

## ¿Qué son las Distribuciones?

Una distribución GNU/Linux es una combinación específica multitud de elementos diferentes, pero manteniendo una base común, que es el sistema GNU con el kernel Linux.

Estas, aparte de incorporar la base inicial consistente en el sistema operativo GNU con el kernel Linux, también es pueden nutrir de muchos más elementos, como el entorno de escritorio o las aplicaciones de usuario que los desarrolladores y empaquetadores de cada distribución seleccionen (que pueden, o no, formar parte del proyecto GNU).

Esto da lugar a una enorme variedad de propuestas.

GNU/Linux Distributions Timeline, un completo diagrama temporal donde puedes ver representadas la gran mayoría de distribuciones, junto con la evolución y bifurcaciones que han ido teniendo a lo largo de su historia.

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

## Principales Distribuciones

Hay muchas distribuciones. Las hay para todos los gustos y requerimientos. Algunas están mantenidas enteramente por una comunidad, mientras que otras tienen el respaldo de una empresa detrás. Unas están más orientadas a servidores o entornos críticos, y otras son las delicias de los usuarios más noveles. Y así podría ir siguiendo, pero prefiero pasar a introducirte las que considero más importantes y relevantes a día de hoy.

### Debian



Debian es la madre de una gran parte de las distribuciones GNU/Linux que puedes encontrar hoy día, como es el caso de Ubuntu. Es un sistema de software libre, nacido en 1991 de la mano de Ian Murdoch y mantenido enteramente por la comunidad. Destaca por ser una de las distros más estables y robustas, lo que la hace ideal para utilizarse en servidores o en entornos críticos. Su sólida base es el motivo por la cual ha dado origen a la familia de distribuciones más numerosa a día de hoy.

### Ubuntu



Ubuntu es una de las distribuciones más conocidas y utilizadas en el panorama linuxero. Nació en 2004 con el objetivo principal de acercar a Linux al gran público, y una de sus premisas ha sido siempre la de ofrecer la mejor experiencia de usuario posible. Está basada en Debian, y mantenida por Canonical Ltd., junto con una gran comunidad de desarrolladores y usuarios que también contribuyen al proyecto.

### Linux Mint



Linux Mint es también una de las distribuciones más utilizadas. El proyecto toma la base de Ubuntu, fue fundado en 2006 por Clément Lefèbvre, y está mantenido por la comunitario. Nace con el foco puesto en la experiencia de usuario y la facilidad de uso, al igual que Ubuntu, pero ofreciendo una experiencia de usuario propia y diferenciada.

### Open SUSE



openSUSE es otra de las grandes. Es un proyecto comunitaria, pero auspiciado por SUSE Linux GmbH (compañía que está detrás de la SUSE Linux Enterprise, una distribución de pago y enfocada a uso empresarial). Es una distribución estable y robusta, pero para un usuario principiante, no la veo tan adecuada como otras distros como pueden ser Ubuntu o Mint.

## Fedora



Fedora es también una de las más conocidas. Nació en el año 2003, y es una distro de propósito general que destaca especialmente por ser de las más punteras, y de las primeras en adoptar las últimas novedades del panorama Linuxero en cada lanzamiento. Es un proyecto comunitario, pero que desde 2014 está amparado bajo el paraguas de la poderosa Red Hat (compañía detrás de Red Hat Enterprise Linux o RHEL).

## Centos



CentOS es una distribución de la familia de Red Hat / Fedora que nació en 2004 como un clon a nivel binario de Red Hat Enterprise Linux o RHEL, la distribución empresarial de Red Hat, Inc. De hecho, la principal diferencia con respecto a esta última es el hecho de que CentOS es un proyecto comunitario (aunque ahora bajo el paraguas de Red Hat, Inc.) y se puede descargar de forma totalmente gratuita.

## Arch Linux



Arch Linux nació en 2002, y es un proyecto mantenido enteramente por su comunidad de usuarios. Arch se diferencia por ser un sistema enormemente flexible y personalizable por parte del usuario. Por eso mismo, requiere del usuario un cierto trabajo manual, que en otras a menudo viene automatizado y predefinido de fábrica.

## Shell

Una shell es una herramienta que acepta órdenes o instrucciones por parte del usuario y ejecuta operaciones. En Linux podemos interactuar con multitud de shells diferentes, tanto a nivel de interfaz de línea de comandos, como a nivel de interfaz gráfica.

