

FPI.CFGS: DESARROLLO DE APLICACIONES MULTIPLATAFORMA

SISTEMAS INFORMÁTICOS

UT06 - Introducción a los SSOO GNU/Linux.

Índice

1. Introducción.....	2
1.1. Distribuciones.....	2
1.2.Licencias de software.....	3
2. Instalación.....	3
3. Entornos gráficos de escritorio.....	9
4. Primeros pasos.....	10
4.1. Intérprete de comandos.....	10
4.2. Estructura de directorios.....	13
4.3. Instalar y quitar componentes.....	14
4.3.1. Synaptic.....	15
4.3.2. apt-get.....	15
4.3.3. Aptitude.....	16
4.3.4. Instalación manual.....	17
5. Administración de usuarios.....	18
6.Sistema de ficheros.....	19
6.1. Particionamiento.....	20
6.1.1. Herramientas gráficas.....	20
6.2. Monitorización	21
7. Permisos.....	21
7.1. Establecer los permisos.....	22
8. Procesos.....	23
8.1. Programación de tareas.....	24
8.2. El comando tar.....	25

1. Introducción

Linux fue concebido por el finlandés Linus Torvalds, estudiante de la Universidad de Helsinki, quien comenzó trabajando sobre el código fuente de Minix (un pequeño UNIX desarrollado por Andy Tanenbaum) para lograr un Unix mínimo, capaz de ejecutar al menos un shell y un compilador. Primero fue la versión 0.02 ya que la 0.01 nunca llegó a ser compilada con éxito. Luego Linus anunció en Internet su proyecto de la siguiente manera:

"Si suspiras al recordar aquellos días cuando los hombres eran hombres y escribían sus propios manejadores (drivers). Si te sientes sin ningún proyecto interesante y te gustaría tener un verdadero sistema operativo que pudieras modificar a placer. Si te resulta frustrante tener sólo Minix. Entonces este artículo es para ti".

De esa forma Linux fue liberado en Internet y la respuesta de los programadores y usuarios de UNIX fue contundente. Pronto todos querían aportar sus conocimientos para que Linux se convirtiera en un sistema operativo estable, robusto y potente. Finalmente llegó la primera versión estable del Kernel, la versión 1.0. De allí en adelante, Linux fue evolucionando a un ritmo vertiginoso hasta convertirse en un fuerte rival de los sistemas operativos comerciales.

Desde su aparición los sistemas GNU/Linux han ido evolucionando y mejorando sus prestaciones drásticamente. Hoy en día puede encontrar los sistemas GNU/Linux en multitud de sistemas: desde grandes servidores hasta pequeños equipos domésticos como teléfonos móviles.

1.1. Distribuciones

El kernel del sistema operativo Linux fue desarrollado por Linus Torvalds y licenciado bajo GPL. Pero un sistema GNU/Linux no es sólo su núcleo ya que existen hoy en día un gran número de aplicaciones desarrolladas también bajo licencias libres que permiten que los sistemas GNU/Linux tengan una gran versatilidad y funcionalidad.

Existen muchas aplicaciones desarrolladas bajo licencia libre pero, sin duda alguna, las más importantes son: OpenOffice, Apache, Firefox, ... Estas son solamente unos ejemplos, pero existen aplicaciones libres para cualquier uso que imagines.

Puesto que el kernel y las aplicaciones son libres entonces puedes crear tu propio sistema operativo con los programas que más te gusten y crear tu propia distribución; lógicamente nuestra distribución también tendrá que ser libre. Una distribución es una agrupación de un conjunto de programas, imágenes, temas de escritorio, etcétera.

Si deseas hacer tu propia distribución GNU/Linux es recomendable que accedas a www.installinux.com donde a través de un asistente podrás personalizar y crear tu propia distribución.

Existen muchas iniciativas tanto empresariales como gubernamentales para crear su propia distribución. Un claro ejemplo lo puedes encontrar en las distribuciones Guadalinux (de la Junta de Andalucía), gnuLinux (de Extremadura), trippbox (distribución empresarial de telefonía IP), etcétera.

A partir de la libertad de los usuarios, empresas y organismos para personalizar su propia distribución han surgido una gran cantidad de distribuciones que nacen, evolucionan, derivan en otras distribuciones y como no, algunas mueren.

En esta web <http://distrowatch.com/> puedes encontrar una recopilación de casi todas las distribuciones existentes. Echándole un vistazo, encontrarás distribuciones como: Debian, Fedora, Ubuntu, Gentoo, ...

1.2.Licencias de software

Sin duda alguna no se puede hablar de los sistemas GNU/Linux sin mencionar las licencias de software libre, germen de todo el desarrollo de los sistemas GNU/Linux.

La Licencia Pública General de GNU, más conocida como GNU GPL, es una licencia creada por la Free Software Foundation en 1989. Su principal objetivo es garantizar la libertad de compartir y modificar el software. El término libre (free en inglés) se refiere a la libertad de poder modificar y distribuir el software, no a su precio.

Al desarrollar un programa, escribir un artículo o crear cualquier obra que se desea distribuir libremente, puedes optar por dos caminos diferentes: dominio público o licencia libre. Lógicamente, las dos opciones permiten liberar el software garantizando la autoría de la obra.

Si creas una aplicación para dominio público, el software estará disponible de forma gratuita para otras personas pero el código fuente no estará disponible. Por el contrario, si eliges una licencia libre, entonces el proceso es mucho más enriquecedor ya que pone disponible el código fuente por lo que permite que otros programadores puedan modificar, mejorar o adaptar nuestro software a sus necesidades.

Cuando se licencia un software bajo GPL se permiten los siguientes grados de libertades:

- Libertad 0. Ejecutar el programa sea cuál sea nuestro propósito.
- Libertad 1. Estudiar el funcionamiento del programa y adaptarlo a tus necesidades.
- Libertad 2. Redistribuir copias.
- Libertad 3. Mejorar el programa y luego distribuirlo.

Lógicamente, los términos de la licencia permiten la libertad de poder utilizar y adaptar el software a sus necesidades pero el resultado debe seguir siendo libre. Este punto es muy importante ya que permite que no se “rompa” la cadena de software libre.

Las licencias Creative Commons están inspiradas en la licencia GPL, pero están destinadas a facilitar el uso y distribución de los contenidos garantizando la autoría de la obra.

2. Instalación

Para aprender a instalar un sistema GNU/Linux, usaremos Ubuntu por su amplia difusión, facilidad a la hora de instalarlo en cualquier ordenador, y por sus posibilidades para instalar software adicional

El proceso de instalación del sistema GNU/Linux resulta bastante sencillo gracias al asistente que te guía durante todo el proceso de instalación. Antes de iniciar la instalación necesita tener en cuenta el uso que le vas a dar al sistema ya que de ello dependerá mucho el hardware del equipo. Como regla general necesitas un equipo con al menos 512 Megabytes de RAM y unos 5 Gigabytes de disco duro. Aunque todo depende del uso que quiera darle al sistema.

En Ubuntu existen tres versiones: Desktop (para equipos de escritorio), Netbook (para portátiles) y Server (para servidores). Además, para cada tipo puedes utilizar la versión de 32 bits y de 64 bits. Como el objetivo de la unidad es aprender a instalar un sistema operativo GNU/Linux, realizaremos la instalación de la versión Desktop.

Para iniciar la instalación debes iniciar el equipo con el CD de la distribución. Si el equipo no muestra la pantalla de inicio (véase la figura 1) puedes entrar en la BIOS del equipo y configurarla para que arranque el sistema directamente desde CD.

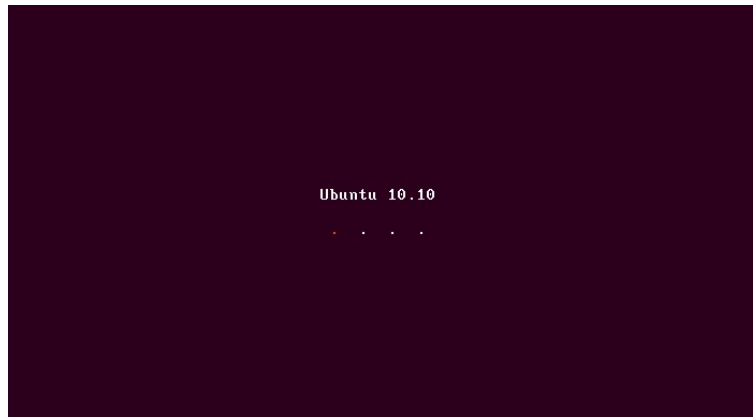


Figura 1. Pantalla de inicio

Una vez iniciado el sistema, aparece la pantalla de bienvenida donde debes seleccionar el idioma Español e indicar si quieres Probar Ubuntu o si deseas Instalar Ubuntu. En nuestro caso vamos a realizar la instalación de Ubuntu.



