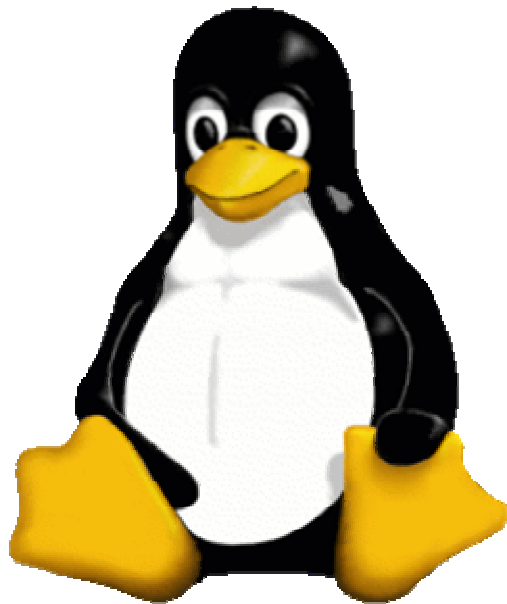


# CURSO DE LINUX



Organiza: CAP de Torrejón de Ardoz  
Ponente: Manuel Santos González  
Marzo 2004

**ÍNDICE DE CONTENIDOS**

<b>1. HISTORIA Y CONTEXTO DENTRO DEL MUNDO UNIX</b>	<b>1</b>
1.1. ¿QUÉ ES LINUX?	1
1.2. HISTORIA DE UNIX	2
1.3. EVOLUCIÓN DE LINUX	3
1.4. DISTRIBUCIONES	4
<b>2. APROXIMACIÓN A LINUX</b>	<b>7</b>
2.1. CARACTERÍSTICAS GENERALES	7
2.2. MODO GRÁFICO VS MODO COMANDO	8
<b>3. TRABAJO EN MODO COMANDO</b>	<b>10</b>
3.1. ACCESO A UN SISTEMA LINUX EN MODO COMANDO	10
3.2. LA SHELL	10
3.3. PRIMEROS COMANDOS	11
3.4. ESTRUCTURA DE DIRECTORIOS	12
3.5. COMANDOS PARA TRABAJAR CON LA ESTRUCTURA DE DIRECTORIOS	13
3.6. JERARQUÍA ESTÁNDAR DE DIRECTORIOS	14
3.7. PROPIEDADES DEL SISTEMA DE FICHEROS	16
3.8. COMODINES PARA LA GENERACIÓN DE LISTAS DE FICHEROS	17
3.9. COMANDOS PARA EL TRABAJO CON EL SISTEMA DE FICHEROS	17
3.10. ENLACES	20
3.11. OPERADOR PIPELINE O TUBERÍA	21
3.12. REDIRECCIÓN DE COMANDOS	21
3.13. COMANDOS GENERALES Y DE INFORMACIÓN DEL SISTEMA	22
3.14. GESTIÓN DE PROCESOS	24
3.15. OTROS COMANDOS	27
3.16. EL EDITOR EN MODO COMANDO UNIVERSAL: VI	28
3.17. LAS HERRAMIENTAS MTOOLS	33
<b>4. ADMINISTRACIÓN BÁSICA</b>	<b>34</b>
4.1. INTRODUCCIÓN	34
4.2. ARRANQUE DEL SISTEMA	34
4.3. GESTIÓN DE USUARIOS	35
4.4. FICHEROS DE CONFIGURACIÓN DE LA SHELL BASH DE USUARIO	37
4.5. FICHEROS DE DISPOSITIVOS	37
4.6. MONTAR Y DESMONTAR DISPOSITIVOS	38
4.7. COPIAS DE SEGURIDAD: TAR, GZIP Y GUNZIP	40
<b>5. MODO GRÁFICO</b>	<b>41</b>
5.1. INTRODUCCIÓN	41

5.1.1. PANTALLA GRÁFICA DE CONEXIÓN	41
5.1.2. ABRIR UN INTÉRPRETE DE COMANDOS DE LA SHELL	42
5.1.3. SALIR EN MODO GRÁFICO Y APAGAR EL ORDENADOR	42
<b>5.2. ELEMENTOS DEL ESCRITORIO GRÁFICO</b>	<b>43</b>
5.2.1. EL PANEL	43
5.2.2. MENÚ PRINCIPAL	44
5.2.3. APPLETS	44
<b>5.3. CONFIGURACIÓN BÁSICA DEL ENTORNO GRÁFICO</b>	<b>45</b>
5.3.1. AÑADIR ICONOS Y APPLETS AL PANEL	45
5.3.2. CONFIGURACIÓN DEL PANEL EN EL ESCRITORIO	45
5.3.3. USO DEL ICONO EMPEZAR AQUÍ	46
5.3.4. CAMBIAR EL FONDO DEL ESCRITORIO	46
<b>6. APLICACIONES EN ENTORNO GRÁFICO</b>	<b>47</b>
<hr/>	
<b>6.1. ADMINISTRADOR DE ARCHIVOS: NAUTILUS</b>	<b>47</b>
<b>6.2. TRABAJO CON DISQUETES</b>	<b>47</b>
6.2.1. MONTAJE Y DESMONTAJE DE UN DISQUETE	47
6.2.2. FORMATEO DE DISQUETES: GFLOPPY	49
<b>6.3. ACCESO A UN DISCO CD-ROM</b>	<b>50</b>
<b>6.4. TRABAJO CON ARCHIVOS DE TEXTO</b>	<b>51</b>
6.4.1. GEDIT	51
6.4.2. VISUALIZACIÓN DE PDFs: XPDF	52
<b>6.5. TRABAJAR CON DOCUMENTOS: OPENOFFICE.ORG</b>	<b>53</b>
6.5.1. OPENOFFICE.ORG WRITER	53
6.5.2. OPENOFFICE.ORG CALC	55
6.5.3. OPENOFFICE.ORG IMPRESS	56
6.5.4. OPENOFFICE.ORG DRAW	57
<b>6.6. NAVEGACIÓN WEB</b>	<b>58</b>
6.6.1. MOZILLA	58
6.6.2. GALEON	59
<b>6.7. APLICACIONES DE CORREO ELECTRÓNICO</b>	<b>60</b>
6.7.1. EVOLUTION	61
6.7.2. MOZILLA MAIL	62
<b>6.8. TRABAJAR CON CD-Rs Y CD-RWs</b>	<b>62</b>
6.8.1. USO DE LA APLICACIÓN CD CREATOR	62
6.8.2. USO DE X-CD-ROAST	64
<b>6.9. APLICACIONES DE AUDIO: CD PLAYER Y XMMS</b>	<b>67</b>
6.9.1. CD PLAYER	67
6.9.2. XMMS	67
<b>6.10. TRABAJAR CON IMÁGENES</b>	<b>68</b>
6.10.1. NAUTILUS PARA VER IMÁGENES	68
6.10.2. VISOR DE IMÁGENES GTHUMB	68
6.10.3. GIMP	70
<b>6.11. COMPRESIÓN DE ARCHIVOS: FILE ROLLER</b>	<b>71</b>
<b>7. CONFIGURACIÓN Y TRABAJO EN RED</b>	<b>73</b>
<hr/>	
<b>7.1. CONFIGURACIÓN EN ENTORNO GRÁFICO</b>	<b>73</b>

<b>7.2. CONFIGURACIÓN EN MODO COMANDO</b>	<b>74</b>
<b>7.3. OTROS COMANDOS</b>	<b>76</b>
<b>7.4. USO DEL SERVICIO NFS</b>	<b>77</b>
<b><u>8. ADMINISTRACIÓN AVANZADA</u></b>	<b><u>79</u></b>
<b>8.1. GESTIÓN DE USUARIOS EN ENTORNO GRÁFICO</b>	<b>79</b>
<b>8.2. COMANDOS DE GESTIÓN DE DISCOS</b>	<b>80</b>
<b>8.3. CUOTAS</b>	<b>81</b>
<b>8.4. EJEMPLO DE TAREA DE ADMINISTRACIÓN</b>	<b>83</b>
<b>8.5. ARCHIVOS DE REGISTRO DEL SISTEMA</b>	<b>85</b>
<b>8.6. PLANIFICACIÓN DE TRABAJOS: CRON</b>	<b>86</b>
<b><u>9. INSTALACIÓN DE APLICACIONES</u></b>	<b><u>88</u></b>
<b>9.1. RPM: RED HAT PACKET MANAGER</b>	<b>88</b>
<b>9.2. OTROS MÉTODOS DE INSTALACIÓN</b>	<b>90</b>
<b>9.3. HERRAMIENTA DE INSTALACIÓN GRÁFICA</b>	<b>90</b>
<b><u>10. BIBLIOGRAFÍA</u></b>	<b><u>92</u></b>

# 1. HISTORIA Y CONTEXTO DENTRO DEL MUNDO UNIX

## 1.1. ¿Qué es Linux?

La definición más extendida de Linux es que es un sistema operativo gratuito basado en Unix que se ejecuta en máquinas PC con tecnología Intel.

De esta definición se obtienen tres ideas fundamentales:

- Es un sistema operativo que se desarrolló inicialmente para funcionar en **máquinas PC con procesadores compatibles Intel**, es decir, su principal competidor en este aspecto es el sistema operativo Windows de Microsoft. Actualmente hay implementaciones de Linux para otras plataformas, como Alpha, Sparc, Atari, PowerPC, ...
- **Es gratuito**, es decir, no hay que pagar ningún tipo de licencia a ninguna empresa desarrolladora de software. Algo muy a tener en cuenta. Además, a esta filosofía se adhiere una gran cantidad de software que funciona bajo Linux.
- **Está basado en el sistema Unix**. Esta característica aporta a Linux una gran robusted y fiabilidad ya que, aunque el código fuente no es el mismo, Linux utiliza de Unix todas las características que han hecho de Unix un sistema operativo muy fiable.

A estas características se le pueden añadir una más, y es que, su distribución va acompañada del **código fuente**. Esto es debido a que Linux se distribuye bajo los términos de licencia GNU General Public License, lo cual significa básicamente, que cualquiera puede libremente, copiarlo, cambiarlo y distribuirlo pero sin posibilidad de aplicar restricciones en futuras distribuciones y teniendo que incluir el código fuente. (Más información en <http://www.gnu.org>)

A Linux se le ha considerado un clónico de Unix para sistemas compatibles Intel, y en cierta forma es así. Por ello puede ser interesante hacer una recapitulación de la historia de Unix antes de empezar con la de Linux. Conocer bien un sistema como Linux equivale a conocer también mucho de Unix.