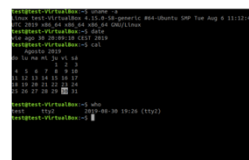
El concepto de interfaz de usuario, que no es otra cosa que el medio o método de interacción entre el usuario y la máquina. Se trata de un concepto general, que aplica a todos los sistemas informáticos que requieran de interacción humana.

Por un lado tenemos a la interfaz de línea de comandos o CLI, y por otro lado tenemos a la interfaz gráfica o GUI.

### CLI

*Command Line Interface*

Interacción mediante texto.  
Se basa en el uso de un lenguaje codificado.



### GUI

*Graphical User Interface*

Interacción a través de elementos gráficos.  
Posibilita una interacción amigable e intuitiva.



## El entorno gráfico o Shell GUI

Mientras que en Windows y macOS el entorno de escritorio es algo que viene implícito en el sistema (digamos que no tiene una entidad propia por él mismo), aquí la historia es diferente. De entrada, GNU/Linux se puede usar en modo consola, sin necesidad de ningún entorno gráfico. Pero es que, además, no existe un único entorno gráfico que sea el estándar, sino que la variedad es la norma (con sus pros y sus contras).

El entorno gráfico en su conjunto es lo que podemos decir que es lo que determina la experiencia visual de usuario. La experiencia gráfica de la que disponemos en cualquier distribución Linux es el resultado de un complejo entramado de protocolos, servicios y herramientas interdependientes, empezando por el servidor gráfico, en la capa más baja, pasando por los gestores de ventanas, y acabando en los entornos de escritorio.

## El Servidor Gráfico

El servidor gráfico es la primera capa de software, a nivel gráfico, que interactúa con el hardware del dispositivo. Generalmente proporciona las herramientas necesarias para permitir disponer de una interfaz gráfica en Linux, pero luego debe acompañarse de otros elementos clave, como el gestor de ventanas, o el entorno de escritorio propiamente dicho.

GNU/Linux siempre ha ido tradicionalmente muy ligado a X Windows System como sistema gráfico, pero en los últimos años han salido otras propuestas, entre las que se puede destacar el caso de Wayland, que aunque sigue en pleno desarrollo, todo apunta a que podría convertirse el sucesor de X.

### El Entorno de Escritorio

A diferencia de lo que ocurre en el caso de Windows o macOS, en GNU/Linux el entorno de gráfico no es algo que vaya grabado a fuego al sistema. Muchas distribuciones se ofrecen en varias ediciones diferenciadas, en función del escritorio que utilizan. Aparte de esto, nada te impide instalar tu mismo el entorno que desees (basta con instalar los paquetes necesarios que lo componen).

Generalmente, a parte de los componentes más core del entorno en sí, cada uno de ellos también suele incorporar un set de aplicaciones básicas predefinidas, como su propio navegador web, reproductor multimedia, o procesador de textos. Lo mejor de todo, y prueba de la gran flexibilidad que te encontrarás en GNU/Linux, es que puedes utilizar una distribución con un entorno de escritorio concreto, pero a la vez, utilizar aplicaciones pertenecientes a otro.

Podemos ver un poco más en detalle cuales son las principales opciones en ese contexto. Los más usados y conocidos, que son GNOME y KDE Plasma, pero han ido apareciendo otros actores, como Cinnamon, Xfce, Plasma, LXQt o LXDE.

### GNOME

GNOME es un entorno de escritorio de software libre y que se usa mayormente en combinación con el sistema GNU/Linux. El proyecto nació en 1999, «Thessaloniki», liberada el pasado septiembre de 2019. Probablemente es, junto con KDE Plasma, el entorno más utilizado, y el que tiene un mayor equipo de desarrollo detrás.

El proyecto GNOME está también íntimamente relacionado con varios de los avances más prominentes en el mundo GNU/Linux, como pueden ser el gestor de inicio systemd, o el gestor de ventanas Wayland, que viene a sustituir el viejo X11.

GNOME lo puedes encontrar como entorno de escritorio predeterminado en una gran parte de las distribuciones GNU/Linux, como es el caso de Debian, Ubuntu o Fedora.