Figura 2. Pantalla de bienvenida

Para iniciar la instalación, el sistema muestra una pantalla con los requisitos que debe tener el sistema. Tal y como muestra la figura 3 estos requisitos son: tener al menos 2 Gigabytes de espacio de disco duro, tener conexión a Internet y en el caso de tratarse de un portátil, estar conectado a una toma de corriente. Además, podemos indicarle al sistema que descargue automáticamente las actualizaciones del sistema mientras que se realiza la instalación. En nuestro caso, pulsamos Adelante.

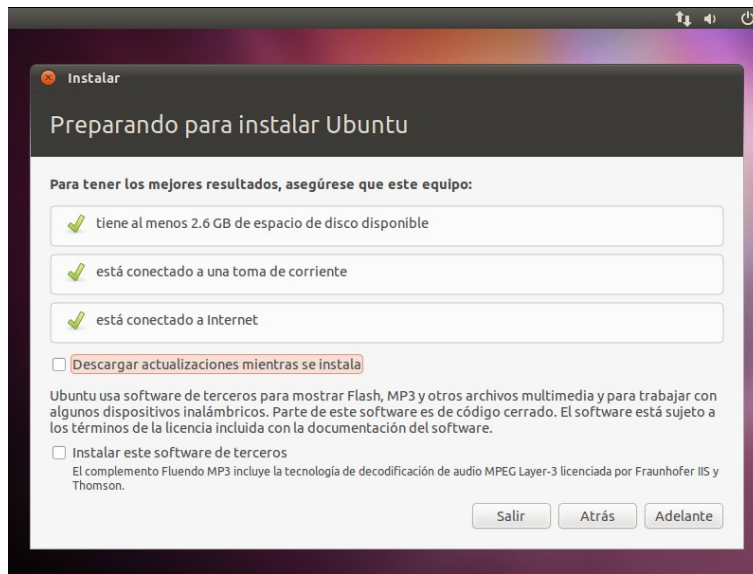


Figura 3. Requisitos del sistema

El siguiente paso es realizar el particionamiento del sistema. Para ello puedes seleccionar la opción Borrar y usar el disco entero en el caso de querer hacer el particionamiento automático, o puedes seleccionar Especificar particiones manualmente. Para no perder la información del sistema vamos a seleccionar la opción Especificar manualmente y pulsar Adelante.

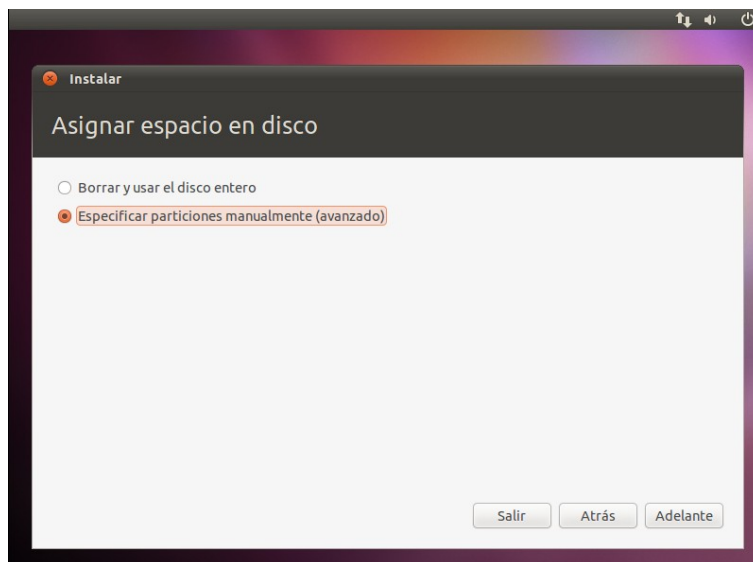


Figura 4. Asignar espacio en disco

Para realizar el particionamiento del sistema debes tener en cuenta que hay que crear la partición donde se guardan los datos (/) y la partición swap. La partición swap o área de intercambio, debe tener al menos el doble del tamaño de la memoria RAM del sistema.

Para empezar el proceso de particionamiento pulsa en Nueva tabla de particiones.

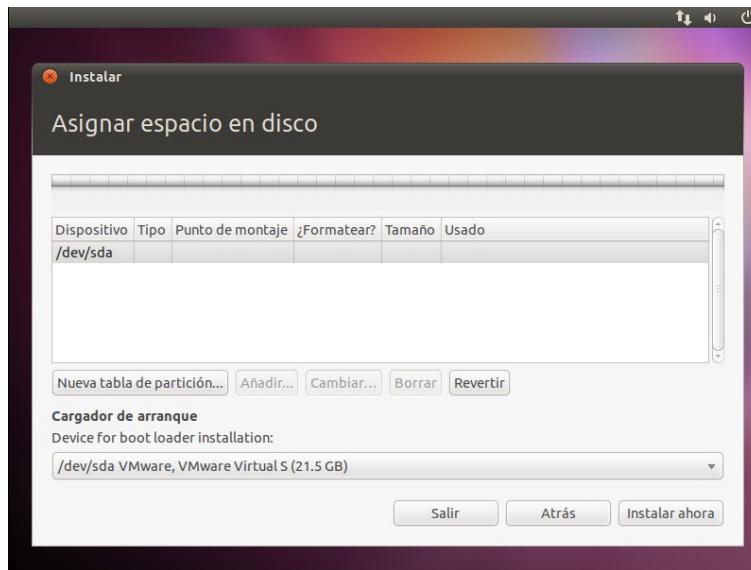


Figura 5. Particionamiento

Crea la partición del sistema de tipo ext4 con al menos 2 Gigabytes con punto de montaje "/" y pulsa Aceptar.



Figura 6. Crear partición del sistema

Crea la partición de swap (área de intercambio) con el doble de tamaño que la memoria RAM del sistema.



Figura 7. Crear partición swap

Una vez creadas las dos particiones pulsa Instalar ahora.

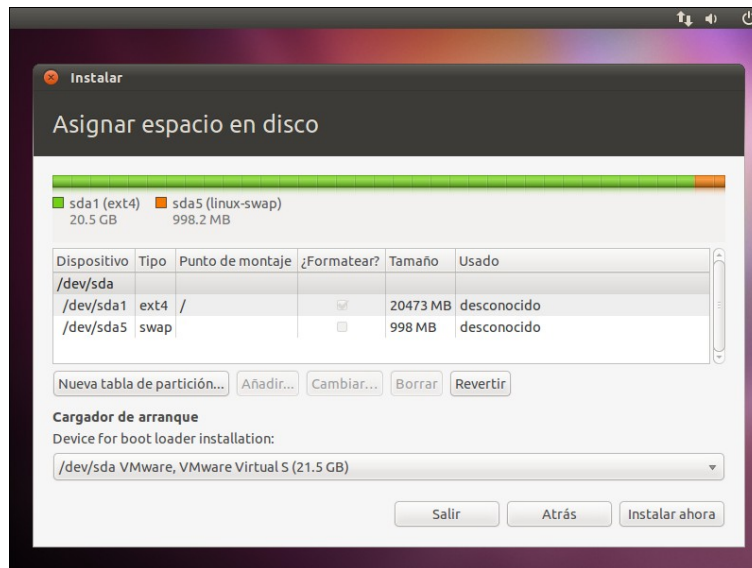


Figura 8. Tabla de particiones creada

Selecciona la zona horaria donde te encuentras y pulsa Adelante.



Figura 9. Zona horaria

Indica la distribución de tu teclado y pulse Adelante.