Primero, hay que tener claro que Unix, hoy en día es una familia de sistemas operativos con una cosa en común, potencia y fiabilidad. Las máquinas basadas en sistemas tipo Unix se utilizan en entornos empresariales como servidores de todo tipo de aplicaciones. También se utiliza mucho en entornos académicos.

**Un ejemplo de su fiabilidad: apenas existen virus en Unix.**

## 1.2. Historia de UNIX

UNIX nació en 1969 en los Laboratorios Bell, de la empresa americana AT&T. Por aquel entonces, se había completado el desarrollo del sistema operativo MULTICS por parte de varias compañías, entre las que destacaban AT&T y General Electric. Como uno de los primeros sistemas de tiempo compartido, MULTICS incorporaba la mayoría de las ideas que aparecen en los sistemas operativos actuales. Desgraciadamente MULTICS resultó mucho más complejo y pesado de lo que era necesario, lo cual lo hizo inviable. A finales de los sesenta, AT&T abandonó el proyecto MULTICS.

Sin embargo, dos ingenieros de AT&T, Ken Thompson y Dennis Ritchie, quienes habían participado en el proyecto, decidieron construir el sistema basado en un diseño que ellos mismos elaboraron junto con Rudd Canaday. Adquirieron una máquina DEC PDP-7 y se pusieron a desarrollar. Como muchos proyectos interesantes, éste empezó con la creación de un juego: Thompson y Ritchie desarrollaron un juego de "Viaje espacial" para la PDP-7. Luego crearon una nueva estructura de sistemas de archivos y nuevo software. Le añadieron un entorno de procesos con planificación y completaron un sistema operativo rudimentario. Al resultado se le llamó UNIX, haciendo referencia a que era una simplificación de MULTICS. La primera versión de UNIX se codificó en lenguaje ensamblador. Un año después Ritchie instaló el sistema en una DEC PDP-11.

La segunda edición de UNIX fue realizada en 1971. La segunda edición dio forma al UNIX moderno con la introducción del lenguaje de programación C y del concepto de los pipes. Los pipes fueron importantes por muchas razones. Representaron una nueva forma de tratamiento de datos. Desde un punto de vista moderno, los pipes son un mecanismo orientado a objetos, porque entregan datos desde un objeto, o programa, a otro objeto.

El lenguaje C es otro producto de los Laboratorios Bell. Fue formado a partir de conceptos de otros tres lenguajes: B, CPL (Combined Programming Language) y Algol-60. A finales de 1973, después de que Ritchie añadió soporte para variables globales y estructuras, C se convirtió en el lenguaje de programación de UNIX de manera preferente.

Así se rompió la tradición de escribir sistemas operativos en lenguaje ensamblador. Sólo algunas subrutinas de alto rendimiento dentro del núcleo del sistema están escritas en ensamblador. Esto permitió hacer a UNIX muy portable y ganar popularidad.

El ascenso del C fue responsable del concepto de portabilidad. Escrito en C, el entorno UNIX pudo ser relativamente fácil de trasladar a diferentes plataformas hardware. Las aplicaciones escritas en C pudieron ser fáciles de transportar entre diferentes variantes de UNIX. En esta situación nació el primer criterio de sistema abierto: portabilidad OS, la posibilidad de mover software desde una plataforma hardware a otra de una forma estándar. La portabilidad de UNIX se convirtió en el modelo de transportar aplicaciones en C desde un sistema UNIX a otro.

AT&T distribuyó copias de Unix a universidades de todo el mundo, lo cual propició más innovaciones y la Universidad de California en Berkeley desarrolló su propia implementación: BSD (Berkeley Software Distribution), la cual es ampliamente utilizada. Al tiempo que AT&T fortalecía y optimizaba Unix con un enfoque comercial, la implementación BSD dominaba en los ambientes universitarios y técnicos.

A finales de los setenta, AT&T introdujo un nuevo esquema de denominación para su versión del sistema Unix. Hasta entonces, las revisiones importantes se referían a las nuevas versiones que emergían del área de investigación de los laboratorios Bell. AT&T cambió su numeración a sistema III y sistema V. El sistema V sustituyó al sistema III a mediados de los ochenta. El sistema IV se utilizó dentro de los laboratorios Bell, pero se consideró un producto de transición. AT&T normalizó el nombre de sistema V y sus versiones recientes se denominan sistema V revisión 3 y sistema V revisión 4, que con frecuencia se abrevian como SVR3 y SVR4, respectivamente. Las versiones BSD fueron la plataforma dominante para experimentación en conceptos de red. El primer experimento importante en redes de área extensa, el ARPANET, fue orientado a UNIX.

### 1.3. Evolución de Linux

La historia de Linux empieza en Finlandia, en el 91, cuando a un estudiante de la Universidad de Helsinki llamado Linus B. Torvalds se le ocurrió comprarse un PC con procesador 386. Después de observar que con el MSDOS no aprovechaba los recursos de la máquina, decidió usar otro sistema operativo de entonces: Minix. Minix era un pequeño sistema Unix.

Sin embargo, debido a las limitaciones del Minix, Linus decidió reescribir algunas partes del sistema, añadiéndole mayor funcionalidad. Posteriormente, Linus decidió difundir el código fuente por Internet, de manera gratuita y con el nombre de Linux (contracción de Linus y Unix). La primera difusión de Linux tuvo lugar el mes de agosto de 1991. Se trataba de la versión 0.01 y por el momento, funcionaba bajo Minix.

Esta primera versión era lo que se podría denominar un embrión, ni siquiera hubo anuncio oficial. La primera versión "oficial", la 0.02, se hizo pública el 5 de octubre de 1991. En esta nueva versión ya se permitía el uso de algunos programas GNU, como bash, gcc, ...

En estas primeras versiones, Linux era bastante limitado, pero el hecho de que se difundiera el código fuente por Internet, y totalmente gratis, hizo que cada vez más personas empezaran a colaborar con el proyecto, llegando a los cientos de colaboradores que hay trabajando en la actualidad en los cientos de proyectos GNU.

La primera versión estable de Linux fue la 1.0 y apareció en marzo del 94. El número de versión asociado al núcleo tiene un sentido muy particular ya que está ligado a su desarrollo, la evolución de Linux se efectúa en dos fases:

- Fase de desarrollo: el núcleo cuya estabilidad no está asegurada, es el momento donde se añade funcionalidad al núcleo, optimizaciones y demás. En definitiva, es la fase en la que se desarrolla más el núcleo y se caracteriza por su nombre de versión impar: 1.1 , 1.3 ...
- Fase de estabilización: se trata de coger el núcleo desarrollado en la fase anterior, y hacer que este sea lo más estable posible. Aquí las modificaciones son mínimas, se trata más de retoques y pequeños ajustes. Los núcleos estables tiene número de versión par: 1.0, 1.2, 2.0, 2.4 ...

Actualmente, Linux es un sistema Unix completo, estable, que sigue evolucionando y que cada día gana nuevos adeptos. Durante muchos años Linux perteneció, casi por completo al mundo universitario, ahora que Internet llega a millones de usuarios, Linux se está extendiendo a pasos agigantados, incluso en el mundo empresarial.

Inicialmente Linux se diseñó como un clónico de Unix, distribuido libremente para funcionar en máquinas PC con procesadores 386, 486... o sea, la para arquitectura x86. En la actualidad funciona sobre otras muchas plataformas como los procesadores Alpha, Sparc, Amiga, Atari, las máquinas tipo MIPS y sobre los PowerPC.

Hay que resaltar también que Linux respeta las especificaciones POSIX, pero posee también ciertas extensiones de las versiones System V y BSD de Unix. Esto simplifica notablemente la adaptación de programas desarrollados inicialmente para otros sistemas Unix

El termino POSIX significa Portable Operating System Interface. Son unas normas definidas por el IEEE y estandarizados por el ANSI y el ISO. POSIX permite tener un código fuente transportable. Gracias a Internet, Linux a tenido un crecimiento espectacular en los últimos tiempos, haciendo que cada vez más personas trabajen en el proyecto y estos, a la vez, haciendo al sistema cada vez más y más potente.

Hay que recordar también que el término Linux se refiere al Núcleo del sistema (lo que interactúa con el Hardware de nuestra máquina). Cuando hablamos de todo el conjunto que forman el núcleo, y todos los demás proyectos GNU (las shells, compiladores, escritorios y las distintas aplicaciones en general), estaremos hablando ya del Sistema Operativo GNU/Linux.

Por último señalar que, sin ninguna duda, Linux es la implementación libre (o gratuita) de UNIX más conocida. Pero no es la única. Existen algunas otras con tantas capacidades como Linux, e incluso funcionando en más plataformas que Linux. Algunos ejemplos son:

- NetBSD <http://www.netbsd.org>
- OpenBSD <http://openbsd.org>
- FreeBSD <http://freebsd.org>

## 1.4. Distribuciones

Linux es un sistema operativo de libre distribución. Se pueden encontrar en multitud de servidores conectados a Internet, todos los ficheros y programas necesarios para su funcionamiento. Sin embargo, y debido a que la tarea de reunir todos los ficheros necesarios e instalarlos correctamente puede ser bastante compleja, aparecieron empresas que se dedicaron a hacer este trabajo, y aquí surgió el término de distribución.

Una distribución no es más que una recopilación de programas y ficheros (incluyendo la última versión estable del núcleo) , organizados y preparados para su instalación.. Estas distribuciones se pueden obtener a través de Internet o comprando los CDs de las mismas. La mayoría del software que incluyen las distribuciones suele tener licencia GNU.

Normalmente la obtención de las distribuciones por Internet suele ser gratuita. Y si se obtiene la distribución mediante los CDs, suele tener un coste en general bastante aceptable teniendo en cuenta la gran cantidad de software que incluyen, así como manuales e incluso soporte durante un periodo de tiempo (de 3 a 6 meses gratuitos).