### KDE Plasma

Es uno de los escritorios visualmente más atractivos y con más opciones de personalización. A continuación te dejo el vídeo de presentación oficial de la versión más reciente, la 5.16, presentada el pasado Junio de 2019. Actualmente, KDE agrupa su extenso catálogo de software en tres categorías separadas, bajo el nombre de KDE Frameworks, KDE Plasma y KDE Applications. KDE Plasma 5, la quinta y actual generación del escritorio, es el sucesor de KDE Plasma Workspaces y se lanzó inicialmente el 15 de julio de 2014.

Plasma 5 incluye un nuevo tema por defecto, llamado Breeze, y una mayor convergencia entre diferentes dispositivos. La interfaz gráfica se migró totalmente a QML, que usa OpenGL para la aceleración de hardware, proporcionando un mejor rendimiento y un menor consumo energético.

Plasma no es un entorno que deba ir necesariamente asociado a GNU/Linux. Puedes utilizarlo también en Windows o macOS.

### Xfce

Xfce es un entorno de escritorio ligero, enfocado a ofrecer un buen rendimiento sobretodo en ordenadores con pocos recursos, siendo en este caso, una opción más recomendable frente a GNOME y KDE Plasma, que son unos entornos muy completos, pero con un consumo de recursos mayor.

## La Terminal de Linux

La interfaz de línea de comandos, o CLI, por sus siglas en inglés (command-line interface) es un método de comunicación entre usuario y maquina que acepta instrucciones del usuario a través de líneas de texto (siguiendo unas determinadas reglas de sintaxis que puedan ser interpretadas por el sistema operativo).

Es el método que se utilizaba antes de la aparición de las interfaces gráficas o GUI, que sirvió para acercar la informática al gran público, al permitir una interacción del usuario con la maquina mucho más amigable o intuitiva (a costa, obviamente, de una mayor complejidad y consumo de recursos).

Hoy día todos los sistemas operativos se presentan con su propia interfaz gráfica plenamente integrada con el resto del sistema, de modo que hablar de interfaz gráfica o interfaz de línea de comandos, para el común de los usuarios, no tiene cabida.

Al igual que puedes optar por utilizar diferentes entornos de escritorio, también tienes la posibilidad de no disponer de ninguno, y operar directamente a través de la interfaz de línea de comandos (de hecho, algunas distribuciones, como la versión de Ubuntu para servidores, vienen de serie así).

En Linux tenemos a multitud de shells o interpretes diferentes. El más conocido de todos probablemente es Bash, debido a que es el que suele venir por defecto en la gran mayoría de distribuciones GNU/Linux, pero también destacan otros como Bourne shell (sh), Korn shell (ksh) o C shell (csh).

### Bash

Bash es una herramienta Open Source perteneciente al proyecto GNU, y que fue escrita por Brian Fox. Su nombre es el acrónimo de «Bourne-Again Shell», en referencia a Bourne shell (sh), uno de los primeros interpretes de comandos de Unix. De hecho, Bash hereda muchas propiedades de otras shells como sh, csh o zsh. Actualmente es la interfaz de línea de comando predeterminada en la mayoría de distribuciones GNU/Linux así como en macOS.

Como ocurre con otras shells de Unix, además de interprete de comandos, Bash es también un lenguaje de scripting. Esto lo hace extremadamente potente para multitud de tareas relacionadas con la administración de sistemas, automatización de tareas, etc.

### Las Terminales TTY

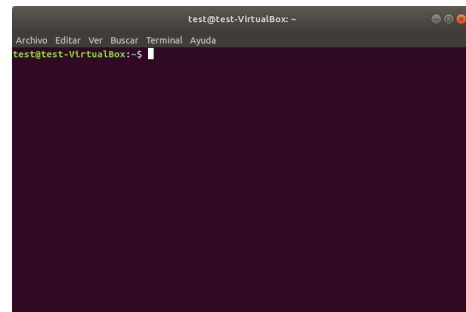
Las TTY son las diferentes ventanas de terminal que permiten interactuar con el sistema a través del intérprete de comandos, fuera de la entorno gráfica (salvo una de ellas, que es la que suele estar reservada para la sesión gráfica).

En la gran mayoría de distribuciones basadas en Debian, como Ubuntu o Linux Mint, existen un total de 7 pantallas TTY, las cuales pueden usarse simultáneamente. En el caso concreto de Debian, la TTY1 hasta la TTY6 son sesiones de interfaz de línea de comandos, mientras que la TTY7 es, de hecho, la interfaz gráfica de usuario. Aún así, el número puede variar en función de la distribución.

