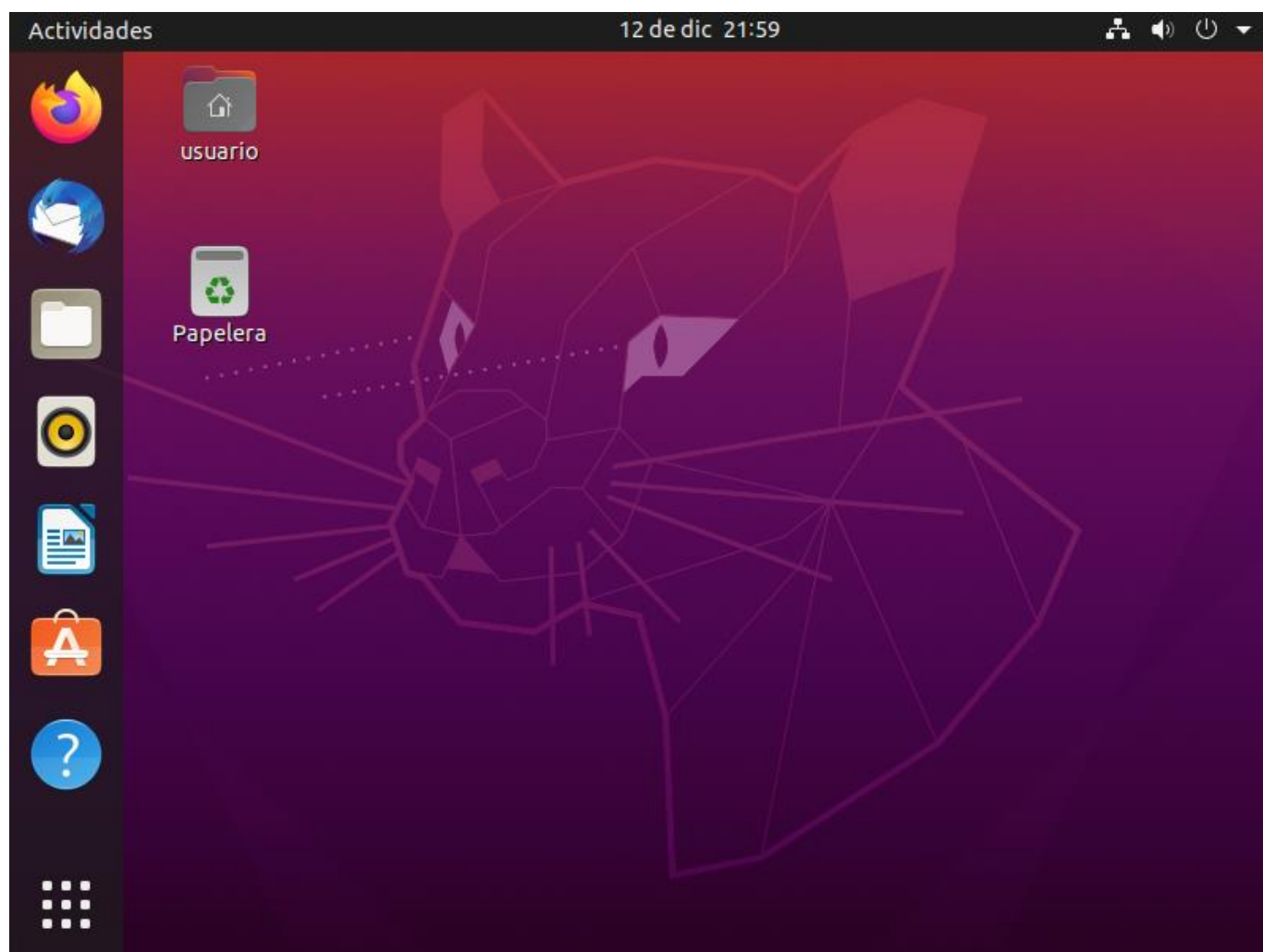
Tras unos minutos aparecerá una ventana emergente, para notificarnos que la instalación ha finalizado y tendremos que reiniciar el ordenador para usar el nuevo Ubuntu. Hacemos clic sobre el botón “Reiniciar ahora”.

Retiramos el CD o el dispositivo desde donde hemos instalado Ubuntu y pulsamos la tecla “Enter” para que se reinicie.

El ordenador será reiniciado y, en algunos segundos, veremos la pantalla de inicio. Hacemos clic sobre nuestro nombre de usuario e introducimos la contraseña.



¡Ya está! Ya podemos disfrutar de Ubuntu.





## 6.6. PARTICIONES

### 6.6.1. Cuántas particiones requiere Ubuntu

Ubuntu requiere como mínimo dos particiones:

- Una para el sistema, llamada raíz (/)
- Otra llamada swap o de intercambio que constituye una zona de memoria virtual en disco duro.

A partir de ahí, el usuario puede crear tantas particiones como considere necesarias. Por ejemplo, podríamos crear 3 particiones:

- Una partición raíz (/) para el sistema con los directorios y archivos del SO y los programas que instalemos
- Una partición swap
- Una partición home (/home) donde se guardarán los directorios y archivos de los usuarios

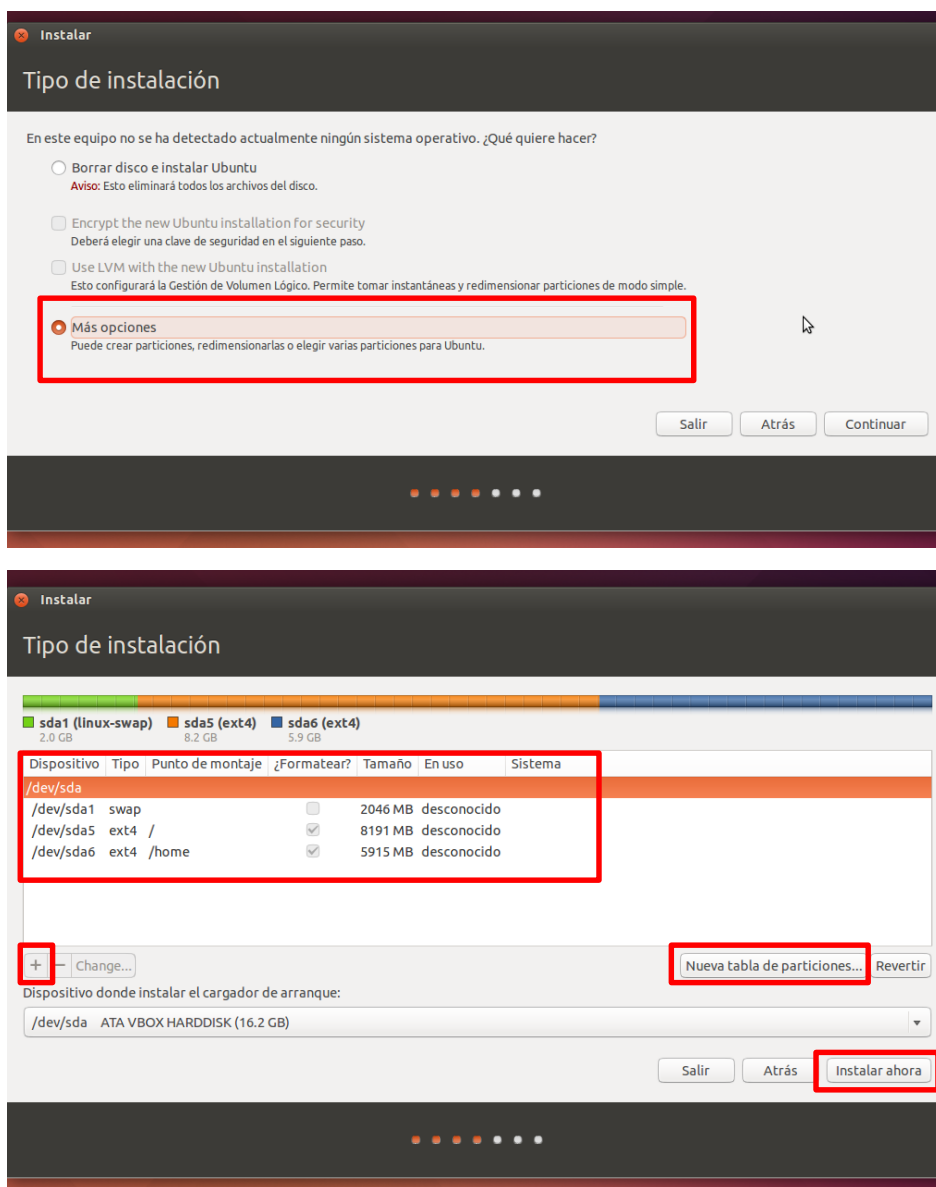
A continuación, se muestra cómo se crearía estas 3 particiones mediante la herramienta que integrada que trae el proceso de instalación:

### 6.6.2. Esquema de particiones: Utilización de la herramienta avanzada de particionado

El llamado “modo experto” de particionado, que en el asistente de instalación se encuentra tras la opción “Más opciones”, nos permite a nosotros mismos establecer el esquema de particiones del disco duro, eliminado, redimensionado y creando las particiones necesarias.

La barra horizontal representa el disco duro y el código de colores indica cada una de las particiones que existen en el disco.

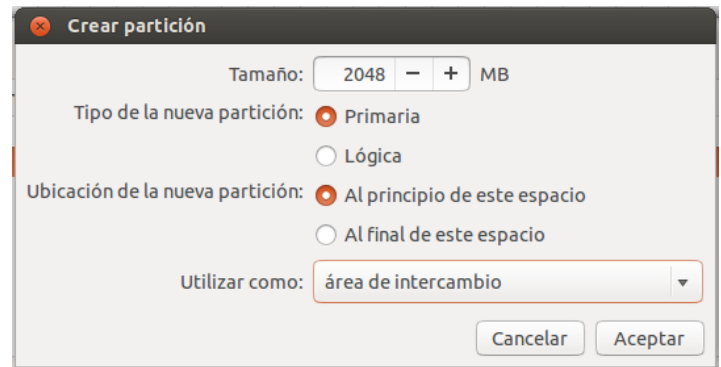
De este modo, crearemos una partición swap o de intercambio, la partición raíz del sistema y la partición home. Al final, nos quedarán las particiones del siguiente modo:



*Nota: En este ejemplo, como estamos partiendo de un sistema completamente nuevo, las particiones se formatearán de forma obligatoria. Sin embargo, cuando estemos reinstalando el sistema sobre una instalación previa, podremos desmarcar cualquier partición para que no se formatee (normalmente, la partición /home). De esta forma, no perderemos los datos que contenga. Sí será obligatorio formatear la partición raíz (/).*

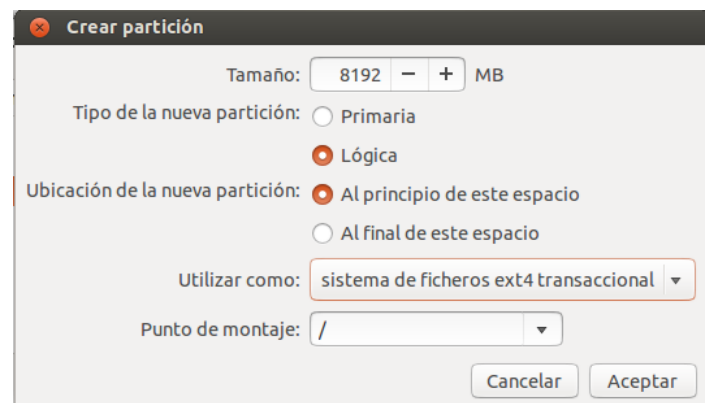
### Proceso de creación de la partición swap:

- Tipo de la nueva partición: Primaria
- Tamaño: 2GB (2048MB). Si el equipo dispone de menos de 2GB de RAM, el tamaño de la partición swap es recomendable que sea el doble de la RAM. Si se dispone de más memoria, basta con asignar a la swap el mismo tamaño que la memoria RAM.
- Ubicación de la nueva partición: al principio de este espacio
- Utilizar como: "área de intercambio".
- Punto de montaje: no es posible editar el punto de montaje puesto que la partición swap no contiene ningún directorio del sistema de ficheros de Linux



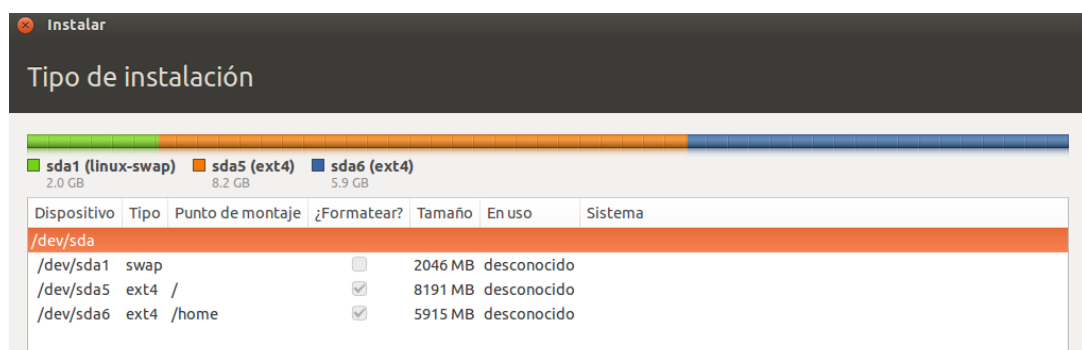
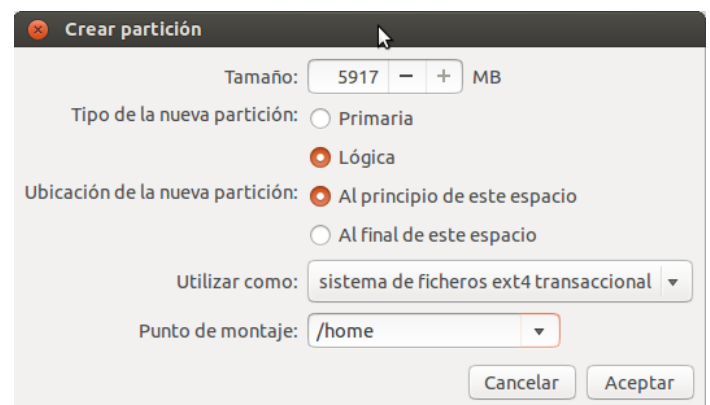
### Proceso de creación de la partición raíz:

- Tipo: Lógica. Puesto que en un sistema solo puede existir un máximo de 4 particiones primarias, Ubuntu sugiere que creamos particiones lógicas para no agotar el número de primarias innecesariamente.
- Tamaño: 8GB (8192MB).
- Ubicación: a continuación de la última creada (la swap), es decir, al principio de este espacio
- Utilizar como: seleccionamos el sistema de archivos con el que se formateará la partición, que será el sistema nativo de Ubuntu Linux, es decir, ext4
- Punto de montaje: el punto de montaje de una partición es idéntico al directorio del sistema de ficheros de Linux que contiene. Esta es la partición raíz, por lo que elegiremos como punto de montaje la raíz del sistema de ficheros, representada mediante el símbolo /