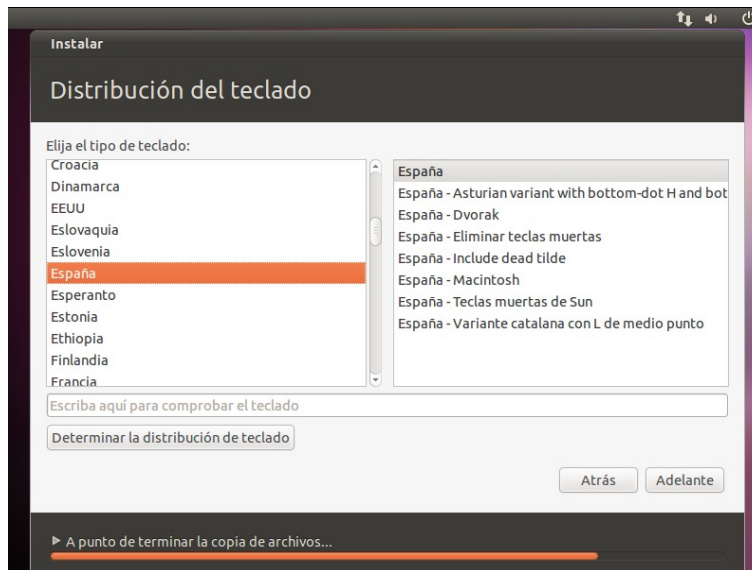


Figura 10. Distribución del teclado

A continuación se va a crear una cuenta de usuario. Para ello, introduce tu nombre, el nombre del equipo, el nombre de usuario y tu contraseña. Y para acabar pulsa Continuar.

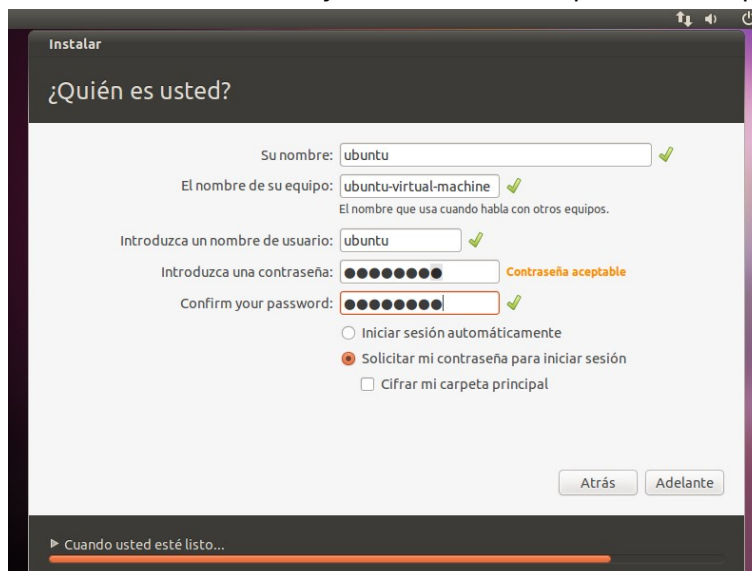


Figura 11. Usuario del sistema

Para finalizar la instalación pulsa Reiniciar Ahora.

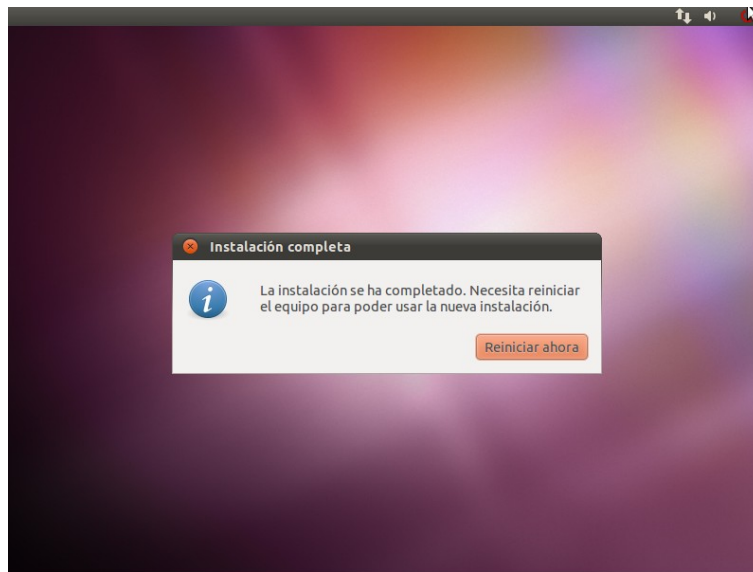


Figura 12. Instalación Completa

Una vez reiniciado el equipo, aparece la pantalla de inicio de sesión. Una vez introducido el nombre de usuario y contraseña ya puedes empezar a utilizar el sistema.

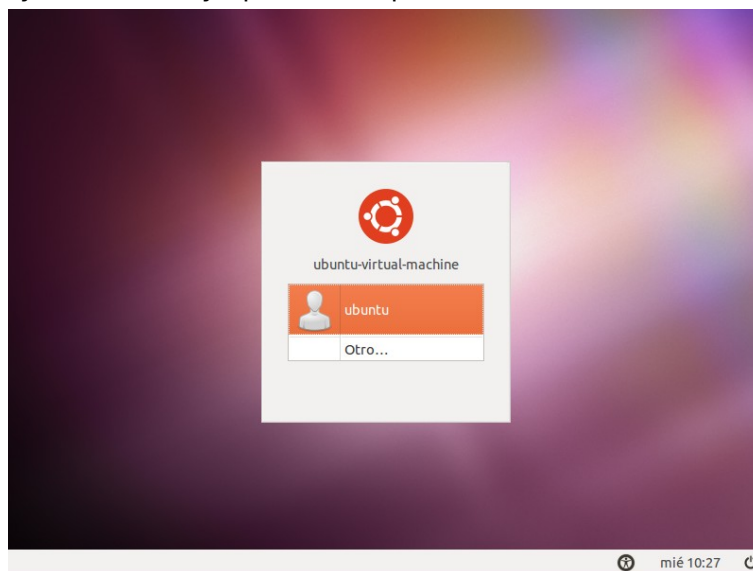


Figura 13. Iniciar sesión

3. Entornos gráficos de escritorio

Uno de los elementos que ha propiciado la gran expansión de los sistemas GNU/Linux en empresas y hogares es la utilización de entornos gráficos sencillos y amigables. Los sistemas GNU/Linux cuentan con diversos entornos gráficos, muy potentes, que permiten utilizar el sistema fácilmente.

X-Windows (o sistema de ventanas X en castellano) es el nombre por el que se conoce al entorno gráfico usado por los sistemas Unix. Desarrollado desde mediados de la década de los 80 en el MIT (Instituto Tecnológico de Massachussets) se encuentra actualmente en su versión 11, por lo que normalmente suele ser referenciado como X11. El grupo de desarrollo inicial ha ido dando lugar en el tiempo a diversos sucesores: X Consortium (desde 1994 a

1996), The Open Group (de 1997 a 1998), XFree86 (de 1992 a 2004) y X.org (desde 2004). X.org Foundation se fundó a partir de un grupo de desarrolladores y actualmente se encarga de desarrollar y coordinar el entorno X.

X-Windows proporciona una interfaz gráfica (GUI) al mundo de Linux. X-Windows, al igual que el sistema operativo Windows de Microsoft, ofrece una forma de manejo de algunos de los elementos de interacción más comunes como ventanas, cuadros de diálogo, botones y menús. X-Windows es quien proporciona las capacidades gráficas que hacen de las plataformas basadas en Linux la elección para el desarrollo de muchas aplicaciones de ingeniería y diseño, y es x-Windows lo que hace posible que Linux sea un competidor serio en el mercado de los sistemas operativos para PC.

De forma simple, se puede decir que x-Windows es una interfaz gráfica completa para Linux y, por extensión, para Unix. X-Windows está compuesto por dos elementos principales: el servidor X, encargado de mostrar visualmente los elementos en la pantalla de forma totalmente independiente del sistema operativo, y el gestor de ventanas, cuyo objetivo es la gestión y administración de las ventanas mostradas para las aplicaciones, su apariencia, creación, colocación en la pantalla, etcétera. De esta forma x-Windows es capaz de distribuir el procesamiento de las aplicaciones siguiendo el paradigma cliente-servidor: el servidor provee los servicios para acceder a teclado, ratón y pantalla, mientras que los clientes son las aplicaciones que toman estos recursos para poder interactuar con los usuarios.

Esta forma de diseño en dos partes es lo que origina que existan diferentes implementaciones de gestores de ventanas, con diferentes características funcionales y visuales. El servidor X, como puedes imaginar es altamente portable y en el caso de Ubuntu permite utilizar los tres principales entornos de escritorio o GUI (Graphical User Interfaces o Interfaces Gráficas de Usuario). **GNOME** está orientado a la simplicidad, **KDE** ofrece un mayor conjunto de aplicaciones así como posibilidades de caracterización por defecto, o **Xfce** está optimizado para su uso con requisitos hardware bajos. Aparte de los citados, es posible utilizar otros entornos de escritorio como Fluxbox, Sugar o LXDE.