### Los Emuladores de Terminal

Aparte de la interfaz de línea de comandos que nos ofrece la shell a través de las diferentes terminales TTY, en Linux los propios entornos de escritorio de Linux también permiten utilizar la terminal a través de los emuladores de terminal, o PTS.

Un emulador de terminal es una aplicación que permite virtualizar un interprete de comandos pero dentro de la propia interfaz gráfica. La mayoría de entornos de escritorio en GNU/Linux incorporan su propio emulador de terminal.



Tienen la ventaja de posibilitar la interacción a través de la línea de comandos, pero sin necesidad de salir del entorno gráfico.

Existen muchos emuladores de terminal, aparte de los que suelen venir pre-instalados en los principales entornos de escritorio. Algunos de los más conocidos, a parte de los propios de GNOME y KDE, son Guake, Yakuake o Xterm.

En cualquier distribución GNU/Linux tienes básicamente dos maneras de usar la línea de comandos en GNU/Linux: operando directamente con la shell a través de las diferentes terminales TTY, o bien desde el entorno gráfico, utilizando un emulador de terminal (que puede ser el que venga con tu escritorio, o cualquier otro de tu preferencia).

Para acceder a las diferentes sesiones TTY debes utilizar las combinaciones de teclado 'CTRL+ALT+F1', para la TTY1, hasta 'CTRL+ALT+F7'. Para iniciar sesión deberás introducir tu nombre de usuario y contraseña, y a partir de ahí ya puedes empezar a interactuar con el interprete de comandos.

Otra manera más práctica de utilizar la línea de comandos es mediante el emulador de terminal desde la propia sesión gráfica. En este caso, basta con acceder a tu aplicación de terminal favorita y listos. No es necesario que inicies sesión aquí, ya que ya reconoce el usuario con el que has iniciado la propia sesión gráfica.

## Prompt de Usuario y Superusuario

Dependiendo del usuario que esté logueado en la sesión de la terminal, el prompt será ligeramente distinto. De este modo, si te encuentras logueado como usuario estándar, el prompt será similar a éste:

```
user@computer:~$
```

Por el contrario, si haces un log-in como superusuario (a partir del comando `sudo su`), el prompt adoptará este formato:

```
root@computer:/home/user#
```

## Comandos

Un comando Linux es un programa o utilidad que se ejecuta en la línea de comandos. Una línea de comandos es una interfaz que acepta líneas de texto y las procesa en forma de instrucciones para tu ordenador.

Cualquier interfaz gráfica de usuario (GUI) no es más que una abstracción de los programas de línea de comandos. Por ejemplo, cuando cierras una ventana haciendo clic en la «X», hay un comando que se ejecuta detrás de esa acción.

Los comandos de la terminal utilizan parámetros y argumentos, los parámetros son aquellos precedidos por el guión o signo de menos "-" y los que no lo tienen son los argumentos.

```
$ ls -l proyectos
```

En el ejemplo anterior el comando `ls` tiene como parámetros `-l` y como argumentos `proyectos`.

En una forma sencilla si hablamos del comando, los parámetros y argumentos.

Los comandos son instrucciones que se quieren realizar.

Los parámetros indican como estas instrucciones deben realizarse.

Los argumentos indican sobre que deben realizarse dichas instrucciones.

Un parámetro también llamado flag es una forma de pasar opciones al comando que se ejecuta. La mayoría de los comandos de Linux tienen una página de ayuda que podemos llamar con la flag -h. La mayoría de las veces, las flag son opcionales.

Un argumento es la entrada de información que damos a un comando para que pueda ejecutarse correctamente. En la mayoría de los casos, el argumento es una ruta de archivo, pero puede ser cualquier cosa que se escriba en el terminal.

Puedes invocar flags utilizando guiones (-) y guiones dobles (--), mientras que la ejecución de los argumentos depende del orden en que los pase a la función y no llevan ningún prefijo.

## Documentación de los comandos con Páginas de manual

Hay un comando “man” que es una herramienta con la que contamos en los sistemas Unix o del tipo Unix, también conocida como páginas de manual, que se utiliza para acceder a la documentación disponible de sus herramientas y así aprender sobre comandos, archivos, llamadas de sistema, etc, en un sistema operativo tal y como GNU/Linux.

Es la abreviatura de manual, proporciona información sobre el comando solicitado o permite a los usuarios buscar comandos relacionados con un determinada palabra clave.