### Proceso de creación de la partición home:

- Tipo: lógica
- Tamaño: el espacio libre restante (unos 5GB)
- Ubicación: a continuación de la última creada (la raíz), es decir, al principio de este espacio
- Utilizar como: seleccionamos el sistema de archivos con el que se formateará la partición, que será el sistema nativo de Ubuntu Linux, es decir, ext4
- Punto de montaje: esta es la partición home, que contendrá los datos de los usuarios, por lo que elegiremos como punto de montaje /home

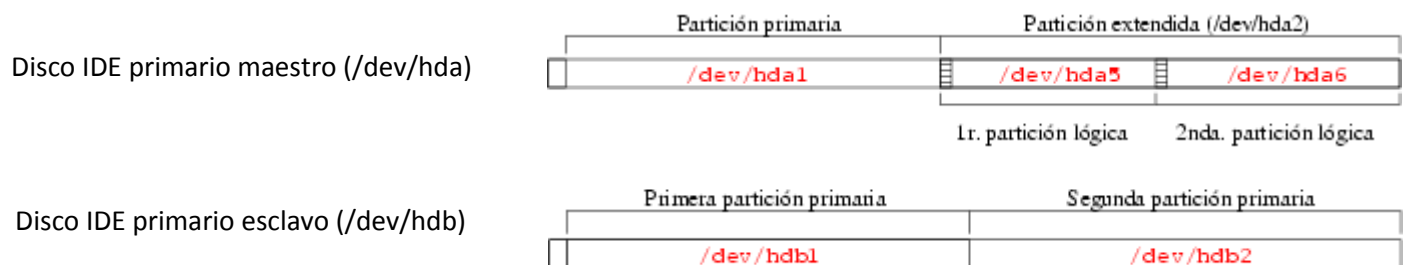


### 6.6.3. Nombrado de particiones en Ubuntu

Ejemplos de nombres de particiones de disco bajo Linux son: /dev/sda1, /dev/hda2, /dev/hdb3 o /dev/sdc5

A continuación, se explica esta nomenclatura:

- /dev corresponde al directorio del sistema de ficheros de Linux desde donde se accede a los dispositivos hardware.
- A continuación, aparece el tipo de disco:
  - o la letra “h” (hard) si el disco duro donde se encuentra la partición posee interfaz IDE
  - o la letra “s” si es de tipo “serial ATA o SCSI”.
- La letra “d” que forma el nombre “hd” o “sd” hace referencia a “drive (unidad)”.
- Tras “hd” o “sd” se incluye una letra del abecedario (comenzando por la “a”):
  - o El primer disco duro Serial ATA o SCSI se llamará sda, el segundo sdb y así sucesivamente.
  - o El disco duro maestro en el canal primario IDE es hda y el disco esclavo es hdb.
- Por último, un número indica el tipo de partición:
  - o Los números del 1 al 4 indican una partición primaria. Por ejemplo: hda1
  - o Del 5 en adelante, indican una partición lógica (dentro de una extendida). Por ejemplo: hda6



## 7. AJUSTES TRAS LA INSTALACIÓN DE UBUNTU

Tras la instalación de un sistema operativo, hemos de realizar ciertas tareas de configuración inicial. Lógicamente, estos ajustes estarán en función del cometido que vaya a tener el ordenador. En este caso, dado que el equipo hará funciones de servidor, las tareas que vamos a acometer son las siguientes:

- Habilitar el usuario root
- Configurar las actualizaciones del sistema
- Configurar la red (se explicará en el tema 11 de Sistemas Informáticos)
- Cambiar el nombre del equipo
- Establecer los ajustes adecuados para la fecha y hora del sistema

### 7.1. HABILITAR EL USUARIO ROOT

En general se desaconseja tener activada la cuenta root. No obstante, si deseamos activarla para poder iniciar sesión con ella, el truco es muy sencillo: sólo tenemos que asignarle una contraseña mediante el comando:

```
sudo passwd root
```

Tras escribir la contraseña de administración, el sistema nos pide la nueva contraseña para root. Como de costumbre, la contraseña se solicita por duplicado para evitar que se produzcan errores tipográficos.

Ahora si cambiarnos a una consola de texto (Alt + Ctrl + F1) e intentamos identificarnos como root veremos que el prompt (el texto que aparece delante del cursor cuando el sistema está esperando una orden) es ahora una almohadilla (#), en lugar del signo dólar (\$) que corresponde a un usuario normal. Además, la primera palabra es el nombre del usuario.

Recuerda: No es recomendable el uso de la cuenta root de forma continuada. Si la has activado, de forma temporal, para realizar tareas de administración, es muy aconsejable volverla a desactivar en cuanto sea posible. Para ello:

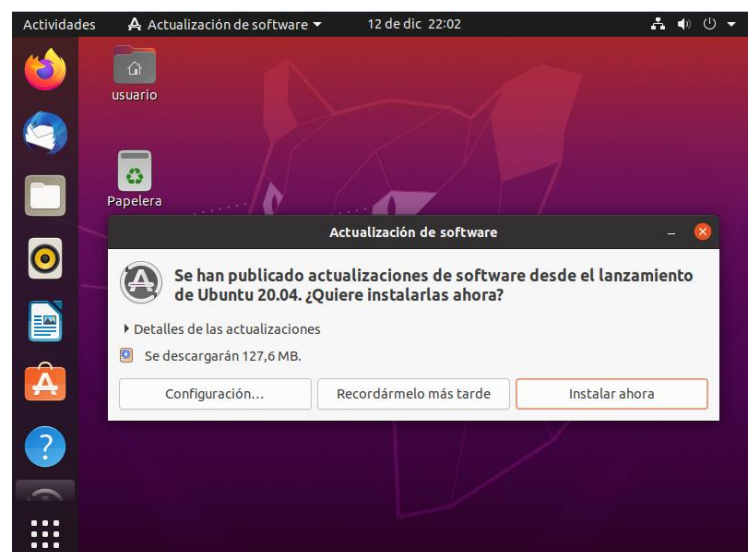
```
sudo passwd -l root
```

### 7.2. CONFIGURAR LAS ACTUALIZACIONES DEL SISTEMA

#### 7.2.1. Instalación de actualizaciones tras instalación del sistema

Tras la instalación de Ubuntu, cuando se inicia el sistema por primera vez, si el equipo dispone de conexión a Internet, nos indicará qué actualizaciones se encuentran disponibles para su descarga e instalación.

Dado que la tarea de actualización modifica la configuración del sistema, durante el proceso se solicitará que nos autentiquemos como root (superusuario) tecleando la contraseña del usuario que creamos al instalar el sistema. Esto es así, porque Linux no concede privilegios de administración a ninguna cuenta de usuario (incluida la que creamos al instalarlo) para evitar que malware pueda modificar nuestro sistema sin nuestro permiso. Por ello, obliga cada vez que queremos hacer un cambio en el sistema a adquirir privilegios de administración.

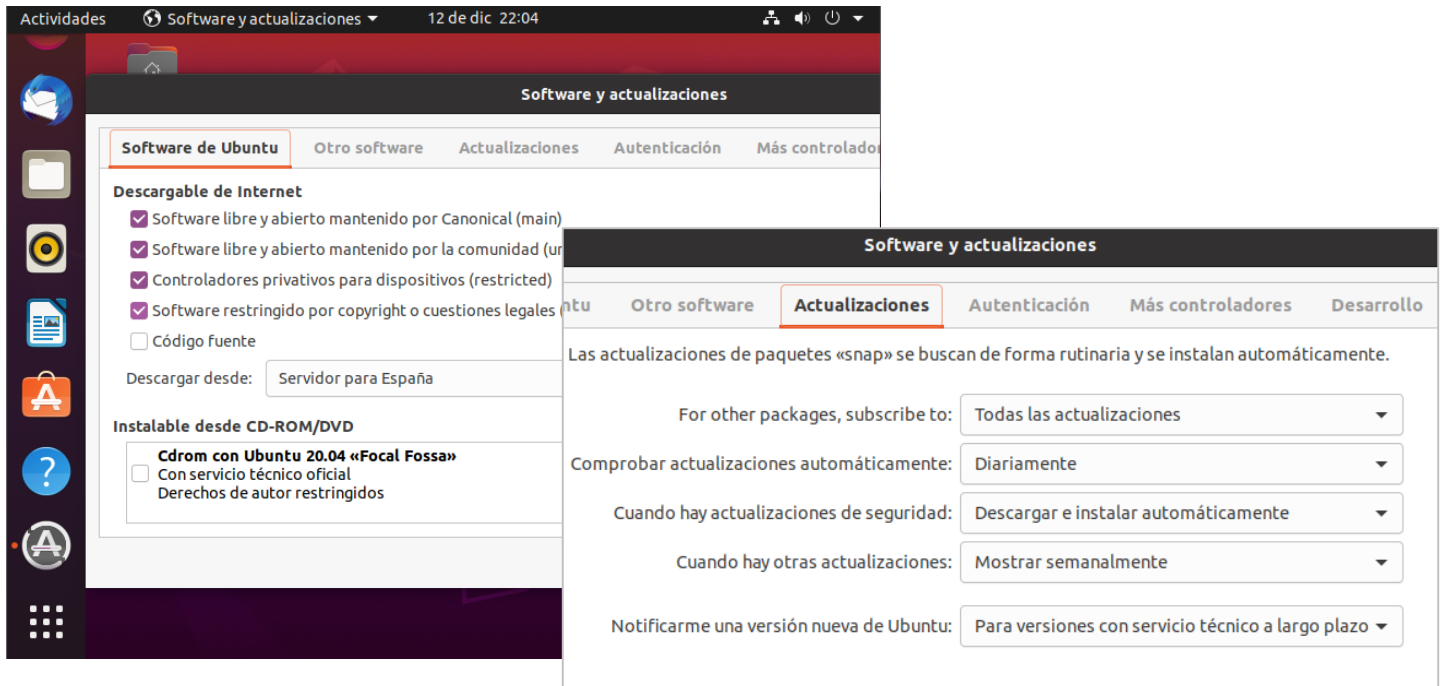
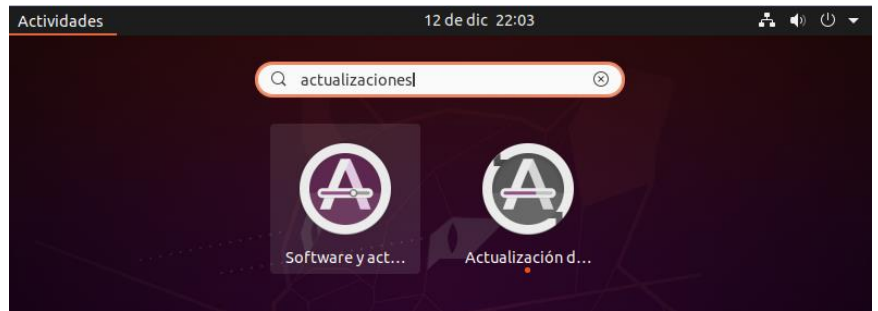


*Nota: recuerda que por motivos de seguridad y eficiencia hay que procurar mantener el SO y demás software que instalemos actualizado.*

### 7.2.2. Configuración de actualizaciones

Para configurar cuándo y cómo se actualiza el SO, hemos de ir a la herramienta “Software y actualizaciones”.

Una vez, aquí, podremos configurar, por ejemplo, los tipos de actualizaciones que deseamos que se instalen, así como la frecuencia con la que deberán comprobarse.



Para actualizar el sistema también podemos ejecutar el comando:

```
sudo apt-get update && apt-get upgrade
```

También podríamos crear una tarea con crontab para que las actualizaciones pendientes se descarguen y se instalen de forma automática todos los días a una hora en particular. Para ello ejecutamos:

```
sudo crontab -e
```

A continuación, escribimos la tarea programada:

```
00 3 * * * root apt-get update -y && apt-get dist-upgrade -y
```

Esto hará que todos los días, a las 3:00, se ejecuten con las credenciales de root dos comandos apt-get:

- El primero añadirá al sistema todas las actualizaciones que hayan aparecido en los repositorios desde la última actualización.
- La segunda comprobará si hay disponible una actualización general del sistema a la versión siguiente. Si la hay, la instala.

### 7.3. CAMBIAR EL NOMBRE DEL EQUIPO

Si deseamos cambiar el nombre que le dimos al equipo durante el proceso de instalación, hemos de editar un par de archivos del sistema:

Primero comenzaremos por el archivo `/etc/hostname`, que es el que guarda, en realidad, el nombre del equipo. Para ello ejecutaremos:

```
sudo nano /etc/hostname
```

y al abrirse el editor, cambiaremos el nombre que aparece por el nuevo nombre de equipo. A continuación, pulsaremos `Ctrl + X` y confirmaremos el guardado.

Al salir comprobamos que el prompt sigue mostrándonos el nombre antiguo. Sin embargo, si utilizamos el comando `hostname`, que sirve, precisamente, para mostrar el nombre del equipo, vemos que el nombre que devuelve es correcto. Esto ocurre porque la terminal no ha actualizado aún la información en el prompt. Para comprobarlo, cerramos la ventana de la terminal. Al volverla a abrir comprobamos que ahora el prompt ya es correcto.

Además del archivo `/etc/hostname`, hemos de cambiar el archivo `/etc/hosts`. Este archivo es el primer lugar donde busca el sistema para resolver un nombre de dominio, por lo que también aquí tenemos que cambiar la referencia que contiene a la dirección de loopback (la que se refiere al propio equipo). Para ello ejecutaremos:

```
sudo nano /etc/hosts
```

buscamos la referencia al nombre antiguo y la sustituimos por el nombre nuevo. A continuación, pulsaremos `Ctrl + X` y confirmaremos el guardado.

### 7.4. REVISAR LOS AJUSTES PARA LA FECHA Y HORA DEL SISTEMA

El reloj y el calendario del sistema vienen configurados de forma predeterminada para que no sea necesario realizar ningún ajuste importante. Asimismo, Ubuntu dispone de algunas características que pueden resultar muy útiles en ciertos momentos. Se accede a los ajustes de fecha y hora mediante la herramienta Configuración → Fecha y hora.



## 8. INTERFAZ GRÁFICA

Existen dos tipos de interfaces: modo texto y modo gráfico. La interfaz texto consiste en la consola o terminal a través de la cual se introducen comandos, mientras que la interfaz gráfica utiliza botones, iconos y ventanas.

El sistema gráfico en sistemas tipo Unix está basado en componentes o unidades independientes, algo así como piezas sueltas que pueden acoplarse unas a otras: primero tenemos el **servidor X** (X Window System: X.org, Xfree86), después están los **gestores o manejadores de ventanas** (Window Managers: Icewm, Fluxbox...) y por último, los **escritorios** (Desktops: KDE, Gnome; Xfce...).



De este modo, en Linux, la interfaz gráfica se carga gracias al **X-Windows System** (sistema de ventanas X) conocido como **X11** (la versión actual es la 11) o **X**, el cual define una serie de protocolos de comunicación y visualización de ventanas. El funcionamiento es del tipo cliente-servidor (un servidor X se comunica con varios programas cliente). El servidor X **controla los dispositivos periféricos como el teclado, el ratón, la pantalla** (hardware de vídeo). El servidor X se ejecuta cuando se arranca el sistema.