Actualmente sería difícil precisar el número de distribuciones existente de Linux, no obstante hay datos que apuntan a que son cerca de 300 distribuciones diferentes. A continuación se describen brevemente algunas distruciones Linux más conocidas:

## **REDHAT**

RedHat es, sin duda, una de las más populares e importantes distribuciones de Linux. Se dice que RedHat personifica Linux para el mundo, ya que son responsables de liderar muchos de los más importantes proyectos open source, así como tecnologías base que acaban, casi siempre, en nuestro sistema Linux.

Para el tema de la instalación, se utiliza un programa en modo gráfico, por pasos, y de una forma bastante fácil.

Otra de las características a recordar, es la formato de los paquetes de software, llamado RPM. La mayoría de distribuciones basadas en RedHat, suelen adoptar el formato RPM para gestionar dichos paquetes. Eso no quiere decir que estas distribuciones no puedan soportar otro de tipo de paquetes, por ejemplo, en el formato de Debian.

<http://redhat.com>

## **DEBIAN**

Debian es la única distribución GNU/Linux NO COMERCIAL. Está desarrollada por programadores de todo el mundo con el objetivo común de construir el sistema operativo basado en software libre lo más robusto posible. De hecho, Debian es una de las distribuciones más imponentes y poderosas. Esta formado por más de 3000 paquetes.

Debian tiene la fama de ser una distribución para usuarios avanzados, por ejemplo, en el tema de la instalación, ésta es completamente en modo texto, y automatiza muy poco las cosas. A cambio de perder comodidad, Debian permite una flexibilidad sorprendente, siempre que sepas lo que estás haciendo... además, Debian es la distribución GNU/Linux que corre en más arquitecturas, desde los x86 (Intel, AMD ...), Alpha, ARM, Power PC y algunos más.

Quizá Debian no sea la mejor distribución para empezar en el mundo Linux, debido a la más o menos complejidad comentada antes, pero cabe resaltar que es una de las más robustas, la gente de Debian no incorpora un paquete hasta que este esté infinitamente probado, al no tener exigencias comerciales, esto hace de Debian una distribución diferente.

Debian gestiona sus paquetes con su propio formato, o sea, que no usa los RMPs de RedHat.

<http://www.debian.org>

## **MANDRAKE**

Es una de las distribuciones más fáciles de usar. Mandrake es ideal para todos los novatos, a parte de ser una distribución muy completa. Está basada en RedHat.

Su filosofía es hacer las cosas lo más sencillas al usuario. Aspecto que notaremos nada más empezar la instalación, esta es: gráfica, muy sencilla, por pasos... desde luego, en este aspecto, es una de las mejores.

En el momento de la instalación, tu decides si quieres preparar tu máquina como un servidor, como una estación de trabajo, una máquina para desarrolladores... en el tema de escritorios, se incluyen los últimos GNOME y KDE, así como unos cuantos más. Junto con KDE2, también incorpora Koffice. En uno de los pasos de instalación podrás elegir la política de seguridad de tu máquina de entre 3 opciones, seguridad baja, media y alta.

Mandrake, sin duda, está enfocado a usuarios noveles, y no para empresas. Aunque permite montar y administrar un servidor, esta no es ni de lejos su obsesión, que no es otra que la de facilitar la entrada a este fantástico mundo, de la forma más sencilla y agradable.

<http://linux-mandrake.com>

## SUSE

Así como la RedHat es la reina en Norteamérica, en Europa es SuSe quien se lleva la palma. Es una distribución alemana, y enfocada, descaradamente, a los negocios. La distribución de SuSe contiene gran cantidad de paquetes (unos 2000), y afortunadamente, unos buenos y completos manuales (de hecho, las versiones más recientes contienen 3 manuales, que en total nos dan unas 1000 páginas de información en castellano).

Suse, al igual que RedHat y alguna que otra distribución más, tiene más de una versión. Una, enfocada a usuarios normales, y la otra destinada a las empresas para montar potentes servidores. Al igual que Debian, funciona en un montón de arquitecturas distintas.

El programa de instalación el YaST2, nos permitirá instalar cómoda y sencillamente el sistema, también nos servirá para administrarlo una vez instalado. También usa los RPM para la gestión de paquetes software.

SuSe también destaca por su política de soporte a sus productos (ya decíamos que está enfocado a los negocios...), y, además, deber ser la única capaz de sacar al mercado tres versiones al año.

# 2. APROXIMACIÓN A LINUX

## 2.1. Características generales

A continuación se enumeran algunas de las características generales de Linux, obtenidas de la documentación oficial de Linux: FAQ\_Linux\_V2.0.2 (<http://www.linux-es.org/Faq>):

- **Multitarea:** La palabra multitarea describe la habilidad de ejecutar varios programas al mismo tiempo. Linux utiliza la llamada multitarea preeventiva, la cual asegura que todos los programas que se están utilizando en un momento dado serán ejecutados, siendo el sistema operativo el encargado de ceder tiempo de microprocesador a cada programa.
- **Multiusuario:** Muchos usuarios usando la misma máquina al mismo tiempo.
- **Multiplataforma:** Las plataformas en las que en un principio se puede utilizar Linux son 386-, 486-, Pentium, Pentium Pro, Pentium II/III, Pentium 4, Amiga y Atari, también existen versiones para su utilización en otras plataformas, como Alpha, ARM, MIPS, PowerPC y SPARC.
- **Multiprocesador:** Soporte para sistemas con mas de un procesador está disponible para Intel y SPARC.
- **Protección de la memoria entre procesos,** de manera que uno de ellos no pueda colgar el sistema.
- **Carga de ejecutables por demanda:** Linux sólo lee del disco aquellas partes de un programa que están siendo usadas actualmente.
- **Política de copia en escritura para la compartición de páginas entre ejecutables:** esto significa que varios procesos pueden usar la misma zona de memoria para ejecutarse. Cuando alguno intenta escribir en esa memoria, la página (4Kb de memoria) se copia a otro lugar. Esta política de copia en escritura tiene dos beneficios: aumenta la velocidad y reduce el uso de memoria.
- **Memoria virtual usando paginación (sin intercambio de procesos completos) a disco:** A una partición o un archivo en el sistema de archivos, o ambos, con la posibilidad de añadir más áreas de intercambio sobre la marcha. Un total de 16 zonas de intercambio de 128Mb de tamaño máximo pueden ser usadas en un momento dado con un límite teórico de 2Gb para intercambio.
- **La memoria se gestiona como un recurso unificado para los programas de usuario y para el caché de disco,** de tal forma que toda la memoria libre puede ser usada para caché y ésta puede a su vez ser reducida cuando se ejecuten grandes programas.
- **Se realizan volcados de estado (core dumps) para posibilitar los análisis post-mortem,** permitiendo el uso de depuradores sobre los programas no sólo en ejecución sino también tras abortar éstos por cualquier motivo.

- Compatible con POSIX, System V y BSD a nivel fuente.
- Todo el código fuente está disponible, incluyendo el núcleo completo y todos los drivers, las herramientas de desarrollo y todos los programas de usuario; además todo ello se puede distribuir libremente. Hay algunos programas comerciales que están siendo ofrecidos para Linux actualmente sin código fuente, pero todo lo que ha sido gratuito sigue siendo gratuito.
- Consolas virtuales múltiples: varias sesiones de login a través de la consola entre las que se puede cambiar con las combinaciones adecuadas de teclas (totalmente independiente del hardware de video).
- Soporte para varios sistemas de archivo comunes, incluyendo minix-1, Xenix y todos los sistemas de archivo típicos de System V, y tiene un avanzado sistema de archivos propio con una capacidad de hasta 4 Tb y nombres de archivos de hasta 255 caracteres de longitud.
- Acceso transparente a particiones Windows (FAT, FAT32, NTFS) mediante un sistema de archivos especial: no es necesario ningún comando especial para usar la partición Windows, ésta parece un sistema de archivos normal de Unix.
- TCP/IP, incluyendo ftp, telnet, NFS, etc.
- Diversos protocolos de red incluidos en el kernel: TCP, IPv4, IPv6, AX.25, X.25, IPX, DDP, Netrom, etc.

## 2.2. Modo gráfico vs Modo comando

Cómo ya ha quedado bastante claro, Linux (sería más correcto decir GNU/Linux ....) es un sistema operativo basado en UNIX. UNIX es un sistema operativo basado en el modo comando, es decir, la interacción entre el sistema operativo y el usuario se realiza a través de una línea de comandos (igual que el conocido MSDOS). El usuario tiene que teclear el comando que realiza la acción deseada y pulsar Enter para el que sistema operativo ejecute la orden. Linux se basa en la misma filosofía.

Sin embargo, debido al auge que han tenido los sistemas operativos basados en una interfaz gráfica, se desarrolló el protocolo XWindow, sobre todo para sistemas tipo UNIX. XWindow es el encargado de visualizar la información de manera gráfica y es totalmente independiente del sistema operativo (los sistemas Unix / Linux no necesitan XWindow para funcionar, pudiendo trabajar en modo texto). De hecho, el sistema XWindow se comporta a todos los efectos como una aplicación más funcionando en un sistema Linux.

En la práctica, y para facilitar el manejo del sistema a usuarios noveles o que vienen del entorno Windows, prácticamente todas las distribuciones están configuradas para que, por defecto, arranquen un gestor de ventanas basado en el sistema XWindow. Además, se incluyen también muchas utilidades y software que utiliza el sistema gráfico para funcionar.

Y, aunque es tentador trabajar en Linux utilizando exclusivamente la interfaz gráfica. Conviene recordar que no hay un estándar claro en cuanto a gestores de ventanas y clientes XWindows. Lo que sí es estándar es el modo comando o también llamado modo texto (aunque hay ciertos

comandos que son diferentes para sistemas System V o BSD, las diferencias son mínimas). Además, como regla general se puede decir que todo lo que se puede hacer en modo gráfico, se puede hacer en modo texto, utilizando comandos. Sin embargo, no todo lo que se puede hacer en modo texto, se puede hacer en modo gráfico.