Cada paquete, aplicación, servicio, herramienta y hasta los internals del sistema operativo y el lenguaje con el cual fue programado, tienen su respectiva página de manual.

Interfaz de las paginas de manual

El manual construye su interfaz en bloques bien definidos y ordenados de forma prácticamente estándar.

Nombre: Nombre del comando descrito por la página de manual.

Sinopsis: Un resumen del comando y su sintaxis.

Configuración: Detalles de configuración de un dispositivo.

La descripción: Una explicación de lo que hace el programa.

Opciones: Una descripción de las opciones de la línea de comando aceptadas por el comando.

Estado de salida: Posibles valores del estado de salida del pedido y lo que podría dar lugar a su uso.

Valor devuelto: Si la página de manual es para una rutina de biblioteca, describe el valor que la rutina de biblioteca puede devolver a la función que llamó a esa rutina.

Los errores: Una lista de valores que se pueden colocar en error. En caso de error.

Medio ambiente: Una lista de variables de entorno que afectan el comando o programa y cómo.

Archivos: Una lista de archivos utilizados por el comando o programa, como archivos de configuración.

Modalidad: Un resumen de los diferentes atributos del pedido.

Versiónes: Detalles del kernel de Linux o versiones de la biblioteca en las que ha aparecido una llamada al sistema o una función de biblioteca o ha cambiado significativamente con respecto a versiones anteriores.

Atenerse a: Una descripción de los estándares a los que puede ajustarse el comando, como POSIX.

Notas: Notas misceláneas.

Ejemplos: Uno o más ejemplos que demuestran el uso del comando.

Autores: Las personas que escribieron o mantuvieron el orden.

Las páginas de manual se almacenan en /usr/share/man.

Las páginas de manual en línea están organizadas en ocho secciones que se muestran a continuación:

Sección Descripción

- 1 Programas ejecutables o comandos de shell
- 2 System calls (funciones provistas por el kernel)
- 3 Library calls (funciones en librerías del sistema)
- 4 Archivos especiales (usualmente ubicadas en /dev )
- 5 Formatos de archivos (generalmente archivos de configuración)
- 6 Juegos
- 7 Información varia
- 8 Comandos de mantenimiento

## Instalación de software

En GNU/Linux todo el software (desde la propia base del sistema, hasta las diferentes aplicaciones de usuario que puedes instalar voluntariamente) se distribuye en forma de paquetes, los cuáles se gestionan a través de los denominados gestores de paquetes, y se obtienen de los repositorios de software oficiales de cada distribución.

### Los Paquetes

Un paquete es un fichero comprimido que contiene todos los archivos que se requieren para la instalación de un determinado software. Así, por entendernos, cualquier pieza de software consta de un paquete o un conjunto de paquetes.

Muchos paquetes son dependientes entre sí, lo que significa que se necesitan entre ellos para poder operar correctamente. Asimismo, las dependencias pueden ser compartidas entre diferentes paquetes. Es decir, es muy común que haya paquetes que sean utilizados por diferentes programas, como parte de sus dependencias.

Los formatos típicos de paquetes que se suelen utilizar son .deb, para todo el universo de distribuciones basadas en Debian, y .rpm, creado por Red Hat y utilizado en muchas otras distribuciones como Fedora, openSUSE o CentOS.

Como la gestión manual de tantos paquetes y dependencias sería muy compleja de cara al usuario, ahí entran en juego los gestores de paquetes, como APT, DNF (el sucesor de Yum en Fedora), Zypper o Pacman.

### Los Gestores de Paquetes

Los gestores de paquetes son una parte esencial en cualquier distribución GNU/Linux, y en última instancia, es el principal factor diferencial (junto con el formato de paquetes como tal) entre unas distribuciones y otras. Obviamente de distribuciones hay a miles, y los gestores de paquetes los podemos contar con los dedos de una mano, ya que es uno de los componentes más core del sistema, y de la cual dependen muchas otras herramientas.

Entre las principales funciones de un gestor de paquetes están las siguientes:

- La gestión automática de las dependencias (al instalar o desinstalar paquetes).
- La búsqueda de actualizaciones de todos los paquetes que tenemos en local, según la última versión disponible en los repositorios oficiales.
- La comprobación de las sumas de verificación y firmas digitales.

Luego, aparte del gestor de paquetes en sí, están las diferentes herramientas o aplicaciones (ya sean por consola o a nivel gráfico) que hacen de interfaz para permitir la interacción del usuario con el propio gestor de paquetes.