Por otro lado, el sistema gráfico X-Window de Linux permite al usuario escoger entre varios **gestores de ventanas** según sus gustos o necesidades. El gestor de ventanas **controla la ubicación y apariencia de las ventanas** (lo único que hace es dibujar las ventanas y presentarlas en pantalla (sin “adornos”), definen el tamaño y la posición, así como los elementos en ellas: iconos para cerrar, maximizar, etc. y, por tanto, permite abrir, cerrar, minimizar, maximizar, mover, escalar y mantener un listado de las ventanas abiertas). Los **gestores de ventanas** difieren entre sí de muchas maneras, incluyendo apariencia, consumo de memoria, opciones de personalización, escritorios múltiples o virtuales, entre otros.



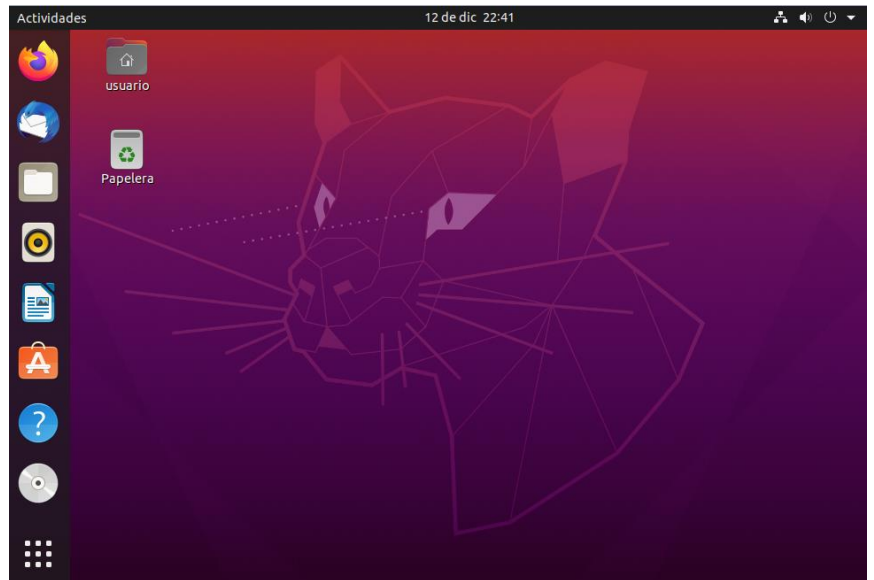
Un **entorno de escritorio** o, simplemente, **escritorio** hace uso de un gestor de ventanas y, en general, consta de **iconos, ventanas, barras de herramientas, carpetas, fondos de pantalla y widgets de escritorio**. Su objetivo es ofrecer al usuario una interacción amigable y cómoda: ofrece facilidades de acceso y configuración, como barras de herramientas, habilidades como arrastrar y soltar, permite poner imágenes (iconos) en la pantalla (escritorio), definir temas para cursores, menús emergentes en el escritorio... Todo eso hace que el entorno gráfico (escritorio) se vuelva pesado.



## 8.1. ELEMENTOS DEL ESCRITORIO DE UBUNTU

Esto es lo que vemos cuando arrancamos un ordenador con Ubuntu con GNOME:

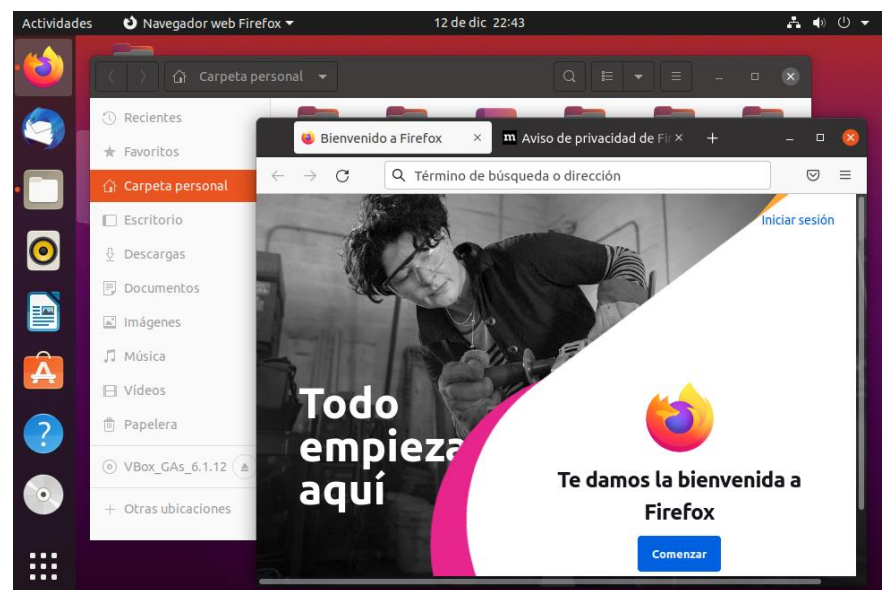
Ubuntu le ha dado su toque personal y ha decidido anclar el dock a la izquierda de la pantalla en lugar de mantenerse escondido.



Veamos cada una de ellas:

### a) El área de aplicaciones

Es la parte del escritorio donde se muestran las diferentes aplicaciones que tengas abiertas.



### b) El panel o área de indicadores/notificación

Se encuentra en la parte superior de la pantalla.



Por defecto incluye las siguientes opciones:

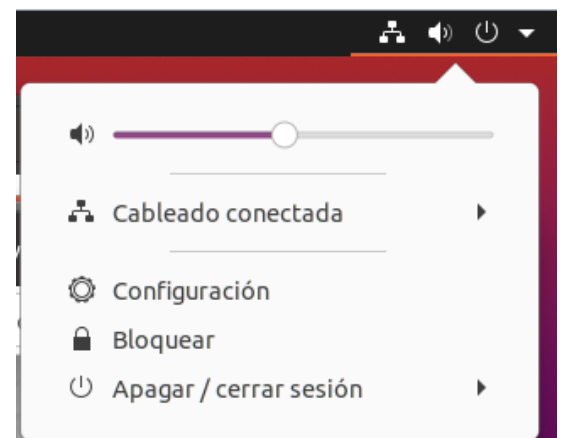
- **Indicador de configuración de sistema y sesiones**

- **Control de configuración del sistema.**

A través del icono de herramientas te permite acceder a algunos dispositivos instalados como la webcam o la impresora, al Software y actualizaciones que te indicará si tu sistema está actualizado o no, a la configuración de las aplicaciones que se inician con el sistema o a muchas y variadas opciones de configuración como la apariencia, el comportamiento del ratón, las opciones de red, el teclado, los monitores, etc.

- **Cambiar de usuario en el acceso al sistema o bloquear.**

- **Configurar o apagar la interfaz de red**



## - Cerrar sesión, apagar o suspender el sistema

*Nota: ¿Qué diferencia existe entre suspender e hibernar un ordenador? La **suspensión** detiene todos los procesos y guarda el estado del sistema en memoria RAM. El ordenador se pone en modo ahorro de energía, pero sigue consumiendo. Todo volverá a la normalidad dando un toque al botón de encendido o pulsando alguna tecla. Por otro lado, la **hibernación** envía el contenido de la memoria a la partición swap, informa al gestor de arranque para el próximo inicio y apaga el ordenador. Durante la hibernación no se consume energía. Para recuperar la sesión pulsaremos el botón de encendido.*

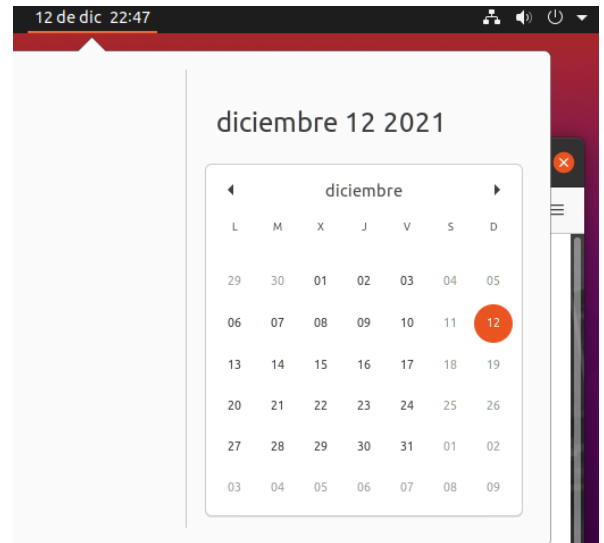
### • Indicador de fecha y hora

Visualiza la hora establecida en el sistema y el calendario mensual (sólo tienes que hacer clic en él para activarlo y un nuevo clic para desactivarlo). Tiene un botón de configuración que te permite configurarlo para que muestre también el día de la semana o cambiar el formato de la hora o del calendario.

### • Indicador de sonido

### • Indicador del administrador de red

### • Otros indicadores:



### Controles de red y bluetooth.

Ambos controles te permiten acceder a la gestión de la red o los dispositivos bluetooth.



### Control de la batería del portátil o de la toma de corriente.

Si dispones de un portátil puedes ver el indicador de la batería. El menú desplegable dispone de un botón que te permitirá configurar el uso de la energía.

## c) El launcher o lanzador de aplicaciones

Es una barra situada a la izquierda del escritorio que incluye accesos a diferentes lugares o aplicaciones básicas del sistema. Puedes utilizar el menú contextual del botón derecho del ratón para acceder a algunas opciones personalizadas en cada lanzador.

De arriba a abajo puedes ver:



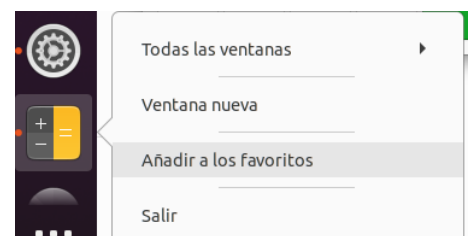
**Carpeta personal.** El lanzador te permite acceder a las carpetas y archivos de tu carpeta personal /home.

Incluye una barra de progreso en el lanzador cuando se están realizando copias de archivos y también listas dinámicas que permite acceder con mayor facilidad al diálogo de copiar o cancelar las acciones que se están llevando a cabo en cada momento o al contenido de los subdirectorios de la carpeta.

**Lanzadores a aplicaciones favoritas.** El Launcher muestra los lanzadores a algunas aplicaciones consideradas populares o favoritas como el acceso a Writer, a Impress, el correo, el navegador o al Centro de software de Ubuntu. Puedes personalizarlo añadiendo tus propios lanzadores al Launcher o también eliminar los que ya tengas establecidos.

Veamos algunas de sus útiles opciones:

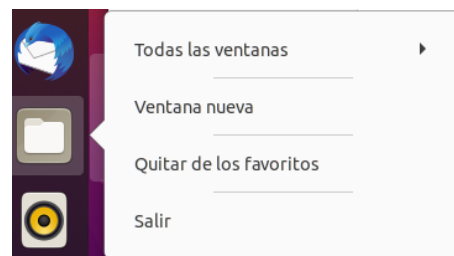
- **Reconocer la aplicación existente tras un lanzador.** Acerca el ratón al icono que se muestra en el Launcher y un mensaje informativo te indicará a qué programa hace referencia. Mostrará su nombre en pantalla.
- **Iniciar una aplicación alojada en el lanzador.** Tan sencillo como hacer un clic con el botón izquierdo del ratón sobre el icono presente en el Launcher.
- **Añadir y eliminar lanzadores en el Launcher.** Para añadir nuevos iconos al lanzador en primer lugar arranca o inicia la aplicación que desees añadir. A continuación, haz clic con el botón derecho del ratón sobre el icono temporal que se muestra en el lanzador y selecciona la entrada **Añadir a favoritos**.



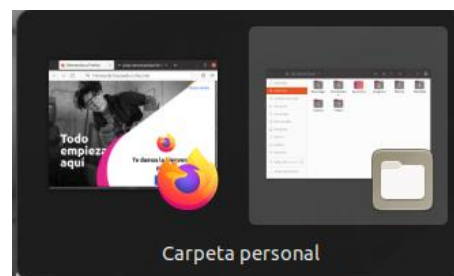
a) **Para quitar los lanzadores del Launcher utiliza un procedimiento similar.** En este caso haz clic con el botón derecho del ratón sobre el lanzador que quieras eliminar del Launcher y selecciona la entrada **Quitar de los favoritos**.

b) **Cambiar el orden de los lanzadores.** Sólo tienes que dejar pulsado un par de segundos el botón izquierdo del ratón sobre el lanzador a mover y seguidamente ya puedes situarlo en una posición diferente.

c) **Diferentes opciones en cada lanzador.** El botón derecho del ratón sobre un icono despliega un globo con el nombre del programa y opciones adicionales. Por ejemplo, el lanzador Writer ofrece la opción de abrir un documento nuevo, el nombre de la aplicación, la opción No mantener en el lanzador para quitar el acceso directo del Launcher y una entrada Salir para finalizar el programa.



▪ **Moverse entre aplicaciones en ejecución.** El escritorio también te permite moverte entre aplicaciones que tienes en ejecución de una manera muy simple, pulsando la combinación de teclas Alt + Tab (mantenlas pulsadas unos segundos hasta que veas en pantalla los iconos que muestra). En efecto, puedes ver en pantalla los iconos de las diferentes aplicaciones abiertas. Con distintas pulsaciones de la tecla Tab puedes activar una u otra según tus necesidades.



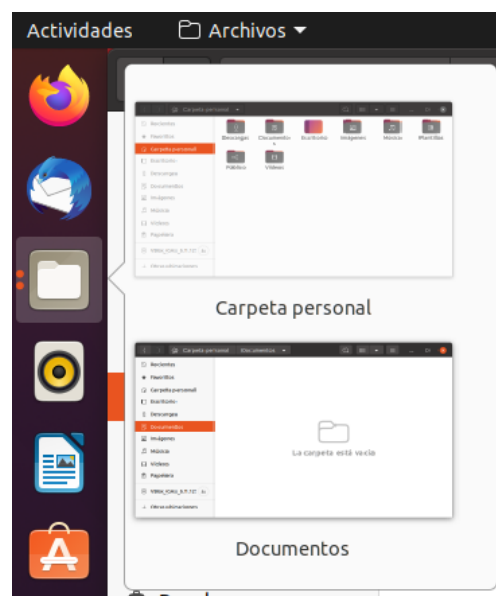
Si añades demasiados lanzadores al Launcher y el número de iconos es mayor de los que caben en tu monitor, si sitúas el puntero del ratón en la parte superior o inferior, dependiendo de donde estén los iconos que no se muestran, los ocultos, se producirá un desplazamiento vertical (arriba o abajo) de los iconos. Es decir, si mueves el ratón por encima de estos lanzadores, el Launcher tendrá un comportamiento scroll permitiéndote acceder así a todos los lanzadores que no veas directamente en pantalla.