Hay una pregunta que siempre aparece en este punto:

### **¿Pero para utilizar Linux hay que saber comandos?**

A continuación se incluye un extracto del documento FAQ\_Linux\_V2.0.2 (<http://www.linux-es.org/Faq>), apartado 14.3:

*La respuesta a esta pregunta es sencilla, todo depende de como vayas a utilizar el sistema, de las ganas de aprender que tengas y de lo que quieras profundizar en el sistema.*

*Hay que diferenciar entre tareas de administración y uso habitual del sistema.*

*Si vas a realizar tareas de administración del sistema, deberías saber, o mejor dicho, tendrás que aprender una serie de comandos para realizar estas tareas. Si vas a utilizar el sistema a nivel usuario, no necesitaras saber ningún comando, solamente tendras que aprender a utilizar los programas que uses.*

*Lo que ocurre habitualmente es que el usuario de la máquina y el administrador, son la misma persona, con lo cual tendrán que saber una serie de comandos mínimos.*

*Como anecdotita, os puedo contar que en lugar donde trabajo, hay usuarios que utilizan Linux y no tienen ni idea de administración del sistema (y no solo en linux), no saben comandos y tampoco los necesitan. Trabajan en modo gráfico, saben utilizar los programas que necesitan y tienen a alguien que se hace cargo de las tareas de administración/configuración del sistema.*

En definitiva, la línea de comandos nos ofrece una gran flexibilidad y eficiencia en el uso del sistema operativo Linux.

En el proceso de arranque de Linux se crean lo que se llaman consolas virtuales, que son ventanas de tipo texto para poder trabajar en modo comando. Estas consolas virtuales son accesibles incluso desde el modo gráfico.

Si utilizamos una configuración de arranque de Linux en modo texto el sistema nos presentará automáticamente en una de estas consolas virtuales. Para conmutar a otra se utilizan las combinaciones de teclas Alt+F1 hasta Alt+F6. Es decir, hay seis consolas virtuales.

Si utilizamos una configuración de arranque de Linux en modo gráfico, para acceder a estas consolas desde el modo gráfico se utilizan las combinaciones Ctrl+Alt+F1 hasta Ctrl+Alt+F6.

Por último, para volver al modo gráfico desde cualquier consola virtual hay que pulsar Alt+F7.

# 3. TRABAJO EN MODO COMANDO

## 3.1. Acceso a un sistema Linux en modo comando

Una de las características que se han mencionado sobre Linux es que es multiusuario, es decir, es un sistema operativo preparado para funcionar con varios usuarios. Por ello, tanto si trabajamos en modo gráfico o en modo comando, después del arranque, lo primero que nos pide el sistema es validarnos. Para ello, habrá que introducir un nombre de usuario (login) y su contraseña (password).

Ejemplo de acceso en modo comando:

```
Red Hat Linux release 9 (Shrike)
Kernel 2.40.20-8 on an i686

localhost login:
```

Cuando trabajamos en modo comando, después de validar el usuario, el sistema nos presenta el llamado prompt que es un símbolo que indica que el sistema está preparado para aceptar comandos. Los símbolos utilizados como prompt son \$ y %. Para el usuario administrador, llamado root, en sistemas Unix, el símbolo del prompt es #.

Lo primero que hay que tener en cuenta antes de empezar a conocer comandos es que, por lo general, **en Linux se distingue entre mayúsculas y minúsculas**. Esto es aplicable tanto para nombres de ficheros y directorios como para los comandos. Para Linux mi\_fichero no es lo mismo que Mi\_fichero ó MI\_FICHERO. Si queremos ejecutar el comando pwd, debemos teclear pwd y no Pwd ó PWD.

## 3.2. La shell

Después de producirse la validación del usuario se ejecuta un programa de gran importancia en el sistema y que se conoce con el nombre genérico de **shell**. Este proceso es el encargado, entre otras cosas de mostrar la línea de comandos por la pantalla y de leer el comando que tecleemos para ejecutarlo. Es equivalente al intérprete de comandos COMMAND.COM de MSDOS. En Unix en general, el sistema nos permite utilizar varias shells. Los nombres de las shells más conocidas son: bash, sh, ksh, tcsh, csh, ...

Cada usuario tiene asignada la shell que se ejecutará cuando acceda al sistema. En Red Hat, por defecto se utiliza la shell bash (Bourne Again Shell) que es una evolución de la primitiva shell sh (Bourne Shell).

De la shell que utilicemos dependerán algunos aspectos, aunque en general, las diferencias sólo serán apreciadas por usuarios avanzados. Sin embargo hay algunas características comunes a las shell más avanzadas y que facilitan mucho el trabajo en línea de comandos:

- El historial: la shell almacena un historial de las órdenes tecleadas anteriormente. Para visualizar los comandos ejecutados anteriormente se utilizan las teclas del cursor arriba y abajo.
- La función de expansión: esta función completa el nombre del fichero del cual sólo se han introducido las primeras letras, en caso de que éste puede ser identificado de forma inequívoca. Para ello hay que pulsar la tecla de tabulación y el nombre se completará.

### 3.3. Primeros comandos

Cuando el sistema presenta el prompt significa que ya está preparado para aceptar los comandos que tecleemos. La forma general de los comandos en Linux es:

```
$ comando [opciones] [parámetros]
```

Las opciones son modificadores para los comandos y no siempre es necesario teclear alguna. Las opciones se suelen indicar con el signo – seguido por una letra. Los parámetros son informaciones que el comando puede necesitar para ejecutarse. Esta necesidad depende de cada comando concreto.

A continuación vamos a ver algunos comandos básicos.

#### Comando para cambiar la contraseña: passwd

```
$ passwd
```

El sistema pide la contraseña antigua y la nueva dos veces. Después de su ejecución la contraseña para el usuario que ha ejecutado el comando habrá sido cambiada.

#### Comandos para salir de la sesión: logout o exit

```
$ logout
```

El sistema finaliza la sesión y vuelve a mostrar la petición de login para una nueva sesión. El comando exit es equivalente.

#### Comando de ayuda: man

Uno de los comandos más importantes del modo texto es el comando man. La forma de ejecutarlo es teclear man seguido del nombre del comando sobre el que queremos obtener ayuda. Por ejemplo:

```
$ man passwd
```

La ayuda que ofrece el comando man se presenta paginada y formateada con algún formateador de textos que esté instalado en nuestro sistema. Normalmente funcionarán las teclas de cursores y las de RePg y AvPg para desplazarnos por el texto de ayuda. Para salir de nuevo al prompt del sistema pulsar la tecla q.

#### Comando para cambiar de usuario: su

Con este comando podemos realizar alguna acción utilizando la identidad de otro usuario sin necesidad de salir de la sesión en la que estamos. Se teclea su, seguido del nombre del usuario. Si

no tecleamos el nombre de un usuario, el sistema interpreta que queremos cambiar al usuario root. En cualquier caso nos pedirá la contraseña.

### **Comandos para salir del sistema y apagar el equipo: halt, poweroff y shutdown.**

Con estos comandos se realizarán todas las acciones necesarias para la salida ordenada del sistema. Después de su ejecución se podrá apagar el equipo. El comando shutdown sólo se puede ejecutar por el usuario root.

### **Comando para reiniciar el sistema: reboot**

## **3.4. Estructura de directorios**

Una de las principales ideas que hay que saber cuando se trabaja en Linux es la organización de la información en las distintas unidades de almacenamiento.

Al igual que en otros sistemas operativos, la información se almacena en ficheros y estos a su vez se pueden agrupar en directorios y subdirectorios formando una estructura en árbol similar a los sistemas MSDOS/Windows. La gran diferencia respecto a estos sistemas es que en Linux (y Unix en general) existe un solo árbol de directorios y no uno para cada partición del sistema. Cualquier disco duro, partición, disquete, CD-ROM o dispositivo de almacenamiento debe formar parte de esta estructura única.

A la operación por la cual un dispositivo de almacenamiento entra a formar parte del árbol de directorios del sistema se denomina **montaje del dispositivo**. Más adelante veremos esta operación en detalle.

Existe un directorio que contiene el resto de archivos y directorios del sistema. A este directorio se le llama **directorio raíz** (root) (no confundir con el usuario root o administrador). De este directorio nace toda la estructura en árbol del sistema de directorios de Linux. El directorio raíz se representa siempre con el símbolo de barra inclinada (/).

En cuanto a los nombres del resto de directorios y archivos, se utilizan restricciones parecidas a los sistemas Windows con la diferencia, como ya se ha comentado, que se distingue entre mayúsculas y minúsculas.

Otro concepto importante: cuando trabajamos en modo comando hay que tener en cuenta siempre el directorio donde nos encontramos. Ya que muchos comandos que hacen referencia a directorios utilizan el directorio actual si no se especifica otra cosa. Al directorio donde nos encontramos se le llama **directorio actual o directorio activo**. Y existe un comando para conocerlo: **pwd**

Como ya hemos visto, cuando un usuario accede a un sistema Linux, debe hacerlo con un login (nombre) y un password (contraseña). Si el sistema reconoce al usuario le permite el acceso, le permite la escritura de comandos (arranca un intérprete de comandos) y le “deposita” en un directorio. Al directorio donde se ubica a un usuario concreto se le llama directorio home.

El **directorio home** es el directorio asignado a cada usuario de un sistema Linux. En él, los usuarios pueden crear archivos y directorios propios, a los que el usuario puede controlar el acceso de otros



usuarios. La mayoría de los usuarios trabajan regularmente con los archivos y directorios que crean bajo su directorio home.

Para hacer referencia a un directorio o fichero, se utiliza lo que se llama **ruta**, es decir, la descripción de todos los directorios por los que hay que pasar para llegar al directorio al que hacemos referencia. Por ejemplo

La ruta se puede especificar de dos formas:

**Ruta absoluta:** es la ruta completa desde el principio del árbol hasta el directorio al que nos referimos. Una ruta absoluta siempre comienza con el directorio raíz (/).

**Ruta relativa:** es la ruta a un fichero o directorio desde el directorio actual. Es decir, es el camino de directorios que hay que recorrer, partiendo del directorio actual.

**Directorio actual, directorio ascendente y directorio anterior:**

Al especificar una ruta, se puede hacer referencia al directorio actual mediante el símbolo punto (.). Para referirse al directorio padre del directorio actual se utilizan dos puntos (..). Y para hacer referencia al directorio en el que se encontraba anteriormente, se utiliza el signo menos (-) (sólo en la shell Korn)

### 3.5. Comandos para trabajar con la estructura de directorios

A continuación vamos a conocer más comandos utilizados para movernos por la estructura de directorios del sistema:

#### **pwd**

Muestra por pantalla el nombre del directorio actual. Ya se mencionó en el anterior apartado.