4. Primeros pasos

Una de las grandes ventajas de los sistemas GNU/Linux es que se adapta completamente al nivel de conocimientos del usuario. Hoy en día, a través de los asistentes y los entornos gráficos es posible utilizar fácilmente los sistemas GNU/Linux sin necesidad de tener amplios conocimientos sobre el sistema.

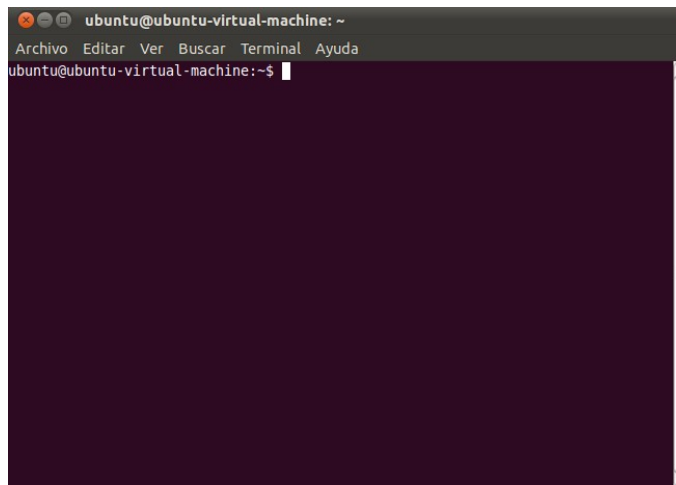
Por supuesto, cuantos más conocimientos tengas mejor puedes aprovechar las prestaciones del sistema. A continuación se van a comentar las tareas más frecuentes en los sistemas GNU/Linux.

4.1. Intérprete de comandos

El intérprete de comandos o shell del sistema es la interfaz entre el usuario y el sistema operativo. La función del shell es recibir las órdenes del usuario a través de la línea de comandos, interpretarlas, ejecutarlas y mostrar su resultado.

Resulta muy útil aprender a utilizar el shell del sistema ya que aunque al principio puede parecer un poco difícil, resulta fundamental para obtener el máximo rendimiento del sistema. El shell permite interactuar directamente con el sistema y con sus ficheros de configuración.

Para iniciar el intérprete de comandos tienes que ir al menú Aplicaciones / Accesorios, ejecutar la herramienta Terminal y aparecerá el terminal del sistema.



Una vez que accedes al sistema se muestra un prompt con el siguiente aspecto o parecido:

```
usuario@ubuntu-virtual-machine:~$
```

donde usuario es el nombre del usuario que está utilizando el sistema, @ubuntu-virtual-machine indica el nombre del equipo. A continuación, se muestra el directorio en el que se encuentra. En el caso de que se encuentre el carácter ~ es porque está en el directorio home. Por último, el símbolo \$ o # indica si es un usuario normal (\$) o es el administrador del sistema (#).

El usuario root es el administrador del sistema y puede realizar cualquier tarea de administración. En algunas distribuciones puede acceder directamente al sistema como usuario root, pero otras distribuciones, como Ubuntu, te obligan a acceder al sistema con un usuario sin privilegios de administrador y luego cambiar de usuario.

Si deseas ejecutar una tarea de forma puntual como root puedes utilizar el comando sudo de la siguiente forma:

```
$ sudo <comando>
```

Si necesita ejecutar múltiples tareas puedes obtener un shell de root ejecutando sudo bash o su:

```
$ sudo bash  
#
```

Además, si lo deseas, puedes activar la cuenta de root al establecer su contraseña:

```
$ sudo passwd root
```

Aunque a lo largo del curso aprenderemos a utilizar el shell del sistema, a continuación puedes ver los comandos más utilizados en los sistemas GNU/Linux.

Manipulación de directorios

cd <ruta>	Cambia de directorio.
cp <origen> <destino>	Copia ficheros o directorios.
mv <origen> <destino>	Mueve o cambia el nombre de un fichero o directorio.
rm <fichero>	Borra un fichero o directorio.
rmdir <directorio>	Borra un directorio.

cd <directorio>	Cambia de directorio.
pwd	Muestra el directorio actual de trabajo.
mkdir <directorio>	Crea un directorio.
tree	Muestra de forma gráfica la estructura de un directorio.
find	Permite buscar ficheros en el sistema.
locate	
ls	Muestra el contenido de un directorio.

Manipulación de ficheros

touch <fichero>	Crea un fichero vacío.
less <fichero>	Muestra el contenido de un fichero.
more <fichero>	
cat <fichero>	
mv <origen> <destino>	Mueve o cambia el nombre de un fichero o directorio.
cp <origen> <destino>	Copia ficheros o directorios.
rm <fichero>	Borra un fichero o directorio.
locate <fichero>	Busca un fichero o directorio en nuestro equipo.

Particionamiento

fdisk	Permite administrar las particiones del sistema.
fsck	Permite comprobar el estado de un sistema de ficheros.
mkfs	Permite formatear un sistema de ficheros.
df	Indica el espacio libre de un sistema de ficheros.
du	Indica el espacio utilizado por un usuario en el sistema de ficheros.
mount	Permite montar sistemas de ficheros.
umount	Permite desmontar sistemas de ficheros.

Comandos generales

startx	Inicia el modo gráfico.
halt	Apaga el equipo.
reboot	Reinicia el equipo.
date	Muestra y permite cambiar la fecha del sistema.
clear	Borra la pantalla.
man	Permite obtener ayuda del sistema.

Procesos

ps	Muestra los procesos activos del sistema.
top	Muestra los procesos del sistema y su rendimiento.
kill	Permite matar un proceso a partir de su PID.
pkill	Permite matar un proceso a partir de su nombre.

Permisos

chmod <permisos> <fichero/directorio>	Establece los permisos de un fichero o directorio.
chown <usuario> <fichero/directorio>	Cambia el usuario propietario de un fichero o directorio.
chgrp <grupo> <fichero/directorio>	Cambia el grupo propietario de un fichero o directorio.

Redes

ifconfig	Permite obtener información y configurar los adaptadores de red.
iwconfig	Permite obtener información y configurar los adaptadores de red inalámbrica.
ping <host>	Permite realizar un ping para comprobar la comunicación con un equipo.
route	Muestra y configura la tabla de enrutado del sistema.
iptables	Muestra y configura el cortafuegos del sistema.
service	Permite administrar los servicios del sistema.

Usuarios

adduser <usuario>	Da de alta un usuario.
userdel <usuario>	Borra un usuario.
usermod	Permite modificar las propiedades de un usuario.
passwd	Cambia la contraseña de un usuario.
addgroup	Permite dar de alta un usuario dentro de un grupo.
su	Permite cambiar de usuario.
sudo	Permite ejecutar un comando como <i>root</i> .
id	Muestra el usuario que se está utilizando.

Grupos

groups	Muestra los grupos a los que pertenece el usuario.
groupadd	Permite dar de alta a un grupo.
groupdel	Permite borrar un grupo de usuarios.

4.2. Estructura de directorios

GNU/Linux, al igual que UNIX, organiza la información del sistema en una estructura de árbol jerárquico de directorios compuesta de ficheros. Esta estructura se forma mediante un sistema de ficheros raíz (file system root) y un conjunto de sistemas de ficheros montables.

Un sistema de ficheros, o file system, es una estructura de directorios completa. Para poder utilizar un sistema de ficheros hay que montarlo; o sea, enlazarlo a la estructura de directorios ya existente. Los sistemas de ficheros se montan automáticamente cada vez que se inicia el sistema operativo. Cuando un usuario se conecta al sistema, se encuentra un único árbol de directorios formado por los distintos sistemas de ficheros que se encuentran montados en ese instante.