El Launcher también te muestra qué aplicaciones tienes abiertas y cuál de ellas es la aplicación activa. La aplicación que ya está en ejecución muestra un punto naranja a la izquierda en su lanzador. Además, la aplicación activa se indica con un sombreado. Los demás programas que están en ejecución sólo muestran el punto a la izquierda. Si dispones de varias instancias abiertas, en ejecución, de una misma aplicación (por ejemplo, varias ventanas de tu carpeta home) el sistema te lo indicará con una marca por cada una de ellas a la izquierda del icono del lanzador.

Observa la captura de pantalla siguiente: dos puntos a la izquierda del icono de Archivos significa que tienes dos instancias abiertas de la ventana en el escritorio.

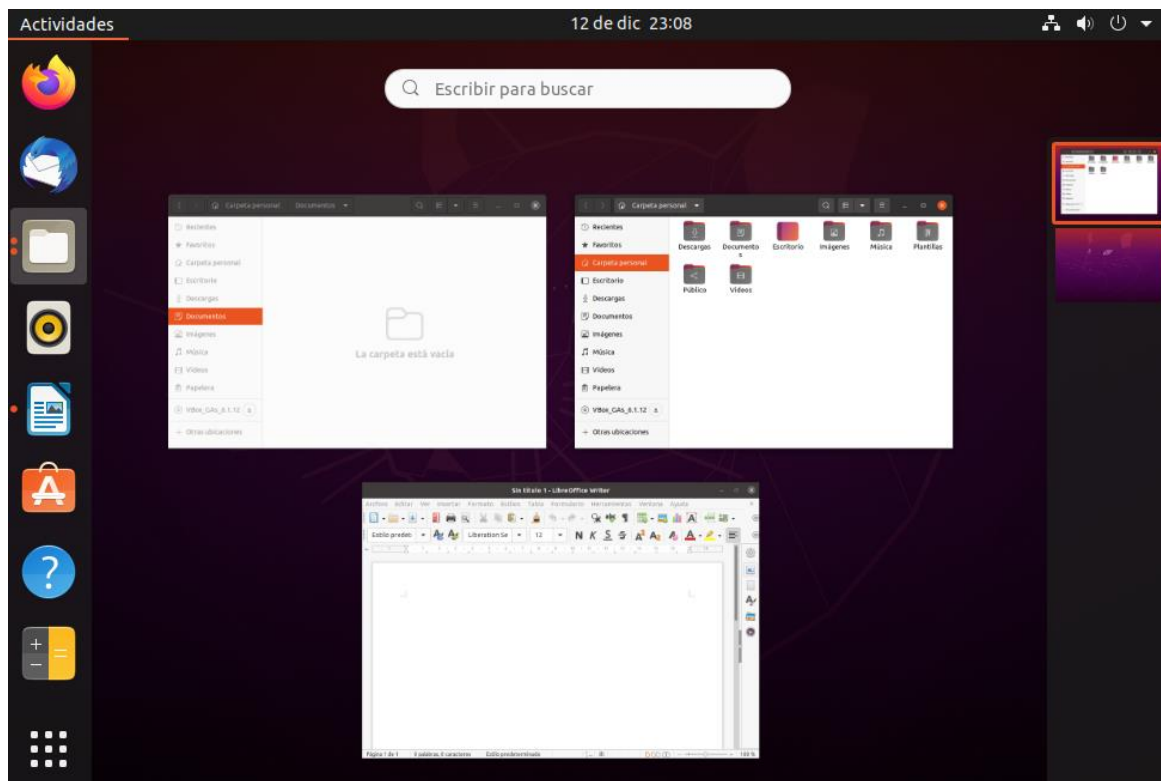
Si ahora haces un nuevo clic en la carpeta home en el Launcher podrás ver las dos ventanas abiertas de Archivos y acceder a cualquiera de ellas simplemente haciendo clic en su ventana para convertirla en ventana activa:



- **Áreas de trabajo (escritorios virtuales).**

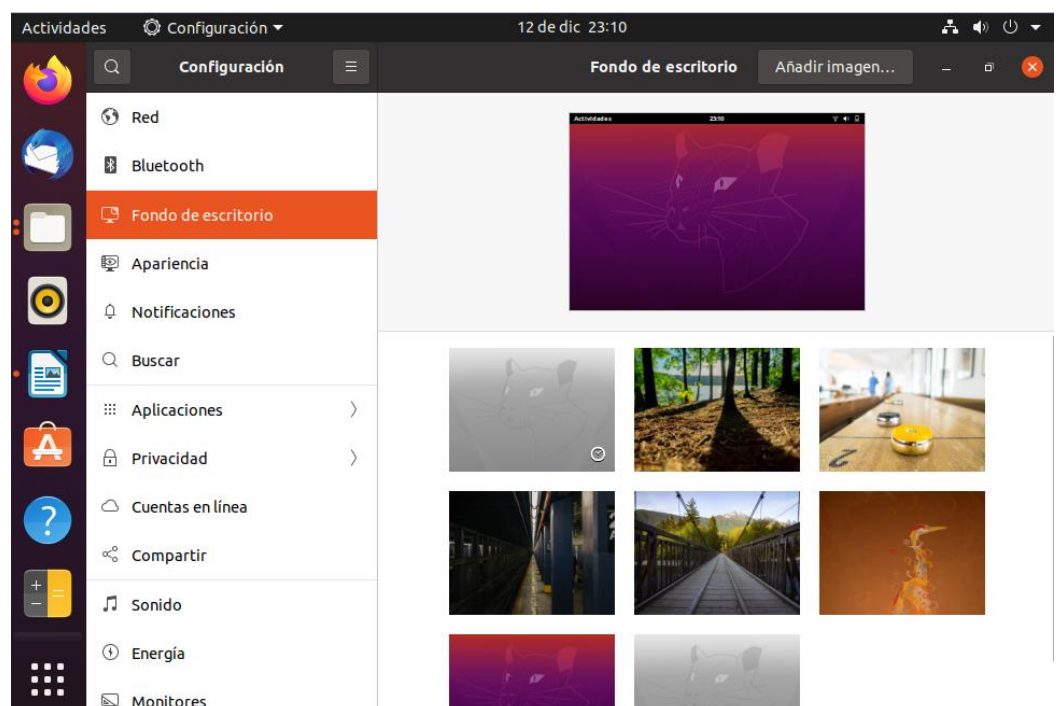
Es muy común trabajar con varias aplicaciones a la misma vez y las áreas de trabajo o escritorios virtuales nos permite agrupar las aplicaciones en ellas, mejorando la navegación por el escritorio.

Para trabajar con ellas pulsaremos sobre **Actividades** y observaremos que a la parte derecha aparecen los escritorios virtuales. Basta con hacer click sobre la barra superior de cualquier aplicación y arrastrarla al área de trabajo deseada.



## 8.2. PERSONALIZACIÓN DEL ESCRITORIO

Además de poder modificar los lanzadores del launcher, podemos personalizar el aspecto del escritorio. Para ello, simplemente hay que hacer clic con el botón derecho en el escritorio y elegir la opción “Cambiar el fondo”:





## 9. SISTEMA DE ARCHIVOS EN LINUX

---

### 9.1. TIPOS DE SISTEMAS DE ARCHIVOS

Un **sistema de ficheros** es el modo en que el sistema operativo organiza los ficheros en el disco duro, gestionándolo de manera que los datos estén de forma estructurada y sin errores.

Linux soporta gran variedad de sistemas de ficheros, desde sistemas basados en discos, como pueden ser **ext2**, **ext3**, **ext4**, **ReiserFS**, **XFS**, **JFS**, **UFS**, **ISO9660**, **FAT**, **FAT32** o **NTFS**, a sistemas de ficheros que sirven para comunicar equipos en la red de diferentes sistemas operativos, como **NFS** (utilizado para compartir recursos entre equipos Linux) o **SMB** (para compartir recursos entre máquinas Linux y Windows).

Los sistemas de ficheros indican el modo en que se gestionan los ficheros dentro de las particiones. Según su complejidad, tienen características como previsión de apagones, posibilidad de recuperar datos, indexación para búsquedas rápidas, reducción de la fragmentación para agilizar la lectura de los datos, etc. Hay varios tipos, normalmente ligados a sistemas operativos concretos. A continuación se enumeran los más representativos:

- **ext2:** Hasta hace poco era el sistema estándar de Linux. Tiene una fragmentación muy baja, aunque es algo lento manejando archivos de gran tamaño. Fue la continuación del sistema de ficheros **ext**, implementado en 1992 e integrado en Linux 0.96. Las principales ventajas que tenía sobre ext eran las siguientes:
  - Compatible con sistemas de ficheros grandes, admitiendo particiones de disco de hasta 4TB y ficheros de hasta 2GB de tamaño.
  - Proporciona nombres de ficheros largos, de hasta 255 caracteres.
  - Tiene una gran estabilidad.
- **ext3:** Es la versión mejorada de ext2, con previsión de pérdida de datos por fallos del disco o apagones (esta técnica que registra a diario los cambios en el sistema de archivos para poder recuperar los datos en caso de fallo se conoce como journaling). En contraprestación, es totalmente imposible recuperar datos borrados. Es compatible con el sistema de ficheros ext2. Sus ventajas frente a ext2 son:
  - Actualización. Debido a que los dos sistemas comparten el mismo formato, es posible llevar a cabo una actualización a ext3, incluso aunque el sistema ext2 esté montado.
  - Fiabilidad.
- **ext4:** Es la última versión de la familia de sistemas de ficheros ext. Sus principales ventajas son:
  - Mejora el rendimiento global y eficiencia: menor uso de CPU, mejoras en la velocidad de lectura y escritura, se reduce el tiempo de borrado de archivos largos, apenas se fragmenta (aunque añade soporte para la desfragmentación).
  - Soporta “undelete”, es decir, permite recuperar archivos borrados de forma accidental.
  - Amplía los límites de tamaño de los ficheros, ahora de hasta 16TB, y del sistema de ficheros, que puede llegar a los 1024PB (1EB).
  - Android adoptó en la versión 2.3 ext4.
- **ReiserFS:** Es el sistema de ficheros de algunas distribuciones de Linux (por ejemplo, Elive o Yoper). Soporta journaling.
- **swap:** Es el sistema de ficheros para la partición de intercambio de Linux. Todos los sistemas Linux necesitan una partición de este tipo para cargar los programas y no saturar la memoria RAM cuando se excede su capacidad. En Windows, esto se hace con el archivo pagefile.sys.

En general, ext2, ext3 y ext4 son compatibles entre sí. Siendo posible, por ejemplo, montar un sistema de archivos ext3 como ext4 y usarlo transparentemente o una partición ext4 (que no use extents) ser montada como ext3.

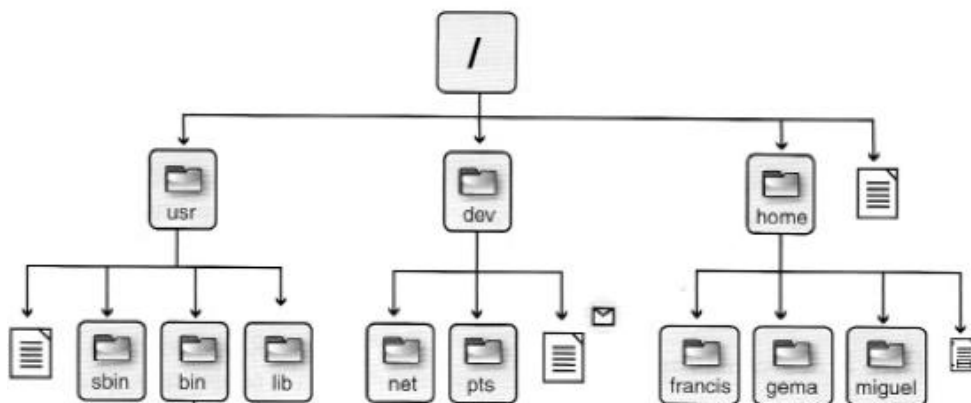
Además de estos sistemas de ficheros, Linux también ofrece soporte para sistemas de ficheros de Windows, como FAT, FAT32 y NTFS.



## 9.2. CARACTERÍSTICAS GENERALES DE LOS SISTEMAS DE FICHEROS DE LINUX

### Directorio raíz “/” :

El sistema de ficheros de Linux tiene una **estructura jerárquica**, donde el **directorio principal (directorio raíz)** es el directorio **/**, del que cuelga toda la estructura del sistema. Por tanto, en **Linux** existe **un único árbol de directorios** y no uno por partición o disco (C:, D:... ) como sucede en Windows. Es decir, en Linux todos los archivos y directorios “cuelgan” este directorio raíz **/**. Por tanto, cuando se conecta por ejemplo, un pendrive, este estará colgando del directorio raíz **/** o de uno de sus directorios, no tendrá una unidad independiente o letra como pasaba en Windows.



### Archivos de usuarios (/home):

- El directorio **home** es el directorio que el sistema crea para cada usuario que ha sido dado de alta. En él cada usuario puede guardar sus archivos y directorios propios.
- Los **directorios personales de los usuarios** se encuentran bajo el directorio **/home**. De este modo el directorio **home** del usuario **mar** sería **/home/mar**
- En cambio, el **directorio personal del usuario root** o superusuario, es decir, el directorio personal del administrador del sistema, no se encuentra en **/home**, sino que se encuentra en **/root**

### Nombre de los ficheros y carpetas:

Respecto al nombre de ficheros y carpetas, debe tener entre 1 y 255 caracteres y se puede usar **cualquier carácter**, **excepto el carácter /** (que se usa para el directorio raíz). De todas maneras no es recomendable usar los siguientes caracteres, por tener significado especial en Linux:

`\ ^ ~ ! # ? & ( ) ' " ` ; . $ = & i < > @ { } * + -`

### Distinción de mayúsculas y minúsculas:

Linux **distingue entre mayúsculas y minúsculas** en todo, es decir, en nombres de carpetas, ficheros, comandos, programas o cualquier elemento. Eso significa que podremos crear en el mismo directorio el fichero “ejercicios.doc” y el fichero “Ejercicios.doc”, porque serán ficheros diferentes.

### Rutas absolutas y relativas:

Existen dos maneras de indicar la ruta a seguir a través del árbol de directorios de Linux hasta alcanzar un fichero o directorio:

- **Ruta absoluta:** Si indicamos la ruta desde el directorio raíz del árbol, construimos una ruta absoluta. Comenzamos indicando el directorio raíz **/** y, a continuación, añadimos los sucesivos directorios hasta alcanzar el destino separándolos mediante la barra **/**.  
Por ejemplo: **/home/mar/doc**
- **Ruta relativa:** Si indicamos la ruta a seguir desde el directorio actual (que se representa por un punto **.**), construimos una ruta relativa.  
Por ejemplo: si estamos en el directorio **home** de **mar**, la ruta relativa a al directorio **doc** sería: **./doc**

## Directorio actual (.), directorio padre (..) y directorio personal (~):

Existen una serie de trucos que facilitan el acceso a ficheros y directorios:

- **..** (dos puntos seguidos) → representa el **directorio padre** del directorio actual
- **.** (un punto) → representa el **directorio actual**
- **~** → representa nuestro **directorio home** (se escribe pulsando Alt Gr + 4)

Puedes usar estos “trucos” a la hora de moverte por el árbol de directorios de Linux.