#### **cd**

Cambia el directorio actual por el especificado como parámetro. Ejemplos:

cd /usr/local/doc	Ejemplo del comando cd utilizando una ruta absoluta
cd local/doc	Suponiendo que nos encontramos en /usr, igual que el anterior pero usando ruta relativa
cd ..	Cambio al directorio un nivel por encima del actual
cd ../usuario1/apuntes	Ejemplo del comando cd utilizando una ruta relativa
cd	Cambio al directorio home del usuario.

**A la hora de escribir rutas para el comando cd o para cualquier otro comando recordar la utilidad del tabulador que nos proporciona la shell de comandos.**

#### **ls**

Visualiza el contenido del directorio especificado como parámetro. Si no se indica ningún directorio se utiliza el directorio actual. Este comando tiene multitud de opciones para alterar el tipo de información mostrada. Por ejemplo, la opción -l muestra información extendida sobre cada fichero o directorio. Ejemplos:

ls -l	Lista extendida de archivos y directorios del directorio actual.
-------	--

<code>ls -l /var/log</code>	Lista extendida del directorio <code>/var/log</code>
<code>ls -a</code>	Lista de todos los archivos y directorios, incluidos los ocultos, que en Linux empiezan por el carácter punto ( <code>.</code> )
<code>ls -R</code>	Lista de archivos y directorios y de todos los directorios dentro de él.
<code>ls -C</code>	Lista de archivos en columnas.
<code>ls -lt</code>	Lista extendida de archivos ordenada por fecha.

### ***mkdir directorio***

Creación de nuevos directorios.

### ***rmdir directorio***

borrado de directorios vacíos. Para borrar todo: `rm -rf nombre_directorio`.

## **3.6. Jerarquía estándar de directorios**

En los sistemas Unix las ubicaciones de los ficheros en el sistema siguen unas normas determinadas con el objeto de aumentar el nivel de organización. La mayoría de las distribuciones de Linux siguen el llamado estándar de jerarquía del sistema de ficheros FHS (Filesystem Hierarchy Standard) (más información sobre este estándar en <http://www.pathname.com/fhs/> ). Algunos de los directorios más importantes son los siguientes:

### ***/bin y /usr/bin***

Estos directorios contienen la mayoría de los ficheros ejecutables y comandos más comunes del sistema Linux.

### ***/sbin y /usr/sbin***

Estos directorios también contienen comandos y ficheros ejecutables normalmente ejecutados en tareas de administración con lo cual muchos de ellos sólo son ejecutables para el usuario root o administrador del sistema.

### ***/etc***

Este directorio contiene los ficheros de configuración de todo el sistema. Con local es de vital importancia. Normalmente los ficheros que contiene son ficheros de configuración de tipo texto sin formato. Además, suelen tener sólo permisos de lectura para usuarios normales, es decir, sólo el usuario root los puede modificar.

### ***/root***

Directorio home del usuario root.

### ***/usr***

Directorio destinado a almacenar las aplicaciones, con lo cual su tamaño puede ser elevado si existen muchos paquetes de software instalados. Suele ser una buena opción utilizar una partición separada para este directorio.

**/home**

Directorio donde se almacenan todos los directorios home de los usuarios del sistema, por tanto, en función del número de usuarios y del uso que hagan del sistema, este directorio puede llegar a necesitar también mucho espacio. También es recomendable utilizar una partición separada para este directorio.

**/lib y /usr/lib**

Directorios que contienen librerías compartidas del sistema.

**/tmp y /usr/tmp**

Directorios para almacenar ficheros temporales.

**/boot**

Directorio que contiene los ficheros necesarios para el arranque del sistema. Por ejemplo, aquí se almacenan los ficheros del gestor de arranque si hubiera alguno instalado (LILO, GRUB). También se suele almacenar las imágenes del kernel o núcleo del sistema. Algunas distribuciones aconsejan utilizar una pequeña partición separada para este directorio.

**/dev**

Directorio que almacena archivos de dispositivos. Estos archivos no son realmente archivos si no que una forma en la que los sistemas Linux implementan los controladores de dispositivos. Se hablará más al respecto de estos archivos en el apartado correspondiente.

**/var**

Directorio que contiene información variable en general como colas de impresión, colas de envío y recepción de correos y news, archivos de registro y de eventos del sistema,... En sistemas Linux utilizándose en servidores este directorio puede necesitar mucho espacio con lo cual es recomendable utilizar una partición propia.

**/opt**

Directorio opcional donde se pueden instalar aplicaciones, además de /usr.

**/proc**

Este directorio se corresponde con un sistema de fichero virtual creado por el kernel en memoria. Sirve de interfaz con los parámetros de configuración del kernel.

**/mnt**

Directorio usado por defecto por el sistema para realizar el montaje de otros dispositivos de almacenamiento como disquetes y CDROMs.

### 3.7. Propiedades del sistema de ficheros

Los elementos que podemos encontrar en el sistemas de ficheros, es decir, tanto ficheros como directorios poseen una serie de características o propiedades que pueden visualizarse como hemos visto con el comando `ls -l`. Vamos a descubrir el significado de toda esa información.

Ejemplo de salida del comando `ls -l`:

```
-rw----- 1 root    root      14596 mar 16 01:44 boot.log
-rw----- 1 root    root        3205 mar 16 02:01 cron
drwxr-xr-x 2 lp      sys       4096 mar  5 02:05 cups
-rw-r--r-- 1 root    root      5951 mar 16 01:43 dmesg
drwxr-xr-x 2 root    root      4096 mar 16 01:44 gdm
-rw-r--r-- 1 root    root     65404 mar 16 01:43 ksyms.0
-rw-r--r-- 1 root    root     65404 mar  9 19:50 ksyms.1
-rw-r--r-- 1 root    root     65404 mar  8 23:43 ksyms.2
-rw-r--r-- 1 root    root     69134 mar  5 01:47 ksyms.3
-r----- 1 root    root    19136220 mar 16 01:49 lastlog
-rw----- 1 root    root      3242 mar 16 01:44 maillog
-rw----- 1 root    root     77967 mar 16 02:02 messages
-rw-r--r-- 1 root    root     16109 mar  9 20:56 rpmpkgs
-rw-r--r-- 1 root    root     11301 mar  5 01:39 scrollkeeper.log
-rw----- 1 root    root       841 mar 16 01:45 secure
-rw----- 1 root    root         0 mar  5 01:16 spooler
drwxr-xr-x 2 root    root      4096 feb  4 2003 vbox
-rw-rw-r-- 1 root    utmp     53760 mar 16 02:03 wtmp
-rw-r--r-- 1 root    root     43584 mar 16 02:02 XFree86.0.log
-rw-r--r-- 1 root    root     46606 mar  9 23:55 XFree86.0.log.old
-rw-r--r-- 1 root    root     42948 mar  5 01:48 XFree86.1.log
```

Los campos que aparecen en este listado son los siguientes:

Permisos	NL	Prop	Grupo	Tam	Fecha	Hora	Nombre
drwxr-xr-x	2	lp	sys	4096	mar	5 02:05	cups
-rw-r--r--	1	root	root	5951	mar	16 01:43	dmesg
drwxr-xr-x	2	root	root	4096	mar	16 01:44	gdm

Permisos: Descrito a continuación.

NL: Número de enlaces del fichero (descrito en el apartado sobre enlaces).

Prop: Nombre del propietario o dueño del fichero o directorio.

Grupo: Nombre del grupo al que pertenece el fichero o directorio.

Tam: Tamaño del fichero, en bytes.

Fecha y Hora: Indica la fecha y la hora de creación o modificación del fichero.

Nombre: Nombre del fichero.

Una de las principales características del sistema de ficheros usado en linux es que posee un robusto sistema de permisos. Cada fichero del sistema (en este apartado hablaremos de permisos de ficheros pero lo mismo se puede aplicar a directorios) tiene una serie de permisos que definen su accesibilidad a todos los usuarios del sistema. Para ello se define el siguiente esquema

-	rwX	rwX	rwX
tipo	Usuario	Grupo	Otros

El primer carácter indica el tipo de fichero:



Para avanzar línea a línea, utilizar la tecla de Return. Para avanzar de página en página, utilizar la tecla de espacio. Para salir, tecla q.

### **less [nombre\_fichero]**

Parecido a more. Muestra por pantalla de forma paginada, el contenido de un archivo de texto. Tiene más funcionalidades que more. Por ejemplo, con less se pueden utilizar las teclas de cursor para moverse a través del texto.

### **cp fichero .... directorio**

Copia ficheros y directorios. Si el último argumento es un directorio, cp copia cada fichero fuente a ese directorio. Si el último argumento es un fichero, cp copia el primer argumento que deberá ser un solo fichero, en el fichero destino.

### **mv origen ... destino**

Comando que mueve o renombra ficheros o directorios. Si el último argumento es un directorio, mv mueve cada uno de los anteriores ficheros a ese directorio. Si se pasan como argumento dos nombres de ficheros (o dos nombres de directorios), renombra el primero al segundo.

### **rm fichero...**

Comando para borrar ficheros. Con la opción -r se borra la estructura entera de un directorio, incluido los subdirectorios.

### **find ruta\_búsqueda opciones\_búsqueda**

Busca ficheros en la estructura de directorios que cumplan la condición especificada. Este es un comando muy potente de Linux que admite muchas opciones de búsqueda diferentes. Algunos ejemplos de opciones de búsqueda:

-name 'nombre'	busca archivos con el nombre especificado.
-perm DDD	busca archivos con los permisos especificados DDD.
-user USER	busca archivos que pertenezcan al usuario USER.
-size +-bloques	busca archivos con más (+) o menos (-) tamaño al especificado.
-group	busca archivos que pertenexcan al grupo especificado.

Ejemplos:

```
$ find . -name 'prueba'      busca en el directorio actual archivos que se llamen prueba.  
$ find /tmp -user usuario1   busca en el directorio /tmp archivos que pertenezcan a usuario1.  
$ find /var -size +1000000c  busca en el directorio /var archivos superiores a 1000000 bytes.
```

### **cmp fichero1 fichero2**

Compara dos ficheros de cualquier tipo y escribe el resultado en la salida estándar.

### **diff fichero1 fichero2**

Busca diferencias entre dos ficheros. Parecido a cmp pero más avanzado. Por ejemplo, puede comparar dos directorios.

### **type fichero\_ejecutable**

Muestra por pantalla la ruta completa de un archivo ejecutable.

### **file fichero**

Indica el tipo de fichero pasado como parámetro.