- **/** → raíz.
 - **/bin** → Comandos básicos: mkdir, more, cat, vi, sort, ps, date, ...
 - **/boot** → Ficheros del núcleo del SO y el configuración del grub
 - **/dev** → Dispositivos HW: /dev/sda, /dev/hdb, ...
 - **/etc** → Ficheros de configuración: passwd, hosts, ...
 - **/home** → Directorios home de los usuarios estándar.
 - /home/maria → Directorio home del usuario maría.
 - /home/pepe → Directorio home del usuario pepe.
 - **/lib** → Librerías para que se ejecuten los programas de /bin.
 - **/mnt** → Directorios de montaje de particiones de discos duros y dispositivos extraíbles (montar: mount y desmontar: umount).
 - /mnt/floppy →
 - /mnt/cdrom →
 - **/proc** → Ficheros especiales que envían/reciben información del kernel del sistema (entre otros, los procesos del sistema).
 - **/root** → Directorio home del root.
 - **/sbin** → Comandos de administrador: ifconfig, init, arp, mkfs, ...
 - **/temp** → Ficheros temporales del sistema.
 - **/usr** → Directorio de instalación de los programas de uso común para los usuarios.
 - /usr/X11R6 → Programas de Xwindows.
 - /usr/bin → Programas de uso general.
 - /usr/doc → Documentación del sistema.
 - /usr/etc → Ficheros de configuración general.
 - /usr/include → Ficheros de cabecera .h de librerías.
 - /usr/info → Ficheros de información GNU.
 - /usr/lib → Librerías generales.
 - /usr/man → Manuales man.
 - /usr/sbin → Programas de administración del sistema.
 - /usr/src → Código fuente de programas.
 - **/var** → Información temporal de programas (no borrrable).

4.3. Instalar y quitar componentes

En GNU/Linux se puede realizar la instalación de una aplicación directamente a partir del código fuente o a través de la aplicación compilada (paquete). A las aplicaciones

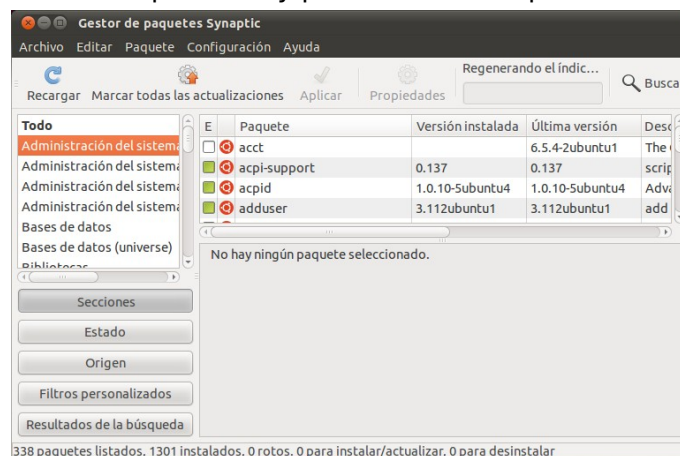
preempaquetadas se le denomina paquete y contienen los binarios, los archivos complementarios y archivos de configuración para poder ejecutarse.

Para facilitar el proceso de instalación se utilizan gestores de paquetes que facilitan la administración de los paquetes. A continuación, vas a aprender a instalar aplicaciones de todas las formas posibles: mediante x-Windows, mediante apt-get, directamente instalando paquetes .deb o a partir del código fuente. Las dos primeras formas son las más fáciles de utilizar y por lo tanto las recomendadas.

4.3.1. Synaptic

Synaptic es una herramienta de x-Windows que facilita las tareas de instalación y eliminación de software. Para utilizar la herramienta ejecuta Gestor de paquetes Synaptic que se encuentra en el submenú Administración dentro de Sistema.

Una vez iniciada la herramienta (véase la figura adjunta) el menú de la izquierda muestra las diferentes categorías de aplicaciones. Si pulsas en una categoría aparecen sus diferentes aplicaciones. Si una aplicación ya se encuentra instalada en el sistema su campo de selección se encuentra activo. Si deseas instalar o desinstalar una aplicación sólo debes seleccionar o deseleccionar la aplicación y pulsar el botón Aplicar.



Se pueden definir repositorios adicionales para aumentar la disponibilidad de software del sistema. Un repositorio es una ubicación de red que almacena paquetes de software junto a los metadatos que los describe. Los repositorios también se utilizarán con el comando apt-get que veremos más adelante.

4.3.2. apt-get

apt-get permite instalar o desinstalar por línea de comandos cualquier paquete. Para empezar, apt-get utiliza una serie de repositorios que se encuentran en el fichero /etc/apt/sources.list. Si lo deseas, puede modificar los repositorios del sistema y actualizar el sistema ejecutando:

```
# apt-get update
```

A continuación, se van a ver los procedimientos más utilizados

- Actualizar el sistema. Permite actualizar el sistema con todas las dependencias. Se utiliza:

```
# apt-get upgrade
```

- Búsquedas. Permite localizar un paquete o término en alguno de los repositorios. Se ejecuta:

```
# apt-cache search <nombre>
```

donde nombre indica el nombre del paquete que desea buscar.

- Consulta de información. Permite consultar información de un paquete.

```
# apt-cache show <paquete>
```

Por ejemplo si quiere información sobre el servidor web ejecute:

```
# apt-cache show apache2
```

- Instalación de paquetes. Permite realizar la instalación de paquetes con la resolución automática de dependencias.

```
# apt-get install <paquete>
```

Por ejemplo si desea instalar el servidor web ejecute:

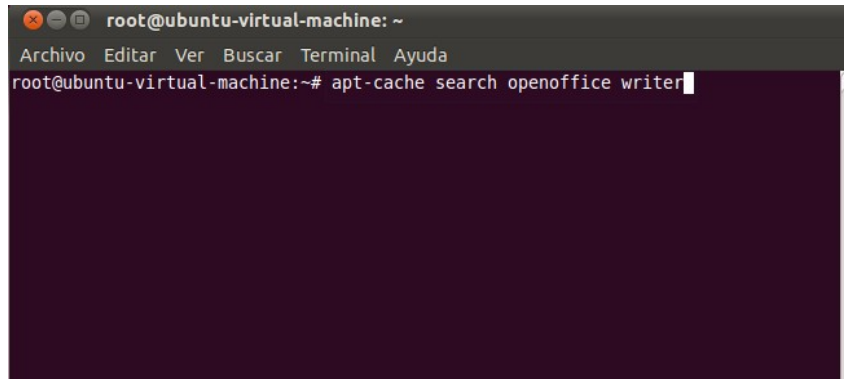
```
# apt-get install apache2
```

- Desinstalar un paquete. Para desinstalar un paquete hay que ejecutar:

```
# apt-get remove <paquete>
```

Por ejemplo si desea desinstalar el servidor web ejecute:

```
# apt-get remove apache2
```



A la hora de buscar o instalar un programa puede utilizar el carácter * para indicar cualquier carácter. Por ejemplo si desea instalar cualquier aplicación que empiece por php entonces ejecuta

```
# apt-get install php-*
```

4.3.3. Aptitude

aptitude es un gestor de paquetes por línea de comandos muy cómodo y sencillo de utilizar. Para poder utilizarlo antes necesitas instalarlo

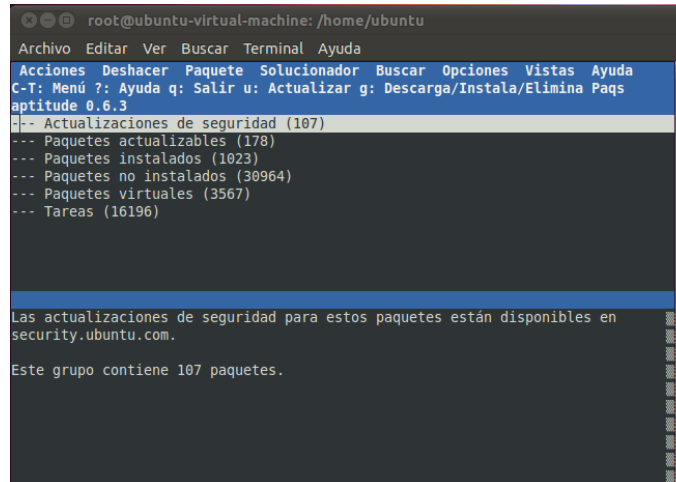
```
root@ubuntu ~ #apt-get install aptitude
```

y ejecutas en el terminal como root el comando:

```
root@ubuntu ~ #aptitude
```


Una vez ejecutada la herramienta puede acceder a las diferentes categorías e instalar o desinstalar el software.

aptitude "recuerda" las dependencias que ha instalado, y apt-get no. Así que, al desinstalar con aptitude, éste borra el paquete y sus dependencias (si no son requeridas por otro programa); apt-get (y Synaptic, que usa apt) sólo elimina el paquete que le indicas, y no sus dependencias. En resumen aptitude mantiene el sistema algo más limpio y optimizado.