### 9.3. DIRECTORIOS IMPORTANTES DE LINUX

- **/bin y /usr/bin**: contienen la mayoría de ficheros ejecutables y comandos comunes de Linux que pueden usar todos los usuarios.
- **/boot**: contiene los ficheros necesarios para el arranque del sistema (ficheros del gestor de arranque LILO o GRUB)
- **/dev**: almacena “ficheros especiales” que representan los dispositivos de E/S. En realidad no son “ficheros” propiamente dichos, sino que es la forma en que Linux implementa los controladores de dispositivos.
- **/etc**: contiene los ficheros de configuración del sistema. Este directorio es MUY importante. Evidentemente, solo un usuario con privilegios de administrador puede modificarlos.
  - **/etc/init.d** → aquí se encuentran los Shell scripts que facilitan el inicio y cierre de los “daemons” o demonios (procesos que se ejecutan en segundo plano para dar algún tipo de servicio)
- **/home**: donde se almacenan el “directorio home” o, dicho de otro modo, la “carpeta personal”, de cada uno de los usuarios. Es como el “documents and settings” de Windows.
- **/lib y /usr/lib**: contienen librerías compartidas del sistema (librerías necesarias para que se ejecuten los programas que tenemos en /bin y /sbin)
- **/lost+found**: encontramos información y los procesos que se ejecutaban antes de una caída del sistema
- **/media**: cuando se monta automáticamente un CD\_ROM, o pendrive o disco duro externo, se crea aquí su subdirectorio. Por ejemplo: /media/usbdisk
- **/mnt**: es el directorio por defecto para realizar el montaje de otros dispositivos de almacenamiento
- **/opt**: directorio opcional donde se puede instalar aplicaciones (las que no se instalan automáticamente), además de en /usr. En algunas distribuciones Linux no existe este directorio.
- **/proc**: contiene archivos que reciben o envían información al kernel (núcleo). Por ejemplo: el fichero partitions contiene información de las particiones del disco y filesystem contiene los sistemas de archivos del sistema.
- **/root**: es el directorio home (personal) del usuario root
- **/sbin y /usr/sbin**: contienen comandos y ejecutables de tareas de administración que en su mayoría solo puede usar, evidentemente, el usuario root
- **/tmp**: almacena ficheros temporales
- **/usr**: almacena las aplicaciones de uso general para todos los usuarios, por lo que si hay muchos paquetes instalados, puede ocupar mucho espacio.
  - **/usr/bin**: contiene programas y comandos para usar los usuarios
  - **/usr/share**: datos compartidos independientes de la máquina, como documentación de programas, imágenes para el Escritorio de Linux...
- **/var**: contiene información variable, como colas de impresión, de envío, de recepción de correos, archivos de registro y de eventos del sistema...
  - **/var/lock**: contiene los ficheros de bloque para indicar que un dispositivo está inaccesible cuando está siendo usado por una aplicación.
  - **/var/log**: almacena los registros detallados de la actividad realizada durante una sesión de trabajo en el sistema
  - **/var/spool**: contiene información preparada para ser transferida, como los trabajos de impresión o los mensajes de correo y los eventos que se hayan programado con los comandos at y cron.

## 9.4. TIPOS DE ARCHIVOS

En Linux existen estos tipos de archivos:

- **Archivos regulares.** Contienen la información con la que trabaja cada usuario. Son los archivos ordinarios de datos.
- **Directorios.** Son archivos especiales que contienen referencias a otros archivos o directorios.
- **Enlaces físicos o duros (hard links).** No es específicamente una clase de archivo sino un segundo nombre que se le da a un archivo.  
Cada archivo en el sistema de archivos de Linux tiene su propio *i-nodo*. Un i-nodo guarda los atributos del archivo y su posición en el disco. Si necesitamos referirnos a un archivo usando dos nombres de archivo distintos, podemos crear un enlace duro. El enlace duro tendrá el mismo i-nodo que el archivo original y guarda el mismo aspecto y comportamiento que el original. Con cada enlace duro creado, se incrementa la *cuenta de enlaces o referencia*. El conteo del número de enlaces duros que tiene un fichero se realiza, como ya hemos visto, en el inodo correspondiente a los datos del fichero. Cuando se borra un enlace duro, se decrementa esta cuenta de enlaces. Hasta que la cuenta de enlaces no alcanza cero, el archivo permanece en el disco.
- **Enlaces simbólicos o blandos (soft links).** También se utilizan para asignar un segundo nombre a un archivo. La diferencia con los enlaces duros es que los simbólicos solamente hacen referencia al nombre del archivo original, mientras que los duros hacen referencia al inodo en el que están situados los datos del archivo original. Sería como los “accesos directos” de Windows.  
De esta manera, si tenemos un enlace simbólico y borramos el archivo original perderemos los datos, mientras que si tenemos un enlace duro los datos no se borrarán hasta que se hayan borrado todos y cada uno de los enlaces duros que existen hacia esos datos en el sistema de ficheros. Además, se mostraría un error al intentar acceder al archivo borrado a través del enlace simbólico.
- **Archivos especiales.** Suelen representar dispositivos físicos, como unidades de almacenamiento, impresoras, terminales, etc. En Linux, todo dispositivo físico que se conecte al ordenador está asociado a un archivo. Pueden ser de dos tipos: de bloques o de caracteres. Un disco duro es un dispositivo de bloque y una impresora es un dispositivo de carácter.
- Otros tipos de archivo son **las tuberías (pipes) o los sockets**.

Tipos de archivos (ls -l)	
-	Archivo ordinario
d	Directorio
l	Enlace simbólico
b / c	Archivo especial (c= de caracteres y b=de bloques)
p	Tubería
s	Socket

## 9.5. FICHEROS Y DIRECTORIOS OCULTOS

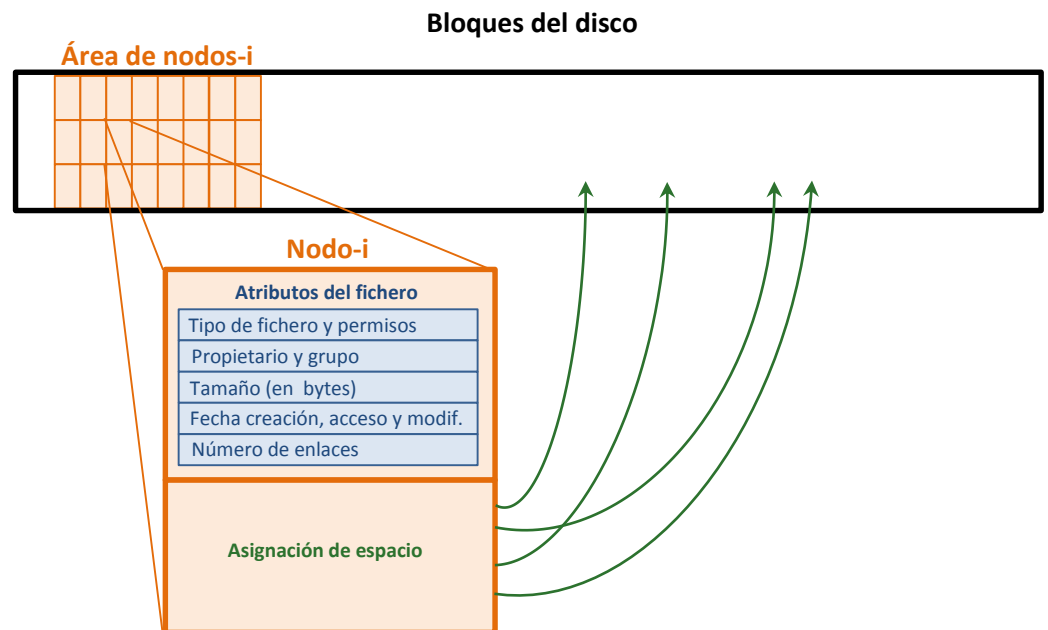
Los ficheros “ocultos” en Linux son **aquellos cuyo nombre comienza con el carácter punto (.)**

Al igual que en los sistemas de ficheros de Microsoft, en todos los directorios existen dos ficheros ocultos especiales: **.** (directorio actual) y **..** (directorio padre).

## 9.6. LOS INODOS O NODOS-I:

En Linux, a cada archivo o directorio se le asigna un número identificador único llamado inodo. Es como el “DNI” de cada fichero/directorio.

De este modo, existe una tabla de inodos en la que hay una entrada por cada inodo en la que se guarda toda la información importante del fichero: el propietario, el grupo, permisos, tipo de fichero, fecha de creación, fecha de última modificación... Además, también guarda las direcciones de los bloques del disco donde se encuentra el fichero.



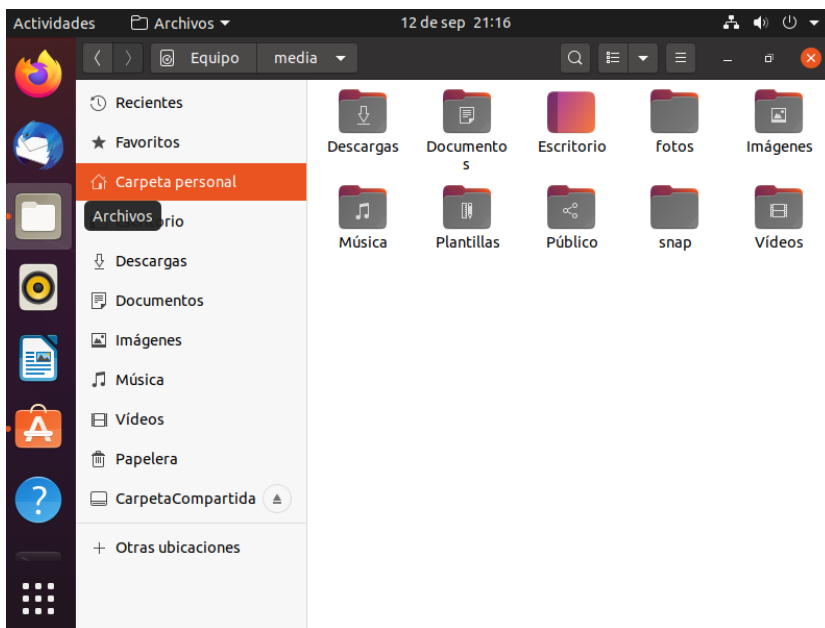
La única propiedad que no se almacena aquí es el nombre del fichero que se almacena en el directorio al que pertenece al archivo y es aquí donde se asocia el nombre del archivo a su inodo correspondiente.

## 9.7. OPERACIONES CON FICHEROS Y DIRECTORIOS EN MODO GRÁFICO

Sobre un fichero o directorio podemos realizar distintas operaciones, como, por ejemplo: crearlos, entrar o salir de estos, leer su contenido, borrarlos...

Para acceder a nuestros archivos y poder trabajar con ellos, haremos clic en el lanzador “Archivos”.

A continuación, se describe cómo realizar algunas de estas habituales operaciones:



Cómo **seleccionar más de 1 archivo/carpeta**:

- Si están juntos:
  - Pinchar con el ratón en el primero y luego pinchar en el último manteniendo pulsada la tecla SHIFT. Así seleccionamos todos los que estén en medio también.
  - Pinchando con el ratón y dibujando un cuadrado que los envuelva.
- Si están separados:
  - Pinchar con el ratón cada archivo, pulsando además la tecla Ctrl del teclado a la vez (con la otra mano)

Cómo **copiar, cortar, pegar** archivos/carpetas:

- Teclado:
  - COPIAR: Ctrl + C
  - CORTAR: Ctrl + X
  - PEGAR: Ctrl + V
- Botón derecho del ratón → cortar | copiar | pegar
- Menú Editar → cortar | copiar | pegar

Cómo **seleccionar todo**:

- Ctrl + A
- Menú Editar → Seleccionar todo

Cómo **crear carpetas**:

- Botón derecho del Ratón → crear una carpeta
- Menú Archivo → Crear una carpeta

Cómo **cambiar el nombre a un archivo/carpeta**:

- Botón derecho del Ratón > Renombrar
- Menú Editar → renombrar
- Tecla F2

Cómo **compartir carpetas a través de la red**:

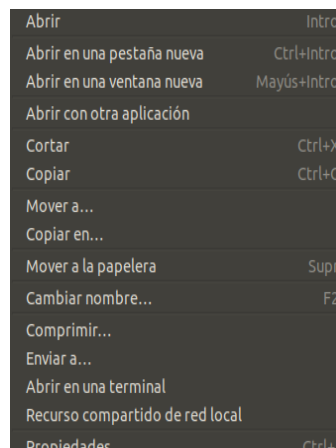
- Botón derecho → opciones de compartición
- Menú Editar → opciones de compartición

Cómo **eliminar un archivo/carpeta**:

- Botón derecho del ratón → mover a la papelera
- Menú Editar → mover a la papelera

Habrás observado que el manejo de archivos y carpetas es diferente que en Windows: fácilmente, desde el botón derecho del ratón puedes:

- Abrir, cortar y copiar
- Abrir con otra aplicación
- Crear un enlace
- Cambiar nombre
- Mover a la papelera
- Comprimir: se utiliza para hacer una copia comprimida de todo el contenido de la carpeta seleccionada
- Recurso compartido de red local: se usa para compartir la carpeta en la red
- Enviar a: se usa para enviar directamente por correo electrónico, o a un DVD
- Propiedades: muestra las propiedades del archivo o carpeta seleccionada.
  - Pestaña básica: vemos las características de un archivo, como el tipo, contenido, fecha de modificación, además nos permite modificar su nombre
  - Permisos: podemos modificar los privilegios o permisos que los usuarios tienen sobre esta carpeta o archivo
  - Compartir: se utiliza para hacer que la carpeta esté a disposición de otros usuarios de la red



## 10. INTERFAZ TEXTUAL: SHELL

---

En Linux la interfaz de línea de comandos se conoce como **Shell**. La Shell Bash es la predeterminada en la mayoría de los sistemas Linux, como Ubuntu. No obstante, aparte de la Shell Bash, existen otras como sh, csh, ksh...

En Ubuntu se accede a la Shell a través del denominado **Terminal**. El terminal consiste en una pantalla en la que podemos introducir comandos.