El gestor de paquetes suele ser un componente compartido entre muchísimas distribuciones que derivan de una misma base común.

Así pues, tenemos a APT, que es el estándar utilizado entre las distribuciones de la familia Debian (como Ubuntu o Linux Mint), a Yum y a DNF, que son los utilizados por los integrantes de la familia Red Hat (como Fedora, CentOS o RHEL).

Todos los gestores de paquetes ofrecen un conjunto de herramientas core para permitir su interacción a través de la línea de comandos.

A modo de ejemplo, y centrándonos en el caso de las distribuciones de la familia Debian, podemos mencionar al propio APT, que aparte de ser el propio gestor de paquetes como tal, ofrece una interfaz de órdenes para permitir la interacción del usuario a través de la línea de comandos.

A través de los diferentes comandos que ofrece Apt, se pueden hacer multitud de tareas, tales como actualizar el catálogo de paquetes, actualizar paquetes, instalar o eliminar paquetes (junto con sus dependencias), entre otras muchas tareas más concretas de mantenimiento.

Y lo dicho con Apt es perfectamente extrapolable a otros gestores de paquetes utilizados en otras distribuciones, como por ejemplo DNF en el caso de Fedora y sus derivados, Zypper en el caso de openSUSE o Pacman en el caso de Arch Linux.

El denominador común en todos estos casos es que la interacción se hace a partir de la línea de comandos. Esto ha promovido que, en busca de facilitar la vida al usuario, hayan ido naciendo otras utilidades para posibilitar una interacción desde la propia interfaz gráfica.

### Los Repositorios de Software

Los repositorios de software permiten ofrecer un catálogo de aplicaciones gestionado desde un lugar centralizado y supervisado por la comunidad de desarrolladores y usuarios.

Las ventajas de esto son varias, empezando por el hecho de evitarte la necesidad de tener que visitar webs externas para la descarga de software. Esto da bastantes garantías al usuario de que todo el software que ha descargado es legítimo, y que además estará soportado, bien por parte de la comunidad de desarrolladores de la distribución, o bien a través de la empresa que haya detrás (Canonical, Red Hat, SUSE, etc.).

Otra ventaja es la facilidad a la hora de administrar las actualizaciones. El gestor de paquetes generalmente te alertará cada vez que detecte que en los repositorios haya versiones nuevas de los paquetes que tengas instalados. De este modo, como sistema y aplicaciones derivan de las mismas fuentes de software, estarás actualizando todo tu sistema en su conjunto.

Cada distribución cuenta con sus propios repositorios de software, y normalmente el catálogo es suficientemente extenso como para cubrir la mayor parte de necesidades de un usuario normal.

### El Prompt

El indicador de shell (shell prompt) se muestra en una ventana de terminal cuando se inicia, o en una consola basada en texto después de iniciar sesión. El indicador de shell permite saber que la computadora está esperando que se escriba un comando.

El prompt en la shell de linux es la información o el símbolo que se encuentra antes del cursor, es decir donde empezamos a escribir los comandos que ingresamos a la shell. En cada distribución de Linux este viene por default de una forma diferente, y dentro de la misma distribución los usuarios suelen personalizar el prompt de muchas maneras.

Por ejemplo:

```
[luis@finanzas ~]$
```

```
gafunes@ubuntu-mac-gaf: ~$
```

En el ejemplo anterior podemos ver que el prompt muestra el nombre del usuario "luis" y el host - nombre del equipo en la red local - "finanzas", el signo de moneda "\$" indica que es un usuario quien va a ingresar los comandos, esto es así porque en ocasiones aparece el símbolo de número "#" que indica que quien ingresa el comando es el usuario root.

El elemento más importante en los prompt es el último carácter de cada uno: un \$ o un # en la mayoría de los casos. Este carácter indica que el usuario actual conectado es o bien un usuario cualquiera sin derechos especiales (\$), o bien el administrador que cuenta con todos los derechos necesarios para la configuración y el mantenimiento del sistema (#).

### Personalización del Prompt

BASH (Bourne-again shell) es el shell predeterminado para la mayoría de las distribuciones modernas de Gnu/Linux. En las siguientes líneas vamos a personalizar el indicador BASH y mejorar su aspecto agregando algunos colores, estilos, modificando elementos, etc. Todo ello sin tener que instalar herramientas, complementos o recurrir a servicios online.