4.3.4. Instalación manual

Si lo deseas puedes realizar la instalación o desinstalación directa de un paquete. Para realizar la instalación debes descargar previamente el paquete y ejecutar:

```
# dpkg -i nombre_paquete
```

Si por el contrario deseas eliminar un paquete, primero debes conocer su nombre exacto. Para ello debes ejecutar:

```
# dpkg-query -s nombre
```

Una vez que conoce el nombre exacto se realiza la instalación ejecutando:

```
# dpkg -r nombre_completo
```

A veces se encuentran aplicaciones que no proporcionan paquetes de instalación, y hay que compilar a partir del código fuente. Para ello, lo primero que debe realizar es instalar las herramientas de compilación ejecutando:

```
# apt-get install build-essential
```

Además, puedes realizar la **instalación directamente desde el código fuente**, es posible que surja algún problema de dependencias. Si sucede esto, entonces debe resolver la dependencia y continuar con el proceso de instalación.

En general, los pasos a seguir para compilar una aplicación son los siguientes:

- Descarga el código fuente.
- Descomprime el código, generalmente está empaquetado con `tar` y comprimido con `gzip` (*.tar.gz o *.tgz) o `bzip2` (*.tar.bz2).
- Accede a la carpeta creada al descomprimir el código.
- Ejecuta el script `./configure` que permite comprobar las características del sistema que afectan a la compilación y crear el archivo `makefile`.

- Compila el código ejecutando el comando `make`.
- Instala la aplicación en el sistema ejecutando `make install`. Si desea desinstalar la aplicación entonces ejecuta `make clean`.

5. Administración de usuarios

GNU/Linux es un sistema operativo multiusuario. Esto significa que permite a varios usuarios utilizar el sistema simultáneamente, a través de la línea de comandos o con conexiones remotas. GNU/Linux controla el acceso al equipo y a sus recursos a través de las cuentas de usuarios y grupos.

En los sistemas GNU/Linux existen tres tipos de usuarios:

- **Root.** Es el usuario más importante ya que es el administrador y dueño del sistema. Se aconseja utilizar la cuenta de root para las tareas específicas de administración y el resto del tiempo utilizar una cuenta de usuario normal.
- **Usuarios normales.** Son los usuarios que pueden iniciar sesión en el sistema y tienen una funcionalidad limitada, tanto en los comandos que pueden ejecutar, como a los ficheros a los que tienen acceso.
- **Usuarios asociados a servicios.** Este tipo de usuarios no pueden iniciar sesión en el sistema. Su utilización es muy útil ya que permiten establecer los privilegios que tiene un determinado servicio. Por ejemplo, el servidor de páginas Web tiene asociado un usuario para poder especificar a qué ficheros tiene acceso; y por lo tanto que ficheros son visibles a través de Internet.

Todos los usuarios del sistema tienen un identificador de usuario (UID) y un identificador de grupo (GID). El administrador del sistema root tiene los identificadores de usuario y grupo 0:0 y los demás usuarios tienen un valor mayor que 0.

Existen varias formas de administrar el sistema, que van variando dependiendo de su facilidad o control sobre el sistema. Básicamente, puede administrar el sistema a través de tres formas diferentes:

- **Interfaces gráficas.** Existen diferentes interfaces gráficas que permiten administrar el sistema de una forma fácil y sencilla. Puede utilizar la interfaz de administración de x-Windows o utilizar la web de administración (webmin). Este método es el más sencillo, pero es el que menos control proporciona sobre el sistema.
- **Terminal del sistema.** Una de las ventajas de los sistemas GNU/Linux es que puedes administrarlo totalmente a través del intérprete de comandos o terminal del sistema. Una de las grandes ventajas de utilizar el terminal del sistema es que permite una gran flexibilidad a la hora de interactuar con el sistema, pudiendo crear pequeños programas (scripts) para simplificar la administración del sistema.
- **Ficheros de configuración.** Por último, la modificación directa de los ficheros de configuración es el método que permite tener un mayor control del sistema. Como desventaja hay que destacar que para administrar el sistema de esta forma hay que conocer muy bien el sistema.

No se puede decir que un método sea el mejor, ya que el uso de un método u otro depende siempre de la tarea que desees realizar y de tus conocimientos. Lo mejor, como siempre, es conocer los tres métodos y utilizar el mejor en cada momento.

Usuarios

adduser <usuario>	Permite dar de alta a un usuario. Cuando das de alta un usuario el sistema solicita sus datos como nombre completo, dirección, contraseña, etcétera.
addgroup	Permite dar de alta un usuario dentro de un grupo.
chage	Permite establecer los periodos de vigencia de las contraseñas.
id	Muestra el usuario que se está utilizando.
passwd	Permite cambiar la contraseña de un usuario. Si ejecutas passwd cambias la contraseña del usuario actual y si ejecutas passwd <i>nombre_usuario</i> cambia la contraseña del usuario indicado.
su	Permite cambiar de usuario.
sudo	Permite ejecutar un comando como <i>root</i> .
userdel <usuario>	Permite borrar un usuario.
usermod	Permite modificar las propiedades de un usuario.

Grupos

groups	Muestra los grupos a los que pertenece el usuario.
groupadd	Permite dar de alta a un grupo.
groupdel	Permite borrar un grupo de usuarios.
groupmod	Permite modificar las propiedades de un grupo.

Manipulación del fichero */etc/shadow*

pwconv	Crea y actualiza el fichero <i>/etc/shadow</i> .
pwunconv	Desactiva el fichero <i>/etc/shadow</i> .

6.Sistema de ficheros

Linux, al igual que UNIX, organiza la información del sistema en una estructura de árbol jerárquico de directorios compuesta de ficheros. Esta estructura se forma mediante un sistema de ficheros raíz (file system root) y un conjunto de sistemas de ficheros montables.

Existen diferentes formas que permiten administrar el sistema de ficheros y cada una de ellas proporciona diferentes resultados dependiendo de si desea administrar el sistema utilizando particiones, volúmenes o sistemas RAID.

Para identificar los discos duros o particiones se utiliza la siguiente sintaxis */dev/sda1* donde:

- **s** indica el tipo de disco duro: s – discos duros SATA o SCSI; y h para discos IDE.
- **a** identifica el primer disco duro, b el segundo, etcétera

- **1** indica el número de partición dentro del disco duro.

Así por ejemplo `/dev/sdb3` identifica la tercera partición del segundo disco duro y `/dev/sdb` identifica el segundo disco duro.

6.1. Particionamiento

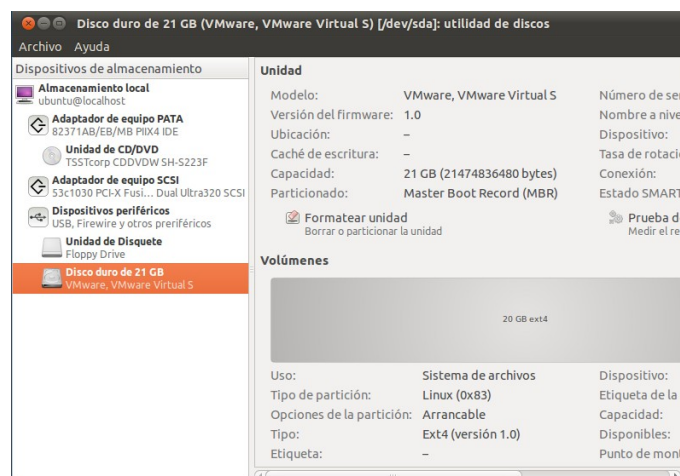
El particionamiento es uno de los procesos más importantes que hay que tener en cuenta, ya que define cómo se van a utilizar los diferentes discos duros del equipo. En el proceso de particionamiento hay que prestar un especial cuidado para no perder datos del sistema.

La administración de las particiones de los sistemas de ficheros se puede realizar con herramientas gráficas como la Utilidad de discos, el Administrador de volúmenes lógicos o, manualmente, con el comando `fdisk`.

6.1.1. Herramientas gráficas

Ubuntu Desktop por defecto instala la herramienta Utilidad de discos para administrar el sistema de ficheros. Utilizando la herramienta *Utilidad de discos* podemos crear, modificar o eliminar las particiones de los discos duros del sistema.

Para ejecutar la herramienta debes ir al menú *Sistema > Administración* y seleccionar la herramienta *Utilidad de discos*.