Podemos acceder a un Shell de varias formas:

- Menú Aplicaciones del Dock → Terminal
- Pulsando simultáneamente las teclas Ctrl+Alt+T

En general, todas las órdenes de UNIX/Linux son programas que están almacenados en el sistema de ficheros. Su sintaxis es similar a la del entorno comando de Windows y es la siguiente:

**orden [-opciones] [argumentos]**

Antes de empezar a trabajar con los comandos de la Shell, ten en cuenta los siguientes **consejos**:

- Para trabajar más rápida y cómodamente con el Shell, puedes utilizar los siguientes **atajos**:
  - o Puedes **completar** los comandos usando la tecla **tabulador**
  - o Puedes **reescribir** comandos ya utilizados usando las **teclas** ↑ y ↓
- Puedes **detener** la ejecución de un comando pulsando **Ctrl+c**
- Mediante el comando **sudo** (que deberás escribir antes del comando) puedes **ejecutar comandos** que requieran **privilegios de root**
  - o El fichero **/etc/sudoers** contiene qué **usuarios o grupos** pueden **usar el comando sudo**
- Mediante la orden **su**, puedes **pasar a ser otro usuario**. Por ejemplo:
  - o **sudo su** → te conviertes en el usuario root
  - o **sudo su mar** → te conviertes en el usuario mar

El **prompt** del Shell bash muestra dos **formas** distintas según el tipo de usuario que lo utilice:

- Si se trata de un usuario **estándar** → finaliza con el carácter **\$**
- Si se trata de la cuenta de superusuario o **root** del sistema → finaliza con el carácter **#**

### 10.1. COMANDOS

A continuación, se indica un resumen de los principales comandos.

Puedes ver un vídeo con ejemplos de algunos comandos básicos en: <https://youtu.be/8e-UBTlu6X8>)

Más vídeos sobre comandos en:

- [https://www.youtube.com/watch?v=IT\\_vW-xd7f8](https://www.youtube.com/watch?v=IT_vW-xd7f8)
- <https://www.youtube.com/watch?v=-cvkGkWLJig>
- <https://www.youtube.com/watch?v=ydTrNpaHSX4>



COMANDO	USO	QUÉ HACE	EJEMPLO
<b>Comandos para manejo de la interfaz</b>			
Man help	man comando comando --help	Información de ayuda de los comandos	man ls muestra el manual de uso del comando ls (para salir de man pulsa q) ls -help
echo	echo texto	Muestra un texto literal por pantalla	echo hola Muestra el texto hola
clear	clear	Borra la pantalla	clear
exit	exit	Salir del shell	exit

<b>Comandos generales de configuración del sistema</b>			
date	date	ver/poner la fecha del sistema	date muestra la fecha y hora actual date -s establece la fecha que le indiquemos
cal	cal	Muestra un calendario	cal
shutdown	shutdown [-opciones] tiempo [mensaje]	Apaga el sistema	shutdown 1 apaga el sistema en 1 minuto shutdown -h now apaga el sistema y el equipo shutdown -t sec: Espera t segundos antes de iniciar el proceso de cierre del sistema shutdown -r: reinicia (reboot) después de shutdown.
reboot	reboot	Reinicia el sistema	reboot reinicia el sistema

<b>Comandos de tratamiento de unidades de disco</b>			
du	du [-opciones] [fichero]	Muestra un resumen del uso de disco para cada fichero o directorio. Recursivo para directorios.	du -h muestra los tamaños de forma legible (p.ej: 1K 234M 2G)
df	df [-opciones] [fichero]	Muestra información sobre el sistema de ficheros en el que reside cada fichero. Por defecto, de todos los sistemas de ficheros.	df -h muestra los tamaños de forma legible (p.ej: 1K 234M 2G)

<b>Comandos de manejo de ficheros y directorios</b>			
touch	touch fichero(s)	Crea un fichero vacío	touch ejercicio1.txt crea el fichero vacío ejercicio1.txt
vi nano	vi fichero nano fichero	Crear o modificar un fichero	vi ej1.txt nano ej1.txt
cat more less	cat fichero(s) more fichero less fichero	Ver el contenido de un archivo	cat ej1.txt muestra por pantalla el contenido del fichero ej1.txt more ej1.txt muestra el contenido del fichero de forma paginada (pulsando intro avanza una línea y pulsando la barra espaciadora avanza una página)
grep	grep [-opciones] patrón [fichero]	Muestra las líneas de un fichero que contienen un determinado patrón (cadena de texto)	grep root /etc/passwd muestra las líneas del fichero /etc/passwd que contienen la palabra root
head	head [-opciones] fichero	Muestra las 10 primeras líneas del fichero	head -n 5 index.php muestra las 5 primeras líneas del fichero
tail	tail [-opciones] fichero	Muestra las 10 últimas líneas del fichero	tail -n 15 messages.txt muestra las ultimas 15 líneas del fichero
wc	wc [-opciones] fichero	Muestra nº de líneas, palabras o caracteres	wc -l muestra el nº de líneas del fichero wc -w muestra el nº de palabras y wc -c muestra el nº de caracteres
cut	cut [-opciones] fichero cut -d -f fichero	Muestra solo ciertas columnas del fichero	Supongamos text.txt contiene el texto “Este es, un ejemplo, del comando cut” cut -d, -f1,2 text.txt muestra por pantalla “Este es, un ejemplo” Con -d indicamos el delimitador de columnas (la ,) y con -f indicamos las columnas a mostrar
sort	sort [-opciones] fichero	Ordena las líneas de un fichero	sort fichero ordena el contenido del fichero
pwd	pwd	Muestra la ruta completa del directorio actual	pwd
ls	ls [directorio] ls -l ls -a ls -r ls -i	Ver el contenido de un directorio	ls SOM muestra todos los ficheros y directorios del directorio SOM ls -l lista los archivos con su información detallada ls -a lista todos los archivos, incluidos los archivos ocultos ls -r SOM lista el contenido del directorio SOM y de todos sus subdirectorios (de forma recursiva) ls -i muestra el nodo-i
cd	cd [directorio]	Cambiar de directorio	cd /home/mar/SOM cambia al directorio SOM usando una ruta absoluta cd ./SOM cambia al directorio SOM usando una ruta relativa cd .. cambia al directorio padre cd ~ cambia al directorio home del usuario

mkdir	mkdir directorio mkdir -p	Crear un directorio	mkdir SOM crea un directorio llamado SOM mkdir -p SOM/PRACTICAS crea todos los directorios que conforman la ruta que le damos, es decir, crea el directorio SOM y dentro de éste, el directorio PRACTICAS
rm	rm fichero(s)	Eliminar un archivo	rm ./SOM/s* borra todos los ficheros del directorio SOM que comienzan por el carácter s (borra solo los ficheros, no subdirectorios)
	rm -r directorio(s)	Eliminar un directorio	rm -r ./SOM/SUBDIR1 borra el directorio SUBDIR1 completo (con sus ficheros y directorios)
cp	cp fichero(s) directorio	Copiar un archivo	cp ejercicio1.doc SOM copia el archivo ejercicio1.doc al directorio SOM cp ej* ./SOM copia todos los ficheros del directorio actual que empiezan por "ej" al directorio SOM cp ej? ./SOM copia todos los ficheros de tres caracteres del directorio actual que empiezan por "ej" al directorio SOM
	cp -r fichero(s) directorio(s) directorio	Copiar un archivo o directorio	cp -r ./SOM/practica1 ./SOM/PRACTICAS copia la carpeta practica1 dentro de la carpeta PRACTICAS
mv	mv fichero(s) directorio mv fichero1 fichero2	Cambiar el nombre o mover un archivo o directorio	mv ej1.txt pr1.txt ./tema1 mueve los ficheros ej1.txt y pr1.txt al directorio tema1 mv ej1.txt ej2.txt cambia el nombre del fichero ej1.txt a ej2.txt
ln	ln [-opciones] fichero enlace	Crea un enlace a un fichero	ln crea un enlace duro
	ln -s		ln -s crea un enlace simbólico
file	file fichero	Determina el tipo de un fichero	file ej1.txt muestra el tipo del fichero ej1.txt
chmod	chmod u g o a = + - r w x fichero(s) directorio(s)	Establecer permisos en un archivo o directorio	chmod u=rwx, g=rx, o=- ej1.txt
	chmod XXX fichero(s) directorio(s)		chmod 750 ej1.txt chmod g-wx ej1.txt
chown	chown [-opciones] propietario:[grupo] fichero	Cambia el dueño y/o grupo del fichero	chown mar:profesores examen.doc el nuevo propietario del fichero examen.doc es mar y su nuevo grupo es profesores
tar	tar [-opciones] [fichero] tar -cvf fichero.tar fichero(s) directorio(s)	Empaquetar ficheros y directorios (no comprime, solo empaqueta)	-c crea un nuevo fichero
	tar -tf fichero.tar tar -rvf fichero.tar fichero(s) directorio(s) tar -xvf fichero.tar tar -tfz fichero.tar		-x extrae archivos de un fichero -t lista el contenido del fichero -v verbose -z comprime o descomprime -f guarda en el fichero indicado
gzip	gzip [-opciones] fichero gzip fichero(s) directorio(s) gzip -d fichero.gz gzip -r directorio	Comprimir y descomprimir un archivo o directorio (reduce el tamaño de los ficheros)	gzip prac1.doc comprime el archivo prac1.doc que pasa a llamarse prac1.doc.gz gzip -d prac1.doc.gz descomprime el archivo prac1.doc.gz gzip -r Comprime los ficheros y subcarpetas de un directorio

### Direccionalientos → >, >>, <, 2>, 2>>, &>

DIRECCIONAMIENTO	QUÉ HACE	EJEMPLO
comando > fichero	crea un fichero nuevo cuyo contenido es el resultado del comando de la izquierda (si ya existía, lo destruye)	echo hola > saludo.txt crea un fichero saludo.txt cuyo contenido es la palabra hola
comando >> fichero	añade al final del fichero (sin borrar nada de lo que había antes) el resultado del comando de la izquierda	echo que tal >> saludo.txt
comando < fichero	Introduce como datos de entrada el fichero	sort < fichero_a_ordenar.doc
comando 2> fichero	Envía la salida de error de comando a fichero	ls -l 2> fichero Si hay un error, lo guarda en fichero (puede dar error si no tenemos permiso de lectura en el dir).
comando &> fichero	Envía la salida estándar y de error a fichero	cat datos1 datos2 &> datos

### Tuberías (pipes) → |

Uso: comando1 | comando2

Permite conectar la salida de un comando con la entrada del siguiente. Ejemplo: cat /etc/passwd | grep mar

### Variables de entorno del shell

Se utilizan para guardar información del sistema y del usuario. Entre las más usadas encontramos:

- \$HOME: Directorio personal.
- \$HOSTNAME: Nombre de la máquina.
- \$USER: Nombre del usuario.
- \$PWD: Directorio de trabajo actual.

De este modo, puedes escribir echo \$HOSTNAME para saber el nombre de tu equipo.

## 10.2. SHELL-SCRIPTS DE LINUX

Equivalen los archivos \*.BAT de MSDOS.

Son ficheros de texto que **contienen comandos** que se irán ejecutando por la Shell.

Para crear un fichero Shell script puedes usar: gedit, nano, vi, joe, vim, pico o emacs, entre otros.

**Para** que un script pueda **ejecutarse**, hay que **añadirle** al fichero de texto **permisos de ejecución**.

Cuando hacemos Shell scripts hay que decirle al sistema con qué Shell queremos ejecutar el script. Por ejemplo, para el Shell Bash escribiríamos en la **primera línea** del script:

**#!/bin/bash**

También es frecuente añadirle la **extensión .sh** para indicar al usuario que se trata de un Shell script.

### A. Operaciones básicas

- **Tabulador (Tab completion)**  
Sirve para completar el comando, sólo hay que escribir una letra y pulsar la tecla tabulador.
- **Tubería (|)**  
El primer comando se convierte en la salida del segundo comando.
- **Comentario (#)**  
Texto que no se ejecuta.
- **Unir varios comandos seguidos (;)**  
Para ejecutar varios comandos seguidos.
- **Ayuda sobre un comando**  
man Comando
- **Listado de todos los comandos**  
Pulsar dos veces la tecla tabulador.
- **Propiedades de un comando**  
man Comando

### B. Variables

- **Inicializar una variable**  
Comando: VARIABLE=Valor  
*Ejemplo:*  
MENSAJE="Hola Mundo"  
echo \$MENSAJE

- **Uso de variables especiales:**

Variable	Función
\$0	Nombre del Shell script
\$1, \$2, \$3...	Parámetros o argumentos que se introducen desde la línea de comandos
\$#	Número de parámetros o argumentos
\$*	Valor de todos los argumentos
\$?	Valor devuelto por el último comando ejecutado
\$\$	PID del Shell script

### C. Arrays

- **Inicializar un array**  
Comando: Valor=(Valor1 Valor2 ValorN)  
*Ejemplo:*  
\$Valor=(1 2 3); echo \${Valor[1]}

### D. Consola

- **Escribir en consola**  
Comando: echo "hola"  
*Ejemplo:*  
Hola
- **Leer en consola**  
Comando:  
read palabra  
echo \$palabra  
*Ejemplo:*  
echo "Escribir palabra:"; read palabra;  
echo "Palabra escrita:" \$palabra  
Resultado:  
Escribir palabra:  
Hola  
Palabra escrita: Hola

### E. Funciones

- **Declaración de la función**  
Function Nombre\_de\_funcion()  
{  
CÓDIGO\_A\_EJECUTAR  
}

- **Llamada a la función**  
Nombre\_de\_la\_función Valores

*Ejemplo:*  
Declarar función suma:  
function suma(){let a=2; let b=1; let c=b+a;  
echo \$c}  
Llamar a la function suma:  
suma  
Resultado  
3

### Estructuras de control: condicionales y bucles

Las estructuras de control las usaremos para que, dependiendo de un determinado valor de una variable o resultado de una condición, se ejecuten unos comandos u otros, o bien para que se ejecuten unos comandos un número determinado de veces, según la expresión que usemos.

#### if-then-fi

```
if [ condición ]  
then  
comando/s  
fi
```

#### if-then-else-fi

```
if [ condición ]  
then  
comando/s  
else  
comando/s  
fi
```

#### if-then-elif-then-else-fi

```
if [ condición ]  
then  
comando/s  
elif [ condición ]  
then  
comando/s  
else  
comando/s  
fi
```

#### case-in-esac

```
case $variable in  
expr1)comando/s;;  
expr2)comando/s;;  
...  
*) comando/s;;  
esac
```

#### for-in-do-done

```
for $variable in valores  
do  
comando/s  
done
```

#### while-do-done

```
while [condición]  
do  
comando/s  
done
```

### Expresiones condicionales

Para evaluar el valor devuelto por la condición, usaremos las siguientes expresiones dentro de corchetes [ ].

Condición	Devuelve verdadero (true) si...
-f \$variable	... variable es un fichero
-d \$variable	... variable es un directorio
-r/w/x \$variable	... variable tiene permiso de r/w/x
-e \$variable	... variable es un archivo que existe
\$var1 -nt \$var2	... var1 es un archivo más nuevo que var2
\$var1 -ot \$var2	... var1 es un archivo más antiguo que var2
-z \$variable	... variable es una cadena vacía
-n \$variable	... variable es una cadena no vacía
\$var1 = \$var2	... var1 es una cadena igual que var2
\$var1 != \$var2	... var1 es una cadena distinta de var2

## Ejemplos de Shell scripts:

- d) Típico script “hola mundo”:

```
#!/bin/bash
echo Hola Mundo
```

En este script, se muestra el mensaje Hola Mundo en la terminal

- e) Script de copia de seguridad muy simple:

```
#!/bin/bash
tar -czf /var/my-backup.tgz /home/yo/
```

En este script, creamos una copia comprimida del directorio home de un usuario.

- c) Script que borra la pantalla y muestra la fecha actual:

```
#!/bin/bash
#borra la pantalla
clear
#Muestra la fecha
Date
```

- d) Script en el que se hace uso de “if”:

```
#!/bin/bash
echo -e “Escribe el nombre de un fichero: \c”
read fichero
if [ -f $fichero ]
then
    ls -l $fichero
else
    echo “Error: $fichero no es un fichero”
fi
```

El script pregunta el nombre de un fichero. Si el fichero existe, muestra la información sobre él en formato largo. Si no existe o no es un fichero muestra un mensaje de error.

- e) Script en el que se hace uso de “for”:

```
#!/bin/bash
echo -e “Escribe el nombre de un fichero: \c”
read fichero
for var in *
do
    if [ $var -nt $fichero ]
    then
        echo “$var es más reciente”
    fi
done
```

Este script te pregunta por el nombre de un fichero. Si existe, para cada fichero del directorio personal, te dirá si es más reciente o no que el fichero que has escrito.

- f) Script en el que se hace uso de “for”:

```
#!/bin/bash
echo "Introduce el directorio a analizar:"
read var
lineas='ls $var | wc -l'
echo "El directorio tiene $lineas elementos"
for elemento in `ls $var`
do
    if [ -d $elemento ]
    then
        echo "$elemento es un directorio"
    elif [ -f $elemento ]
    then
        echo "$elemento es un fichero"
    else
        echo "$elemento no es ni un fichero ni un directorio"
    fi
done
```

Este script pide al usuario una ruta de directorio e indica para cada uno de los elementos que tiene este directorio, si se trata de un fichero, un directorio u otra cosa.

- g) Script en el que se hace uso de “while”:

```
#!/bin/bash
if [ $# -eq 0 ]
then
    echo "Error, no hay parámetros"
else
    while [ $# -ne 0 ]
    do
        if [ -f $1 ]
        then
            echo "$1 es un fichero"
        elif [ -d $1 ]
        then
            echo "$1 es un directorio"
        else
            echo "$1 no es un fichero ni directorio"
        fi
        shift
    done
fi
```

Este script recibe una serie de parámetros por la línea de comandos, los muestra y te dice si son ficheros o directorios.

## Más ejemplos de Shell scripts:

- h) Realiza un script llamado **archi**, al cual haya que suministrarle un parámetro que será la dirección absoluta de un archivo.
- SI el archivo existe
  - ENTONCES lo presentará en pantalla (cat )
  - SINO lo creará y añadirá la frase "Esto es un archivo nuevo"

Lo ejecutaremos así: `./archi /home/usuario/mifichero`

```
#!/bin/bash
if [ -f $1 ]
then
    cat $1
else
    echo "Esto es un archivo nuevo" > $1
fi
```

- i) Realiza un script llamado **dire**, al cual haya que suministrarle un parámetro que será la dirección absoluta de un directorio.
- SI el directorio existe
  - ENTONCES lo listará (ls -ali \$1)
  - SINO indicará que "No existe" y lo creará (mkdir \$1)

Lo ejecutaremos así: `./dire /home/usuario/micarpeta`

```
#!/bin/bash
if [ -d $1 ]
then
    ls -ali $1
else
    echo "No existe"
    mkdir $1
    echo "El directorio ha sido creado"
fi
```

- j) Realiza un script llamado **carnet**, al cual haya que suministrarle dos parámetros. Uno sera un nombre de persona y el otro la edad.
- SI es mayor o igual (-ge) de 18
  - ENTONCES escribirá el nombre y que es mayor de edad
  - SINO escribirá el nombre y que no es mayor de edad

Lo ejecutaremos así: `./carnet juan 23`

En este caso: \$1 es juan y \$2 es 23

```
#!/bin/bash
if [ $2 -ge 18 ]
then
    echo $1 eres mayor de edad, tienes $2
    echo Puedes sacarte el carnet
else
    echo $1 eres menor de edad, tienes $2
    echo No puedes tener carnet
fi
```



- k) Realiza un script llamado **fiche**, que solicite la dirección absoluta de un fichero
- SI el fichero existe
  - ENTONCES lo presentará en pantalla (cat )
  - SINO presentará la frase "El fichero no existe"

```
#!/bin/bash
echo "Introduce la direccion absoluta de un archivo"
read ruta
if [ -f $ruta ]
then
    cat $ruta
else
    echo "El archivo no existe"
fi
```

- l) Realiza un script llamado **dia**, el cuál solicite que introduzca un día de la semana: lunes, martes, miercoles, jueves, viernes, sabado, domingo. Según el día introducido saldrá una frase.

```
#!/bin/bash
echo "Introduzca un dia de la semana"
read dia
case $dia in
    lunes) echo "El lunes es una lata" ;;
    martes) echo "El martes es el segundo dia de la semana" ;;
    miercoles) echo "Hoy es miercoles" ;;
    jueves) echo "Siempre en medio" ;;
    viernes) echo "Ya es finde" ;;
    sabado) echo "A salir" ;;
    domingo) echo "Domingo pingo" ;;
esac
```

- m) Realiza un script llamado **comprime**, el cuál solicite que escriba la dirección absoluta de un archivo.
- SI el archivo existe
  - ENTONCES
    - o SI la carpeta "comprimidos" existe
    - o ENTONCES guarda el archivo comprimido en esa carpeta
    - o SINO crea la carpeta "comprimidos" y guarda el archivo
  - SINO sale un aviso de que ese archivo no existe

Nota: al comprimirlo no se debe eliminar el archivo original.

```
#!/bin/bash
echo Introduce la dirección absoluta de un archivo
read archi
if [ -f $archi ]
then
    if [ -d /home/utele/comprimidos ]
    then
        gzip -c $archi > /home/utele/comprimidos/abc.gz
    else
        mkdir /home/utele/comprimidos
        gzip -c $archi > /home/utele/comprimidos/abc.gz
    fi
else
    echo EL ARCHIVO $archi NO EXISTE.
fi
```

## 11. PERMISOS Y ATRIBUTOS

Al igual que en Windows, en Linux, los usuarios disponen de ciertos permisos o privilegios que limitan su control sobre el sistema.

En modo comando, para saber los atributos de un archivo, así como los permisos que un usuario tiene sobre éste, podemos observar el resultado de la ejecución de la orden `ls -l`.

```
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
```

Diagrama de anotación de la salida de `ls -l`:

- Tipo de Archivo: drwxr-xr-x
- PERMISOS DE PROPIETARIO: rwx
- PERMISOS DE GRUPO: r-x
- PERMISOS DE OTROS: r-x
- Enlaces duros: 3
- Propietario: raul
- Grupo: raul
- Tamaño (en bytes): 4096
- Fecha y hora de la última modificación: 2005-02-16 14:47
- Nombre: Desktop

Concretamente, los tipos de permisos sobre archivos en Linux son los siguientes:

- **Lectura:** Permite visualizar el contenido del archivo (necesario, por ejemplo, con los comandos `ls`, `cat` o `cp`).
- **Escritura:** Permite modificar el contenido del archivo (necesario para editar el contenido de un fichero, por ejemplo, con `gedit`).
- **Ejecución:** Permite ejecutar el archivo como si de un programa ejecutable se tratase. Estos permisos se suelen asignar a archivos Shell, es decir, archivos que realizan funciones propias del sistema operativo, como copias de seguridad, análisis de la integridad del sistema, etc. y Shell scripts creados por los usuarios.

Permisos (ls -l)	
-	Sin permiso
r	Lectura
w	Escritura
x	Ejecución

Ejemplo: `rwX rwx r--`

El propietario y su grupo, tienen permisos de lectura, escritura y ejecución. El resto de usuarios, de lectura.

Los permisos "totales" de un recurso constan de nueve indicadores, donde los tres primeros indican los permisos asociados al **usuario** dueño, los otros tres al **grupo** y los últimos 3 a los **otros**, al resto de los usuarios.

### Permisos en modo comando:

El comando que permite establecer permisos en un archivo o directorio es `chmod`. Podemos usarlo de dos formas:

a) En modo simbólico:

`chmod u|g|o|a [=|+|- r|w|x fichero(s)|directorio(s)]`

Ejemplos:

`chmod u=rwx, g=rx, o=- ej1.txt`

`chmod g-wx ej1.txt`

b) En octal:

Los tres tipos de permisos mencionados también se pueden representar en octal, de modo que se utiliza ``1" cuando se otorga permiso y ``0" para cuando se niega. A continuación, se transforma la representación binaria así obtenida en octal agrupándose por tríos, de la forma rwx. De este modo, la combinación de los tres tipos de permisos para un tipo de usuario irá desde cero (ningún permiso) hasta siete (todos los permisos).

Ejemplos:      rwx      = 111 (7 en octal) (permisos concedidos de lectura, escritura y ejecución)  
                 r-x      = 101 (5 en octal) (permisos concedidos de lectura y ejecución).  
                 r--      = 100 (4 en octal) (permiso concedido de lectura únicamente).  
                 ---      = 000 (0 en octal) (Ningún permiso concedido).

Para establecer en modo comando los permisos en octal usaremos `chmod XXX fichero(s)|directorio(s)`

Ejemplo: `chmod 750 ej1.txt`

Ejemplos:

Permisos (ugo)			Valor octal	Permisos al usuario	Permisos al grupo	Permisos a otros
rw-	rw-	rw-	6 6 6	Lectura y escritura	Lectura y escritura	Lectura y escritura
rwX	rwX	---	7 7 0	Todos	Todos	Ninguno
rw-	r--	r--	6 4 4	Lectura y escritura	Lectura	Lectura
rwX	r-x	---	7 5 0	Todos	Lectura y ejecución	Ninguno
r--	---	---	4 0 0	Lectura	Ninguno	Ninguno

Sólo el dueño de un recurso tendrá derecho a cambiar sus permisos, además del root por supuesto.

### Permisos especiales:

Hay una serie de permisos especiales sobre el sistema de archivos de Linux que pueden resultarnos útiles para determinadas tareas o para organizar directorios colaborativos entre diferentes usuarios.

#### - SUID:

El bit SUID activo en un archivo significa que el que lo ejecute va a tener los mismos permisos que el que creó el archivo.

Ejemplo:

```
$ ls -l /bin/su
-rwsr-xr-x 1 root root 31012 2016-03-04 07:49 /bin/su
```

Esto puede llegar a ser muy útil en algunas situaciones pero hay que utilizarlo con cuidado, dado que puede generar grandes problemas de seguridad. Para que sea efectivo el archivo **debe tener permisos de ejecución**.

`chmod 4775 <fichero>`

Para otorgarlo: `chmod u+s <fichero>`

Para quitarlo: `chmod u-s <fichero>`

Si se le quita el permiso de ejecución al archivo, en los permisos se reemplaza la **s** por la **S**. En este caso no tiene efecto el bit.

- **SGID:**

El SGID es lo mismo que en el SUID, pero a nivel de grupo. Es decir, todo archivo que tenga activo el SGID, al ser ejecutado, tendrá los privilegios del grupo al que pertenece.

Ejemplo:

```
$ ls -l
drwxrws--- 2 mar mar      4096 2016-03-04 21:27 compartido
```

Esto es muy usado cuando queremos configurar un directorio colaborativo: si aplicamos este bit al directorio, cualquier archivo creado en dicho directorio, tendrá asignado el grupo al que pertenece el directorio.

chmod **2775** <directorio>

Para otorgarlo: chmod g+s <directorio>

Para quitarlo: chmod g-s <directorio>

Al igual que con el SUID, aparece una **s** o una **S** reemplazando la **x** en los permisos del grupo.

- **Sticky bit:**

El Sticky bit se utiliza para permitir que cualquiera pueda escribir y modificar sobre un archivo o directorio, pero que solo su propietario o root pueda eliminarlo.

Ejemplo:

```
drwxr-xr-t 13 root root  4096 2016-04-24 20:55 tmp
```

Un ejemplo de uso es el directorio /tmp, que debe tener permisos para ser utilizado por cualquier proceso, pero solo el dueño o root puede eliminar los archivos que crea. Para aplicar el Sticky bit a un directorio:

chmod **1755** <nombre\_directorio>

Para otorgarlo: chmod o+t <directorio>

Para quitarlo: chmod o-t <directorio>

Al directorio con el Sticky Bit aplicado se le agrega la **t** al final del descriptor de permisos. Una vez agregado se puede comprobar creando diferentes archivos con diferentes usuarios y probar a eliminar archivos de diferentes usuarios. No permite la eliminación de éstos.

### Permisos en modo gráfico:

En entorno gráfico, la asignación, modificación y eliminación de permisos o atributos sobre los archivos se realiza de forma análoga a Windows: botón derecho sobre el directorio o archivo, en **Propiedades**, en la pestaña de **Permisos**.



## 12. APLICACIONES

### 12.1. AGREGAR O QUITAR PROGRAMAS

El directorio **/usr** de Linux aloja el **software instalado** en el sistema. Concretamente:

- **/usr/bin** contiene ejecutables
- **/usr/sbin** contiene ejecutables en modo root
- **/usr/lib** contiene librerías
- **/usr/share** contiene documentación de los paquetes

Un **paquete** es un conjunto de ficheros que se utilizan para instalar una aplicación. Los paquetes que contienen código máquina y no código fuente, se les llama paquetes binarios. Las distribuciones GNU/Linux poseen un sistema de administración de paquetes que permite encontrar, descargar e instalar programas.

Existen tres **tipos principales de paquetes** o métodos de distribución:

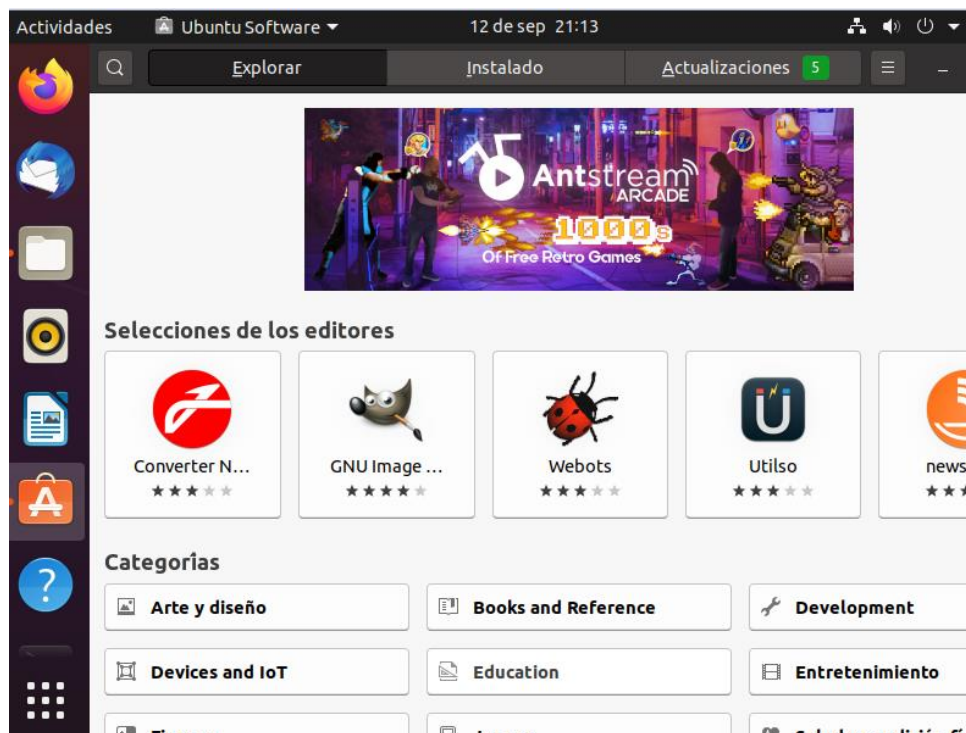
- **Deb** o paquetes Debian
- **Rpm** o paquetes RedHat
- **Tarball**, que es una colección de ficheros agrupados mediante la utilidad tar y comprimidos mediante gzip.