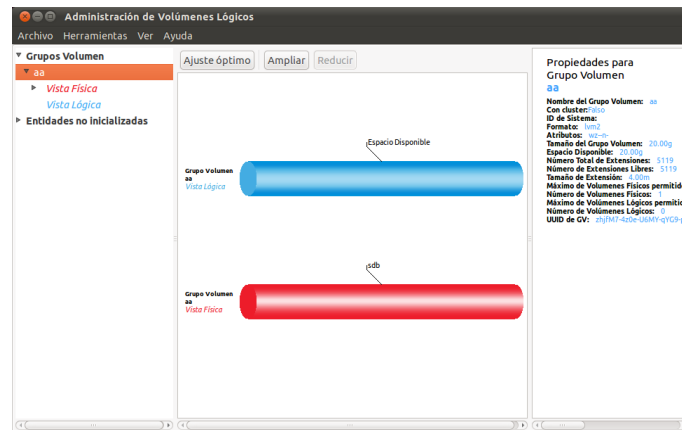


Por otra parte, es posible utilizar el *Administrador de volúmenes lógicos*. A diferencia de la herramienta *Utilidad de discos*, con el *Administrador de volúmenes lógicos* es posible crear volúmenes o unidades RAID. Recuerda que un volumen permite agrupar uno o más discos duros para tener un sistema de ficheros de mayor tamaño. Además, puede crear volúmenes en los que se mejore la seguridad de los datos. Por ejemplo, en un volumen reflejado (o RAID 1) los datos se guardan de forma simultánea en dos discos duros.

Para utilizar el administrador de volúmenes lógicos antes debes instalarlo, ejecutando:

```
# apt-get install system-config-lvm
```

y ejecutar la herramienta *Administración de volúmenes lógicos* que se encuentra en el submenú *Herramientas del sistema* dentro del menú *Aplicaciones*.



6.2. Monitorización

Existen muchas herramientas que permiten monitorizar el sistema de ficheros entre las que destacan:

- **df.** Muestra un resumen sobre el espacio libre que queda en los discos duros del sistema.

```
root@ubuntu-virtual-machine:~# df
S.archivos Bloques de 1K Usado Dispon Uso% Montado en
/dev/sda1 19222656 2711572 15534548 15% /
none 248268 212 248056 1% /dev
none 254244 252 253992 1% /dev/shm
none 254244 104 254140 1% /var/run
none 254244 0 254244 0% /var/lock
/dev/sda1 19222656 2711572 15534548 15% /prueba
root@ubuntu-virtual-machine:~#
```

- **du.** Muestra la cantidad de espacio que están utilizando los directorios o archivos específicos. Por ejemplo, si quieres ver el espacio que ocupa el directorio /datos en Megabytes ejecuta:

```
$ du -ms /datos
```

- **fsck.** Permite comprobar el estado y reparar un sistema de ficheros.

Para tener más información sobre el sistema de ficheros de los sistemas GNU/Linux, puedes consultar este enlace: http://www.guia-ubuntu.com/index.php?title=Sistema_de_ficheros

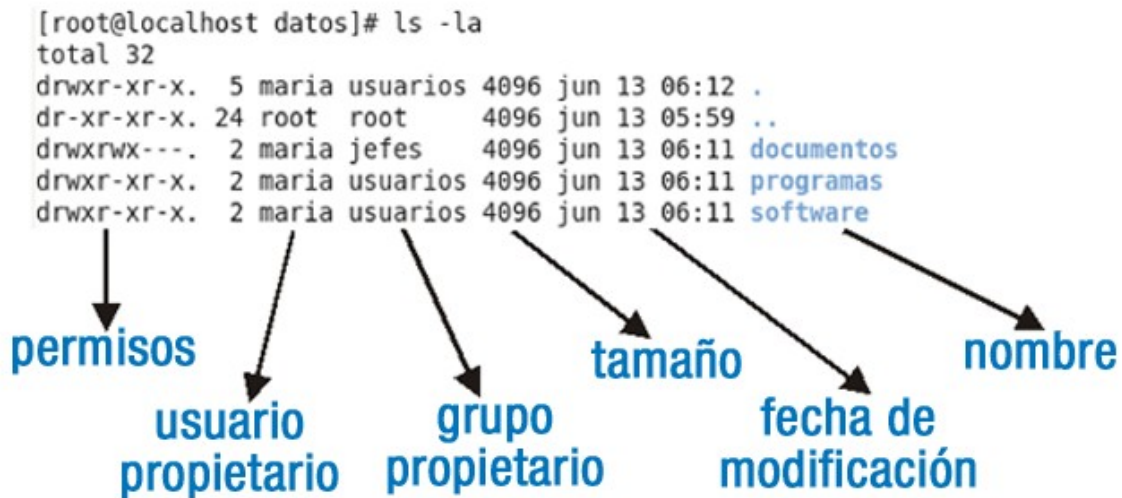
7. Permisos

Es muy importante establecer correctamente los permisos en el sistema de ficheros porque así evitaremos usos indebidos o pérdidas de datos en el sistema.

Si ejecutamos en un directorio el comando **ls -la** puedes ver los permisos del sistema de ficheros. Para cada fichero o directorio se muestran los siguientes datos:

- **Permisos.** Indica los permisos que tiene el fichero o directorio.

- **Usuario propietario.**
- **Grupo propietario.**
- **Tamaño del fichero o directorio.**
- **Fecha de creación o de la última modificación.**
- **Nombre.**



Por ejemplo, los permisos para el directorio `documentos` son `drwxrwx---`. El carácter *d* indica que es un directorio. Luego se muestran tres grupos de caracteres (*rw*x) (*rw*x) (*---*) que permiten indicar los permisos del usuario propietario, del grupo propietario y de los demás usuarios.

El formato para establecer los permisos es (*rw*x) donde *r* indica lectura, *w* escritura y *x* indica ejecución. Si existe el permiso entonces se muestra su correspondiente letra y en el caso de que no exista ese permiso entonces aparece el carácter (-).

Por ejemplo, el directorio `documentos` tiene todos los permisos (*rw*x) para el usuario propietario, que es *maria*, el grupo propietario *jefes* también tiene todos los permisos (*rw*x), y el resto de los usuarios no tiene ningún permiso (*---*).

El directorio `programas` tiene todos los permisos para el usuario propietario *maria* (*rw*x) y tanto para el grupo propietario *usuarios* como el resto de los *usuarios* tiene permisos de lectura y ejecución (*r-x*).

En un fichero el permiso de ejecución permite ejecutar un programa y en el caso de los directorios el permiso permite indicar que es posible entrar en ese directorio.

7.1. Establecer los permisos

Para definir los permisos de un fichero o directorio se emplea el comando `chmod`. Su sintaxis es:

```
# chmod <modo> fichero
```

donde *<modo>* indica los permisos que le quiere asignar al fichero. Por ejemplo, si quiere establecer los permisos *rw-* para el propietario y *r--* para el resto, el comando que se debe utilizar es:

chmod 644 fichero

```
[root@localhost datos]# ls -la
total 32
drwxr-xr-x. 5 maria usuarios 4096 jun 13 06:12 .
dr-xr-xr-x. 24 root root 4096 jun 13 05:59 ..
drwxrwx---. 2 maria jefes 4096 jun 13 06:11 documentos
drwxr-xr-x. 2 maria usuarios 4096 jun 13 06:11 programas
drwxr-xr-x. 2 maria usuarios 4096 jun 13 06:11 software
```

Diagram illustrating the components of the `ls -la` output:

- `permisos` (permissions)
- `usuario propietario` (owner)
- `grupo propietario` (group)
- `tamaño` (size)
- `fecha de modificación` (modification date)
- `nombre` (name)

Con **chmod** se puede establecer los permisos con tres valores numéricos (por ejemplo, 664): el primer valor corresponde al usuario propietario, el segundo al grupo propietario y el tercer valor corresponde a todos los demás usuarios del sistema.

Cada permiso tiene una equivalencia numérica donde *r* vale 4, *w* vale 2 y *x* vale 1. De esta forma si tiene el valor 7 corresponde a (*rwX*), el valor 6 corresponde a (*rw-*), etcétera.

El propietario de un fichero es aquel usuario que creó dicho fichero. GNU/Linux permite cambiar al propietario de cualquier fichero o directorio. Opcionalmente se puede cambiar también al grupo al que pertenece dicho fichero o directorio. Para ello se utiliza la orden `chown` que tiene la siguiente sintaxis:

```
chown <NombreUsuario> [:<NombreGrupo>] <fichero>...
```

donde *<NombreUsuario>* identifica el nuevo propietario de fichero o directorio. *<NombreGrupo>* el nuevo grupo y *<fichero>* identifica el fichero o directorio sobre el que se va a actuar.

Por otro lado, para cambiar el grupo al que pertenece un directorio se utiliza `chgrp`. Su sintaxis es:

```
# chgrp <NombreGrupo> <fichero>...
```

donde *<NombreGrupo>* identifica el nuevo nombre de grupo que se le va a asignar al fichero o directorio *<fichero>*. Se puede actuar sobre varios ficheros a la vez.

En los comandos **chmod**, **chown** y **chgrp** la opción **-R** significa que se establecen los permisos al directorio y a todos los datos que contiene. Por ejemplo, el comando

```
# chmod 777 /datos -R
```

establece todos los permisos a la carpeta `datos` y a todo su contenido.

8. Procesos

En los sistemas GNU/Linux se ejecutan una gran cantidad de servicios que permiten realizar una determinada actividad en el sistema. Cada servicio o demonio consiste en uno o más procesos que se ejecutan en el equipo. Además de los procesos vinculados a servicios, en el sistema se encuentran los procesos que ejecuta un usuario. Por ejemplo, un editor de textos, un navegador Web, etcétera.

Para ver los procesos que se ejecutan en el equipo hay que ejecutar el comando **ps**. Tal y como se muestra en la siguiente figura, para cada proceso se muestra su identificador (PID), terminal donde se ejecuta (TTY), tiempo de uso de CPU (TIME) y el comando que ejecuta (CMD).

```

ubuntu@ubuntu-virtual-machine: ~
Archivo Editar Ver Buscar Terminal Ayuda
ubuntu@ubuntu-virtual-machine:~$ ps
  PID TTY          TIME CMD
 1570 pts/0    00:00:00 bash
 2801 pts/0    00:00:00 ps
ubuntu@ubuntu-virtual-machine:~$

```

Si quieres ver todos los procesos que se ejecutan en el sistema utiliza la opción **-A**:

```
# ps -A
```

Si deseas eliminar un proceso que se está ejecutando en el sistema puede utilizar el comando **kill** de la siguiente forma:

```
# kill -9 <ID del proceso>
```

Otra aplicación que permite ver los procesos que se ejecutan en el sistema es **top**. **top** es una aplicación que, en tiempo real, informa sobre la actividad del sistema. Proporciona información sobre la carga del sistema operativo, grado de utilización de la CPU, memoria y swap, y los procesos que se encuentran en ejecución.

```

ubuntu@ubuntu-virtual-machine: ~
Archivo Editar Ver Buscar Terminal Ayuda
top - 12:20:49 up 20 min,  2 users,  load average: 0.15, 0.37, 0.42
Tasks: 128 total,  2 running, 126 sleeping,  0 stopped,  0 zombie
Cpu(s): 28.6%us, 21.4%sy,  0.0%ni, 50.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   508488k total,  428796k used,  79692k free,  25188k buffers
Swap:  975868k total,    8k used,  975860k free,  251752k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
  934 root        20   0 49476  18m 7432  R 29.0   3.7   0:37.03 Xorg
 2803 ubuntu      20   0 2624  1112  840  R  7.3   0.2   0:00.13 top
    1 root        20   0 2892  1676 1220  S  0.0   0.3   0:02.10 init
    2 root        20   0   0     0   0  S  0.0   0.0   0:00.00 kthreadd
    3 root        20   0   0     0   0  S  0.0   0.0   0:00.13 ksoftirqd/0
    4 root        RT   0   0     0   0  S  0.0   0.0   0:00.00 migration/0
    5 root        RT   0   0     0   0  S  0.0   0.0   0:00.00 watchdog/0
    6 root        20   0   0     0   0  S  0.0   0.0   0:00.18 events/0
    7 root        20   0   0     0   0  S  0.0   0.0   0:00.00 cpuset
    8 root        20   0   0     0   0  S  0.0   0.0   0:00.00 khelper
    9 root        20   0   0     0   0  S  0.0   0.0   0:00.00 netns
   10 root        20   0   0     0   0  S  0.0   0.0   0:00.00 async/mgr
   11 root        20   0   0     0   0  S  0.0   0.0   0:00.00 pm
   12 root        20   0   0     0   0  S  0.0   0.0   0:00.00 sync supers
   13 root        20   0   0     0   0  S  0.0   0.0   0:00.01 bdi-default
   14 root        20   0   0     0   0  S  0.0   0.0   0:00.00 kintegrityd/0
   15 root        20   0   0     0   0  S  0.0   0.0   0:00.48 kblockd/0
   16 root        20   0   0     0   0  S  0.0   0.0   0:00.00 kacpid

```

8.1. Programación de tareas

La programación de tareas permite programar la ejecución de un determinado programa en un momento determinado. Por ejemplo, se puede programar una copia de seguridad, enviar un fichero, comprobar la seguridad del sistema, enviar un informe, etcétera.

Antes de programar las tareas hay que comprobar que el servicio **crond** se encuentra en ejecución mediante el comando:

```
# service crond status
```


Para modificar el fichero de configuración de crond, ejecuta el comando:

```
# crontab -e
```

y aparece un fichero con el siguiente formato:

```
PATH=/bin
0 0 * * * /root/comprobar_seguridad.sh
0 0 1 * * /root/copia_seguridad.sh
```

La sintaxis de las tareas programadas es:

```
# .----- minuto (0 - 59)
# | .----- hora (0 - 23)
# | | .----- día del mes (1 - 31)
# | | | .----- mes (1 - 12) o jan,feb,mar,apr ... (los meses en inglés)
# | | | | .---- día de la semana (0 - 6) (Domingo = 0 o 7) o sun,mon,tue,wed,thu,fri,sat
# | | | | |
# | | | | |
* * * * * Comando a ejecutar
```

Así, en el ejemplo anterior se ejecuta el script *comprobar_seguridad.sh* todos los días a las 0:00h y se ejecuta *copia_seguridad.sh* el primer día de cada mes.

Otra forma de poder programar tareas es guardar el script que quiere ejecutar en las siguientes carpetas de configuración de cron:

```
/etc/cron.hourly      # Ejecuta el script cada hora
/etc/cron.daily        # Ejecuta el script diariamente
/etc/cron.weekly       # Ejecuta el script semanalmente
/etc/cron.monthly      # Ejecuta el script mensualmente
```

8.2. El comando tar

La utilidad **tar** (Tape ARchiver) es una herramienta de fácil manejo disponible en todas las versiones de UNIX/Linux que permite copiar ficheros individuales o directorios completos en un único fichero. Oficialmente fue diseñada para crear ficheros de cinta (esto es, para transferir ficheros de un disco a una cinta magnética y viceversa), aunque en la actualidad casi todas sus versiones pueden utilizarse para copiar a cualquier dispositivo o fichero, denominado “*contenedor*”.

En la siguiente tabla se muestran las opciones de **tar** más habituales. Algunas de ellas no están disponibles en todas las versiones de **tar**, por lo que es recomendable consultar la página del manual de esta orden antes de utilizarla.

Opción	Acción
c	Crea un contenedor.
x	Extrae ficheros de un contenedor.

t	Testea los ficheros almacenados en un contenedor.
r	Añade ficheros al final de un contenedor.
v	Modo verbose.
f	Especifica el nombre del contenedor.
z	Comprime o descomprime el fichero.

En primer lugar debe saber cómo crear contenedores con los ficheros deseados. Por ejemplo, para copiar el directorio `/home/` en el fichero `/root/copia.tgz` hay que ejecutar el siguiente comando:

```
# tar cvf /root/copia.tgz /home/
```

La opción “v” no es necesaria, pero es útil para ver el proceso de empaquetamiento del fichero. En muchas situaciones también resulta útil comprimir la información guardada (`tar` no comprime, sólo empaqueta) por lo que hay que utilizar las opciones “cvfz”.

En lugar de indicar un único directorio con todos sus ficheros y subdirectorios es posible especificar múltiples ficheros (o directorios). Por ejemplo, la siguiente orden crea el fichero `/tmp/backup.tar`, que contiene `/etc/passwd` y `/etc/hosts*`.

```
# tar cvf /tmp/backup.tar /etc/passwd /etc/hosts*
```

Para recuperar los ficheros guardados en un fichero tar se utilizan las opciones “xvf” (o “xvzf” si se ha utilizado compresión con gzip). Puedes indicar el fichero o ficheros a extraer; si no lo haces se extraerán todos los ficheros. A continuación puedes ver un ejemplo:

```
# tar xvf /tmp/backup.tar /etc/passwd
```

En el ejemplo anterior, la restauración se ha realizado desde el directorio de trabajo, creando en él un subdirectorio `/etc` con los ficheros correspondientes en su interior.

Un fichero con extensión “.tar” se llama empaquetado ya que el fichero ocupa lo mismo que su contenido. Mientras que un fichero con extensión “.tar.gz” o “.tgz” está comprimido y ocupa menos espacio que su contenido.