Los deb y rpm han sido adoptados por la mayoría de distribuciones.

Respecto a la gestión del software de Linux, ésta se puede realizar a través de la interfaz gráfica o mediante comandos.

#### 12.1.1. Modo gráfico: Ubuntu Software

Para gestionar paquetes software en Ubuntu mediante interfaz gráfica podemos usar **Ubuntu Software**:



En esta aplicación tienes todo ordenado por categorías, te da información sobre las aplicaciones, sugerencias, plugins o paquetes complementarios, comentarios de otros usuarios... Resulta muy sencillo de utilizar para usuarios con poca experiencia.

### 12.1.2. Modo comando: apt, aptitude, dpkg...

Para gestionar paquetes software en Ubuntu mediante comando podemos usar:

- **Instalación de paquetes deb:**

- **Comando apt:**

- **apt-get update:** actualiza la lista de paquetes disponibles para instalar desde los repositorios del fichero sources.list
- **apt-cache search búsqueda:** si no conocemos el nombre exacto del paquete a instalar
- **apt-get install paquete:** instala paquetes por la red
- **apt-get install --reinstall paquete:** reinstala un paquete
- **apt-get remove paquete:** desinstala paquetes
- **apt-get upgrade:** instala las nuevas versiones de los paquetes ya instalados
- **apt-get purge paquete:** elimina el paquete y sus archivos de configuración

- **Aptitude** es una versión mejorada de apt y se usa con las mismas opciones que este.

- **Comando dpkg:**

- **dpkg -i paquete:** instala el paquete
- **dpkg -r paquete:** desinstala un paquete
- **dpkg --purge paquete:** desinstala un paquete y sus archivos de configuración

- **Instalación de paquetes tgz:**

Los paquetes tgz contienen el código fuente del programa con los archivos necesarios para compilar e instalar el programa. Estos paquetes están un tipo de fichero .tar.gz o tar.bz2.

En primer lugar, mediante los **comandos tar** (para des/empaquetar) y **gzip** o **bzip** (para des/comprimir) desempaquetamos y descomprimos. A continuación, el paquete se instala desde el directorio donde están los fuentes del programa con los siguientes comandos:

- **./configure**
- **make**
- **make install**

- **Instalación de paquetes rpm:**

Para los paquetes rpm en modo gráfico tenemos las utilidades yast y yum

En modo texto, utilizamos el comando rpm (rpm [-opciones] [paquetes])

- **rpm -i:** instala un paquete
- **rpm -u:** actualiza los paquetes ya instalados
- **rpm -e:** desinstala el paquete

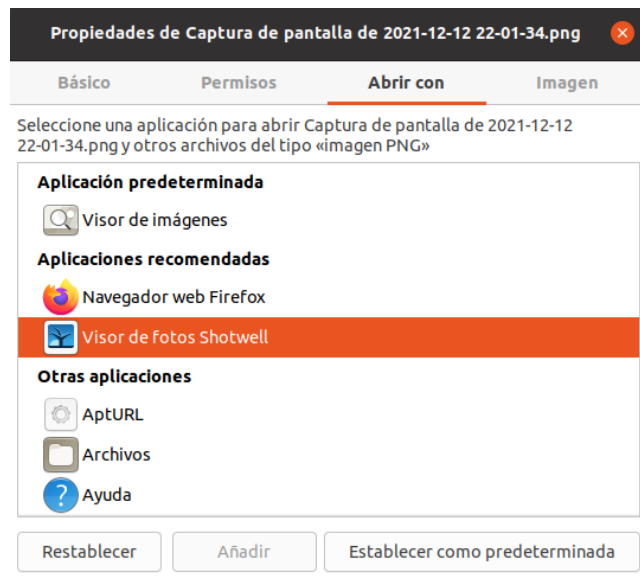
Ojo! Estos métodos de instalación de aplicaciones **no puedes ejecutarlos de forma simultánea** (tampoco apt-get desde la terminal), ya que la primera aplicación que se ejecuta toma el control de ciertos archivos y los bloquea para que ninguna otra aplicación trastoque tu instalación. Esto último también se aplica al proceso de actualización automática.



## 12.2. ASOCIAR UNA APLICACIÓN A UN TIPO DE ARCHIVO

Un tipo de archivo lo podemos abrir con otra aplicación distinta de la que tiene asociada para que se abra con ella. Lo hacemos haciendo clic con el botón derecho sobre el archivo → Abrir con otra aplicación...

Si queremos asociar la aplicación al tipo de fichero para que se abra con ella de forma predeterminada haremos clic con el botón derecho → Propiedades → pestaña “Abrir con”. Seleccionaremos la aplicación y pulsaremos en “Establecer como predeterminada”.



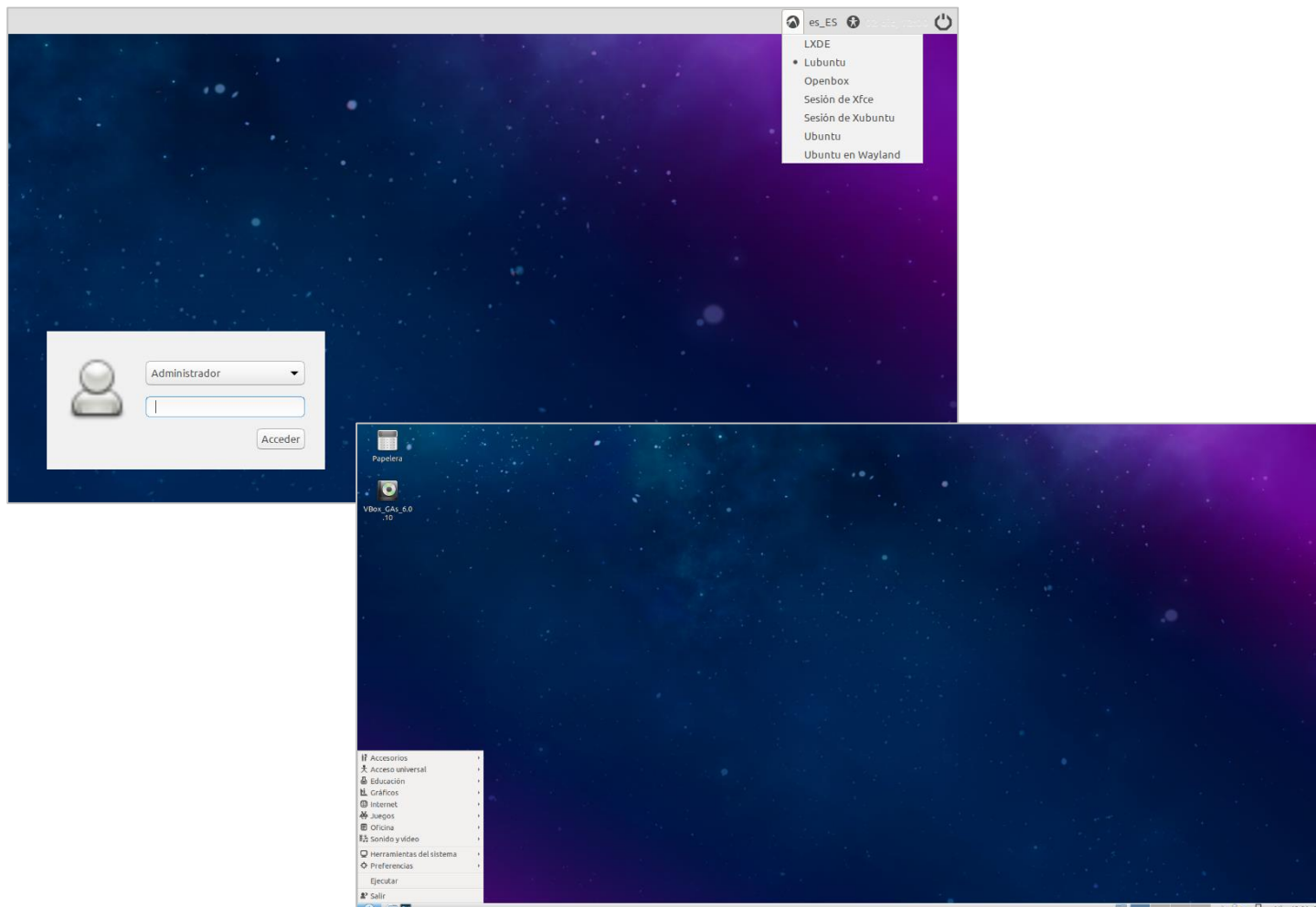
## 12.3. INSTALACIÓN DE ENTORNOS DE ESCRITORIO EN LINUX

Ya has visto el entorno de escritorio por defecto en Ubuntu: Gnome

Ahora vamos a instalar LXDE y XFCE, dos entornos de escritorio que consumen muy pocos recursos.

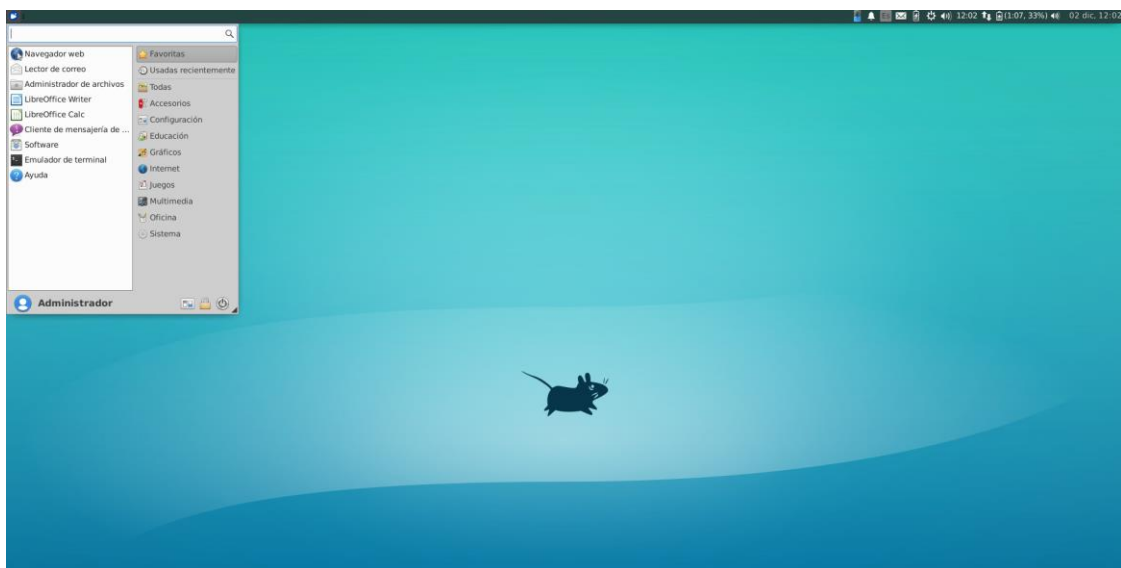
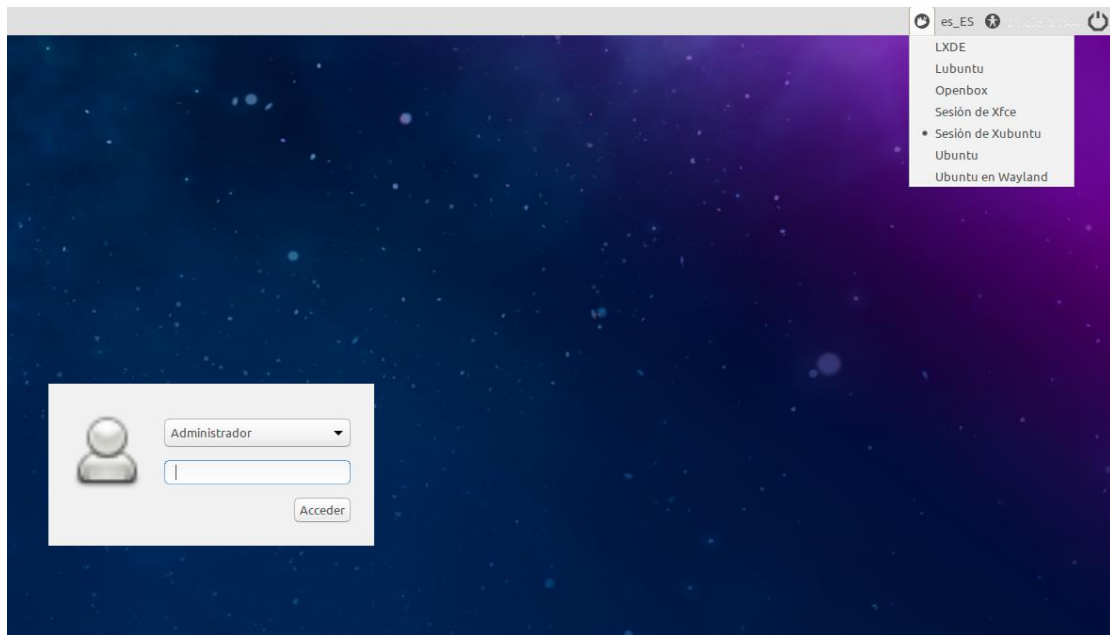
### a) Instalación de LXDE:

```
sudo apt-get install lubuntu-core
```



## b) Instalación de XFCE:

```
sudo apt-get install xubuntu-desktop
```



## 13. ANEXOS

---

### 13.1. INSTALAR GUEST ADDITIONS EN UBUNTU 20.04

Ya sabréis que las Guest Additions de VirtualBox es un complemento indispensable que es necesario instalar para poder usar la máquina virtual a pantalla completa, para trabajar visualmente de una manera más cómoda.

Para instalar las Guest Additions en una MV con Ubuntu 20.04, abrimos un terminal y ejecutamos los siguientes comandos:

- `sudo apt update`
- `sudo add-apt-repository multiverse`
- `sudo apt install virtualbox-guest-dkms virtualbox-guest-x11`

También puedes hacerlo en modo gráfico, desde VirtualBox, haciendo Dispositivos→ Insertar imagen del CD de las <<Guest Additions>>.

Tras esto, solo tenemos que reiniciar la máquina virtual, y las guest additions quedaran instaladas en nuestro sistema Ubuntu virtualizado.