**whereis fichero\_ejecutable**

Muestra la ubicación del archivo ejecutable de un comando, el archivo fuente y la página del manual correspondiente. Esta búsqueda se realiza dentro de los directorios especificados en la variable de entorno PATH.

**chmod permisos fichero...**

Comando para establecer los permisos de un fichero o directorio. Los permisos de un fichero se pueden cambiar sólo por el propietario del fichero o por el usuario root. Existen dos formas de establecer los permisos. La primera utilizando una representación simbólica de los cambios a realizar y la segunda utilizando un número octal que representa el patrón de bits de los nuevos permisos.

Para el primer formato se utiliza un patrón formado por:

1. Las categorías afectadas:
  - u: para el propietario
  - g: para el grupo
  - o: para el resto de usuarios
2. Un carácter para retirar los permisos (-), dejarlos igual (=) o añadir (+)
3. Las abreviaturas de los tipos de permisos:
  - r: lectura
  - w: escritura
  - x: ejecución

Por ejemplo, para dar permisos de escritura a todos los usuarios sobre el fichero 'prueba':

```
$ chmod o+w /home/usuario1/prueba
```

Para quitar todos los permisos de escritura del fichero a todos excepto al propietario:

```
$ chmod go-w /home/usuario1/prueba
```

El segundo formato se forma considerando cada grupo de permisos (rwx) como un número octal. De forma que el valor octal de los permisos de un grupo de permisos es:

lectura(4) + escritura(2) + ejecución (1)

Ejemplos:

rwxr-xr-x	755
rw-r--r--	644
r--r--r--	444
rwxrwxrwx	777
-----	000
rwx-----	700

De forma que la ejecución del comando chmod utilizando el formato octal es:

```
$ chmod 644 prueba
```

**touch nombre\_archivo**

Si el archivo existe actualiza la fecha de modificación del archivo y si no existe, crea un archivo nuevo de tamaño 0 bytes.

**head fichero**

Muestra por pantalla las 10 primeras líneas de un fichero pasado como parámetro. Para que muestre un número de líneas diferente, se usa la opción `-n`:

```
$ head -n 5 /etc/passwd
```

Muestra las primeras 5 líneas del fichero `/etc/passwd`

**tail fichero**

Muestra las últimas 10 líneas de un fichero pasado como parámetro. Para que muestre un número de líneas diferente, se usa la opción `-n`:

```
$ tail -n 3 /etc/passwd
```

Muestra las últimas 3 líneas del fichero `/etc/passwd`

**wc fichero**

Comando que muestra por pantalla el número de líneas, palabras y caracteres del fichero especificado.

## 3.10. Enlaces

En los sistemas Linux los archivos o directorios tienen asignado un número identificativo llamado inodo. Este número inodo representa una entrada en la tabla de inodos donde se almacena toda la información importante del archivo como propietario, grupo, permisos, tipo de archivo, fecha de última modificación, etc. Sin embargo, el nombre no se almacena en la tabla de inodos. El nombre del archivo se almacena en el directorio donde se almacena el archivo y es aquí donde se asocia el nombre del archivo a su inodo correspondiente. Esta estructura permite tener varios nombres haciendo referencia al mismo archivo. Cuando se tienen varios nombres para un archivo, a cada nombre se le llama un **enlace duro** o **enlace hard**. Esto es útil cuando se quiere tener el mismo archivo en dos directorios diferentes.

La idea importante es que en el disco existirá un solo archivo (con un número de inodo único) con varios nombres y cualquier cambio que se haga utilizando cualquiera de los nombres quedará reflejado en el archivo.

Para crear un enlace hard a un archivo se utiliza el comando `ln`:

```
$ ln archivo nombre_enlace
```

Para ver el número de inodo de un archivo:

```
$ ls -li archivo
```

Se puede comprobar si existen más enlaces hard de un fichero en el mismo directorio con el comando:

```
$ ls -li | sort | more
```

Y para encontrar todos los enlaces a un mismo fichero:

```
$ find / -inum 114032
```

Cuando utilizamos el comando `rm` sobre un enlace, se borra el enlace entre el nombre del fichero y el inodo correspondiente. El fichero se borrará sólo cuando no haya enlaces.



También se pueden crear los llamados **enlaces soft o simbólicos**. Este tipo de enlace tiene dos ventajas sobre el enlace hard: se puede crear un enlace soft a un directorio y también se puede crear sobre un archivo o directorio de otro sistema de archivos.

Un enlace soft es un pequeño archivo puntero que el listado de `ls -l` se indica con una fecha (->). (Es equivalente a los llamados accesos directos en los sistemas Windows).

Para crear un enlace soft o simbólico:

```
$ ln -s archivo nombre_enlace
```

### 3.11. Operador pipeline o tubería

Como ya hemos visto, los programas en Linux utilizan la entrada estándar como entrada de datos y sacan sus datos a través de la salida estándar. Esta característica se puede aprovechar para encadenar o enlazar la ejecución de varios comandos, de forma que la salida de un comando sea la entrada del siguiente. Para ello se utiliza el operador pipeline o tubería: |

Por ejemplo:

```
$ ls -l /etc | more
```

El comando `ls -l`, saca por la salida estándar (por defecto la pantalla) una lista de ficheros. Utilizando el operador | dirige esa salida al siguiente comando, el comando `more`. Este comando realiza una paginación de lo que le llega por la entrada estándar (que es la salida de `ls`).

```
$ ls -li | sort | more
```

### 3.12. Redirección de comandos

Los comandos y programas ejecutados en Linux normalmente no lanzan sus mensajes directamente a la pantalla. En lugar de eso, llaman a una rutina especial denominada salida estándar o `stdout`. `stdout` normalmente está configurado para dirigirse a la pantalla pero se puede dirigir fácilmente a cualquier dispositivo como una impresora o un archivo de disco. Este es uno de los elementos verdaderamente elegantes de Linux. Los programas dirigen sus salidas a `stdout` sin tener ni idea de hacia donde se dirige realmente su salida. El operador para redirigir la salida estándar a otro dispositivo es el carácter 'mayor': >

Por ejemplo, el comando `cal` muestra por pantalla un calendario del mes actual:

```
$ cal
```

pero podríamos dirigirlo hacia la impresora:

```
$ cal > /dev/lp0
```

o hacia un archivo:

```
$ cal > /home/usuario1/datos/calendario
```

También existe una rutina especial para presentar los mensajes de error. Esta salida se llama error estándar o `stderr`. Normalmente, `stderr` está dirigida a la pantalla, pero de nuevo se puede dirigir a un dispositivo o a un archivo, de forma totalmente independiente de `stdout`. El operador para realizar esta operación es `2>`

```
$ ls -j 2> /home/usuario1/error.log
```

Se dispone también de una rutina para las entradas, llamada entrada estándar o `stdin`. Si un programa pide su entrada de `stdin`, por defecto se utiliza la información que llega a través del teclado, pero esta entrada se puede suministrar desde un dispositivo o un archivo. El operador para realizar esta operación es el carácter ‘menor’: `<`

```
$ cat < /home/usuario1/prueba
```

Los operadores `>` y `2>` utilizados sobre archivos, producen una nueva copia del archivo especificado eliminando cualquier información que hubiera. Para conservar la información y que el operador añada la información de la redirección, sin sobrescribir se utilizan los operadores `>>` y `2>>`.

### 3.13. Comandos generales y de información del sistema

#### **clear**

Borra la pantalla.

#### **date**

Muestra la fecha y la hora del sistema.

#### **cal**

Muestra por pantalla un calendario. Si no se especifica nada muestra el mes actual.

El siguiente ejemplo muestra el mes de Octubre de 2004 empezando la semana por el Lunes:

```
$ cal -m 10 2004
```

```
    octubre de 2004
lu ma mi ju vi sa do
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

#### **echo**

Muestra por pantalla la línea de texto pasada como parámetro. También se puede utilizar para visualizar el contenido de variables de entorno. Ejemplo:

```
$ echo $PATH
```

Muestra por pantalla el contenido de la variable de entorno `PATH`.

#### **hostname**

Muestra por pantalla el nombre de la máquina.

**who**

Muestra por pantalla los usuarios que se encuentran conectados a la máquina.

**finger [usuario]**

Muestra por pantalla información sobre usuarios.

**whoami**

Muestra por pantalla el nombre del usuario actual.

**uptime**

Muestra por pantalla el tiempo que lleva el sistema activo. Concretamente muestra: la hora actual, cuanto tiempo lleva funcionando el sistema, cuantos usuarios están actualmente conectados, la carga media del sistema durante el último minuto, los últimos 5 minutos y los últimos 15 minutos.

Una información más extendida sobre los usuarios conectados se muestra con el comando `w`

`$ w`

```
07:34:07 up 4:51, 3 users, load average: 0.17, 0.06, 0.02
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
usuariol  tty1    -             2:46am  56.00s 6.41s  0.25s top
root      tty3    -             7:24am  0.00s  0.91s  0.09s w
usuario2  tty4    -             7:33am  40.00s 1.73s  1.54s emacs
```

**uname -a**

Imprime información del sistema. Para ver toda la información que puede obtener, utiliza la opción `-a`. Con esta opción se puede ver, por ejemplo, la versión del kernel del sistema.

`$ uname -a`

```
Linux grupo15 2.4.20-8 #1 Thu Mar 13 17:54:28 EST 2003 i686 i686 i386 GNU/Linux
```

**df**

Informa de la utilización del espacio de disco en sistemas de ficheros. Si no se especifica ningún parámetro muestra información sobre todos los sistemas de ficheros:

```
S.ficheros      1K-blocks      Used Available Use% Montado en
/dev/sda2       3834496      1456908   2182800   41% /
/dev/sda1       101089        9426     86444    10% /boot
none            62996         0        62996     0% /dev/shm
/dev/hda2       1019856      900688   119168   89% /mnt/DOS
```

**du**

Informa de la cantidad de espacio de disco usada por los ficheros especificados. Si se utiliza sin argumentos, `du` informa del espacio de disco para el directorio actual.

`$ du /lib -hs`

Muestra el tamaño del directorio `/lib` utilizando prefijos (M: mega, K:kilo)

**free**

Muestra información sobre la memoria libre y la memoria usada del sistema.

**vmstat**

Muestra información estadística sobre el uso de la memoria virtual del sistema.

### 3.14. Gestión de procesos

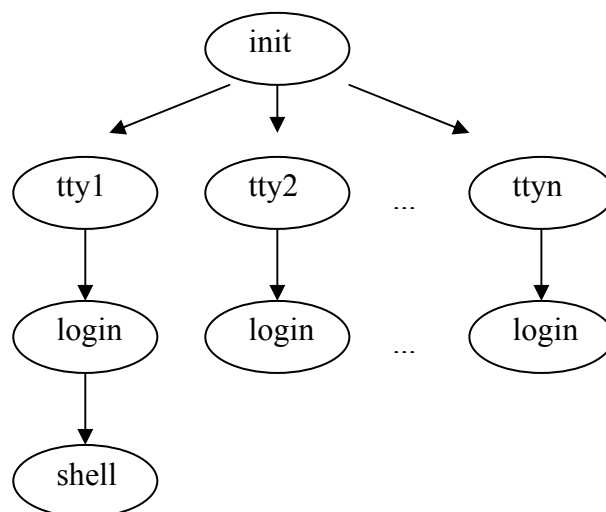
Linux es un sistema operativo multitarea, es decir, permite la ejecución de varios procesos de forma concurrente. Deben quedar claros dos conceptos importantes:

**Programa:** es un conjunto de instrucciones escrito en un lenguaje de programación determinado y almacenado en un fichero, normalmente después de su traducción al lenguaje máquina correspondiente.

**Proceso,** programa en ejecución.

En Linux, la estructura de procesos es jerárquica. Para representar la relación entre los procesos se utiliza el concepto de procesos padre-hijo. Esto implica que todo proceso debe tener un proceso llamado padre que es el que crea dicho proceso. Como en toda estructura jerárquica existe un proceso raíz, este es un proceso llamado **init**. Cada proceso que se ejecuta en el sistema se le asigna un identificador único llamado PID (process identifier). El proceso init tiene asignado el PID 1.

Un ejemplo de la estructura jerárquica de los procesos: el proceso init, entre otras cosas, crea los procesos que gestionan los terminales virtuales, llamados tty1, tty2, ... estos, a su vez, lanzan los procesos login, que son los responsables de pedir el nombre de usuario y contraseña y validar a los usuarios que utilizan los terminales virtuales. En caso de que la validación sea correcta, el proceso login ejecuta a su vez a la shell correspondiente.



A los procesos ejecutados por los usuarios se les suele denominar trabajos o jobs. Gracias a la multitarea, un usuario puede ejecutar varios trabajos de forma simultánea. Sin embargo sólo uno de ellos podrá tener la interacción directa con el usuario. A este trabajo se le llama trabajo en **primer plano (o foreground)**. El resto de trabajos pueden ejecutarse en lo que se llama **segundo plano (o background)**. Sin embargo, si un trabajo en segundo plano genera salida, ésta aparece en el terminal por lo que si se desea evitar, se puede utilizar la redirección. Si un trabajo en segundo plano es interactivo, es decir, requiere entradas en tiempo de ejecución, dicho trabajo se quedará parado o en stop.

Para lanzar un proceso en primer plano o foreground se ejecuta de forma normal:

```
$ ls -l /var/log > ls_log
```

Para lanzarlo en segundo plano o background se añade al final del comando el carácter &:

```
$ ls -l /var/log > ls_log &
$ find / -name 'samba' > busqueda 2>&1 &
```

### Comando para visualizar información sobre los procesos: ps

Este comando admite muchos parámetros y nos proporciona mucha información sobre los procesos que están ejecutándose en el sistema. Algunos de los parámetros más interesantes:

```
$ ps
```

Sin parámetros nos muestra los procesos del usuario que ejecuta desde la shell correspondiente.

```
$ ps -u [usuario]
```

Nos muestra los procesos de un usuario ejecutados desde la shell correspondiente. Muestra más información sobre cada proceso que la ejecución del comando sin parámetros.

```
$ ps -A
```

```
$ ps -e
```

Nos muestra todos los procesos en memoria.

```
$ ps -f
```

Muestra información extendida de los procesos.

```
$ ps -ef
```

Muestra información extendida de los procesos de todos los usuarios. Esta es una de las formas más frecuentes de utilizar el comando ps.

Ejemplo de salida de ps -ef:

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	09:05	?	00:00:03	init
root	2	1	0	09:05	?	00:00:00	[keventd]
root	3	1	0	09:05	?	00:00:00	[kapmd]
root	4	1	0	09:05	?	00:00:00	[ksoftirqd_CPU0]
root	9	1	0	09:05	?	00:00:00	[bdf flush]
root	5	1	0	09:05	?	00:00:00	[kswapd]
root	6	1	0	09:05	?	00:00:00	[kscand/DMA]
root	7	1	0	09:05	?	00:00:00	[kscand/Normal]
root	8	1	0	09:05	?	00:00:00	[kscand/HighMem]
root	10	1	0	09:05	?	00:00:00	[kupdated]
root	11	1	0	09:05	?	00:00:00	[mdrecoveryd]
root	19	1	0	09:05	?	00:00:00	[kjournald]
root	76	1	0	09:05	?	00:00:00	[khubd]
root	1403	1	0	09:09	?	00:00:00	[kjournald]
root	1737	1	0	09:09	?	00:00:00	syslogd -m 0
root	1741	1	0	09:09	?	00:00:00	klogd -x
rpc	1760	1	0	09:09	?	00:00:00	[portmap]
rpcuser	1779	1	0	09:09	?	00:00:00	[rpc.statd]
root	1845	1	0	09:09	?	00:00:00	/usr/sbin/apmd -p 10 -w 5 -W -P
root	1882	1	0	09:09	?	00:00:00	/usr/sbin/sshd
root	1896	1	0	09:09	?	00:00:00	xinetd -stayalive -reuse -pidfil
root	1916	1	0	09:09	?	00:00:00	[sendmail]
smmsp	1925	1	0	09:09	?	00:00:00	[sendmail]
root	1935	1	0	09:09	?	00:00:00	gpm -t ps/2 -m /dev/mouse

```

root      1944      1  0 09:09 ?          00:00:00 crond
root      1955      1  0 09:09 ?          00:00:00 cupsd
xfs       2013      1  0 09:09 ?          00:00:00 [xfs]
root      2022      1  0 09:09 ?          00:00:00 anacron -s
daemon   2031      1  0 09:09 ?          00:00:00 [atd]
root      2039      1  0 09:09 ?          00:00:00 login -- root
root      2040      1  0 09:09 ?          00:00:00 [login]
root      2041      1  0 09:09 ?          00:00:00 [login]
root      2042      1  0 09:09 tty4       00:00:00 /sbin/mingetty tty4
root      2043      1  0 09:09 tty5       00:00:00 /sbin/mingetty tty5
root      2044      1  0 09:09 tty6       00:00:00 /sbin/mingetty tty6
root      2045     2039  0 09:22 tty1       00:00:00 -bash
usuario1  2094     2040  0 09:24 tty2       00:00:00 -bash
usuario2  2134     2041  0 09:24 tty3       00:00:00 -bash
usuario2  2175     2134  0 09:24 tty3       00:00:00 sleep 300
usuario1  2176     2094 14 09:24 tty2       00:00:00 find / -name samba
root      2177     2045  0 09:24 tty1       00:00:00 ps -ef

```

### Comando pstree

Muestra la estructura jerárquica de los procesos en memoria.

### Comando kill

Comando para terminar un proceso en memoria. Se puede utilizar también para enviar señales a un proceso, de hecho, por defecto se envía la señal TERM(15) que es la señal para indicar al proceso que debe finalizar su ejecución.

```

$ kill [-s señal] PID
$ kill -señal PID

```

Se puede obtener la lista de posibles señales con `kill -l`

Si no se especifica el parámetro señal, se envía la señal TERM. Para forzar la finalización de un proceso se puede utilizar la señal KILL (9):

```

$ kill -9 12435

```

Para obtener el PID del proceso que queremos finalizar, utilizar el comando `ps`.

### Comando jobs

Muestra los trabajos en ejecución de un usuario. Cada trabajo en ejecución asociado a un usuario tiene un número de trabajo asignado diferente al PID, este número es el jobid (job identifier)

### Comando: bg jobid

Pasa a segundo plano o background un trabajo en ejecución. Para realizar esta operación normalmente primero hay que parar o ‘congelar’ el proceso para que nos devuelva el control al shell y poder ejecutar el comando. Para ello pulsar la combinación de teclas Ctrl+Z.

### Comando: fg jobid

Devuelve un proceso a primer plano.

### Comando top

Comando que muestra información en tiempo real de la actividad del procesador así como de los procesos que más recursos están consumiendo.

Ejemplo de ejecución de top:

```

09:32:54 up 27 min,  3 users,  load average: 0,08, 0,12, 0,25
41 processes: 38 sleeping, 3 running, 0 zombie, 0 stopped
CPU states:  2,0% user  2,7% system  0,0% nice  0,0% iowait  95,2% idle
Mem:  125992k av,  121152k used,  4840k free,  0k shrd,  36256k buff
      55612k actv,  4k in_d,  252k in_c
Swap: 192772k av,  388k used, 192384k free  22648k cached
  PID USER      PRI  NI  SIZE  RSS  SHARE STAT  %CPU  %MEM   TIME CPU COMMAND
 2236 usuario1  23   0  1248 1248   808 R    47,2  0,9   0:03  0  ls
 2237 usuario2  25   0   768  768   640 R     4,7  0,6   0:01  0  find
    5 root      15   0    0    0    0  SW    2,8  0,0   0:00  0  kswapd
 2257 root      20   0  1016 1016   836 R     1,8  0,8   0:00  0  top
 1403 root      15   0    0    0    0  SW    0,9  0,0   0:00  0  kjournald
    1 root      15   0    92   68   40  S    0,0  0,0   0:03  0  init
    2 root      15   0    0    0    0  SW    0,0  0,0   0:00  0  keventd
    3 root      15   0    0    0    0  SW    0,0  0,0   0:00  0  kapmd
    4 root      34  19    0    0    0  SWN   0,0  0,0   0:00  0  ksoftirqd_CPU
    9 root      15   0    0    0    0  SW    0,0  0,0   0:00  0  bdflush
    6 root      15   0    0    0    0  SW    0,0  0,0   0:00  0  kscand/DMA
    7 root      15   0    0    0    0  SW    0,0  0,0   0:00  0  kscand/Normal
    8 root      15   0    0    0    0  SW    0,0  0,0   0:00  0  kscand/HighMe
   10 root      15   0    0    0    0  SW    0,0  0,0   0:00  0  kupdated
   11 root      25   0    0    0    0  SW    0,0  0,0   0:00  0  mdrecoveryd
   19 root      15   0    0    0    0  SW    0,0  0,0   0:00  0  kjournald
   76 root      25   0    0    0    0  SW    0,0  0,0   0:00  0  khubd

```

### 3.15. Otros comandos

#### **dmesg**

Muestra por pantalla todos los mensajes producidos por el kernel del sistema en el arranque.

#### **grep PATRÓN [fichero...]**

Busca la cadena de texto o patrón especificado en los ficheros pasados como argumento (o en la entrada estándar si no se especifican ficheros). Muestra por pantalla las líneas que contengan dicho patrón.

#### **sort [fichero...]**

Ordena las líneas del fichero (o ficheros) de texto especificado como parámetro.

#### **cut opciones [fichero...]**

Corta y conserva las columnas o campos especificados en las opciones.

#### **uniq [fichero...]**

Elimina las líneas duplicadas de un fichero (o ficheros) de texto.

#### **sleep n**

Proceso que simplemente se ejecuta durante n segundos pero no realiza ninguna acción.

#### **time comando**

Ejecuta el comando especificado e informa del tiempo empleado en la ejecución.

### 3.16. El editor en modo comando universal: vi

Uno de los tipos de archivos más importantes en los sistemas Linux (y en UNIX en general) son los archivos de texto plano. Uno de los principales motivos de ello es que toda la administración del sistema, con pocas excepciones, se puede realizar editando archivos de texto que personalizan el sistema y controlan su forma de operar.

Para poder editar archivos de texto Linux ofrece, entre otros, el editor vi. Este editor tiene un manejo difícil ya que todas las operaciones que se pueden realizar se llevan a cabo mediante comandos, sin embargo, gracias precisamente a la gran cantidad de comandos que se pueden utilizar, lo hace también bastante potente.

En cualquier caso, la verdadera razón de por qué existe un apartado dedicado a vi en prácticamente todos los manuales de Linux e UNIX es precisamente su carácter universal. Este editor está presente en todas las implementaciones de UNIX, incluido por supuesto Linux. Además su uso es bastante fiable, "*Finalmente, si no funciona nada, vi aún funciona*".

La forma de invocarlo es:

```
$ vi archivo
```

La mayoría de los comandos se usan pulsando las teclas indicadas sin que éstas aparezcan en la pantalla ni tampoco es necesario pulsar <ENTER> al final de ellos, sólo los comandos que comienzan con `:/` y `?` son mostrados en la última línea de la pantalla y requieren la pulsación de <ENTER> para finalizar (estos corresponden a los comandos del editor `ex`, en el cual se basa `vi`).

Antes de comenzar a describir los comandos se establecerán las normas de la notación:

<i>c</i>	: Un caracter cualquiera
<i>l</i>	: Una letra del alfabeto inglés
<i>^X</i>	: Pulsar las teclas <CONTROL> y X
<i>caracter</i>	: Un caracter cualquiera
<i>CHARACTER</i>	: Un caracter distinto de espacio
<i>palabra</i>	: Una secuencia de letras y/o números
<i>PALABRA</i>	: Una secuencia de caracteres incluyendo los espacios que siguen
<i>arch</i>	: Algún archivo del disco (Existente o no)
<i>patrón</i>	: Secuencia de caracteres a utilizar en un patrón de búsqueda
<i>movimiento</i>	: Algún comando de movimiento

A continuación se describirán la mayoría de los comandos de vi (sólo se excluyen los más complicados), a muchos de éstos se les puede anteponer un número decimal que indica un factor de repetición del comando, es decir, si se escribe 20 antes del comando, éste se repite 20 veces. Los comandos que tienen esta capacidad serán señalados con una letra *n* en la columna izquierda, al no colocar el valor, el comando se ejecuta sólo una vez.

#### Comandos de Movimiento



<b>N Comando</b>	<b>Descripción</b>
-- ^F	Avanza una página
-- ^B	Retrocede una página
-- ^D	Avanza media página
-- ^U	Retrocede media página
-- ^U	Retrocede media página
<i>n</i> \$	Avanza hasta el final de <i>n</i> -1 líneas adelante
-- ^	Va al primer caracter distinto de espacio de la línea
<i>n</i> _	Va al primer caracter distinto de espacio de <i>n</i> -1 líneas adelante
<i>n</i> -	Va al primer caracter distinto de espacio <i>n</i> líneas atrás
<i>n</i> + ó <ENTER>	Va al primer caracter distinto de espacio <i>n</i> líneas adelante
-- 0 (Cero)	Va al primer caracter de la línea (incluso espacio)
<i>n</i>	Va a la columna <i>n</i> dentro de la línea
<i>n</i> fc	Avanza hasta el caracter <i>c</i>
<i>n</i> tc	Avanza hasta la posición anterior al caracter <i>c</i>
<i>n</i> Fc	Retrocede hasta el caracter <i>c</i>
<i>n</i> Tc	Retrocede hasta la posición siguiente al caracter <i>c</i>
<i>n</i> ;	Repite el último comando 'f', 't', 'F' o 'T'
<i>n</i> ,	Idéntico al anterior, pero en la dirección opuesta
<i>n</i> w	Avanza <i>n</i> palabras
<i>n</i> W	Avanza <i>n</i> PALABRAS
<i>n</i> B	Retrocede <i>n</i> palabras
<i>n</i> B	Retrocede <i>n</i> PALABRAS
<i>n</i> E	Avanza al final de <i>n</i> palabras adelante
<i>n</i> E	Avanza al final de <i>n</i> PALABRAS adelante
<i>n</i> G	Va a la línea <i>n</i> (última si <i>n</i> no se especifica)
<i>n</i> H	Va a la línea <i>n</i> a partir de la primera que se ve en pantalla
<i>n</i> L	Va a <i>n</i> -ésima línea anterior a la última que se ve en pantalla
-- M	Va a la línea del medio de la pantalla
<i>n</i> )	Avanza <i>n</i> sentencias
<i>n</i> (	Retrocede <i>n</i> sentencias
<i>n</i> {	Avanza <i>n</i> párrafos
<i>n</i> }	Retrocede <i>n</i> párrafos
-- \l	Va a la marca <i>l</i>

---

-- 'l	Va al primer CHARACTER dentro de la línea con la marca <i>l</i>
-- ``	Va a la posición anterior al último salto
-- "	Va al primer CHARACTER dentro de la línea en que se encontraba el cursor antes del último salto
-- /patrón	Avanza hasta la siguiente ocurrencia del <i>patrón</i>
-- ?patrón	Retrocede a la anterior ocurrencia del <i>patrón</i>
-- N	Repite el último comando '/' o '?'
-- N	Igual al anterior, pero en la dirección opuesta
-- %	Busca el siguiente paréntesis o su pareja (también con {, }, [, y ])

### Comandos de Inserción de texto

<i>N</i> Comando	Descripción
<i>n</i> <i>itexto</i> <ESC>	Inserta <i>texto</i> en la posición actual del cursor
<i>n</i> <i>atexto</i> <ESC>	Agrega <i>texto</i> en la posición siguiente a la del cursor
<i>n</i> <i>Itexto</i> <ESC>	Inserta <i>texto</i> delante del primer CHARACTER de la línea actual
<i>n</i> <i>Atexto</i> <ESC>	Agrega <i>texto</i> al final de la línea actual
<i>n</i> <i>otexto</i> <ESC>	Agrega <i>texto</i> en la línea siguiente
<i>n</i> <i>Otexto</i> <ESC>	Agrega <i>texto</i> en la línea anterior
<i>n</i> P	Coloca el último grupo de líneas guardado o borrado en la línea siguiente <i>n</i> veces
<i>n</i> P	Coloca el último grupo de líneas guardado o borrado en la línea anterior <i>n</i> veces
<i>n</i> .	Repite el último comando <i>n</i> veces

### Comandos de Reemplazo

<i>n</i> Comando	Descripción
<i>n</i> rc	Reemplaza <i>n</i> caracteres por <i>c</i>
<i>n</i> <i>Rtexto</i> <ESC>	Sobreescribe el resto de la línea, agregando <i>n-1</i> veces
<i>n</i> s	Sustituye <i>n</i> caracteres
<i>n</i> S	Sustituye <i>n</i> líneas
<i>n</i> <i>cmovimiento</i> <i>texto</i> <ESC>	Cambia lo alcanzado por <i>n</i> movimientos por <i>texto</i>
<i>n</i> <i>cctexto</i> <ESC>	Cambia <i>n</i> líneas por <i>texto</i>
<i>n</i> <i>Ctexto</i> <ESC>	Cambia el resto de la línea las y <i>n-1</i> líneas siguientes por <i>texto</i>
-- ~	Intercambia entre mayúsculas y minúsculas
<i>n</i> J	Junta <i>n</i> líneas (Si <i>n</i> no se especifica se junta la actual con la siguiente)

<i>n</i> .	Repite el último comando <i>n</i> veces (J sólo una vez)
-- :[ <i>x,y</i> ]s/ <i>patrón</i> / <i>texto</i> / <i>m</i>	Sustituye el texto alcanzado por <i>patrón</i> por <i>texto</i> entre las líneas <i>x</i> e <i>y</i> (% para señalar todas). El modificador <i>m</i> puede ser g (Global) o c (Con confirmación)
-- &	Repite el último reemplazo dado con el comando anterior

### Comandos de Borrado

#### ***N* Comando Descripción**

<i>N</i> x	Borra <i>n</i> caracteres a partir de la posición del cursor
<i>N</i> X	Borra <i>n</i> caracteres antes del cursor
<i>N</i> <i>dmovimiento</i> <	Borra <i>n</i> veces lo indicado por <i>movimiento</i> (3dw' Borra 3 palabras)
<i>N</i> dd	Borra <i>n</i> líneas
-- D	Borra hasta el final de la línea
<i>N</i> .	Repite el último comando <i>n</i> veces

### Comandos de Copia y Marcado

#### ***n* Comando Descripción**

<i>n</i> <i>ymovimiento</i>	Marca el texto descrito por <i>movimiento</i> para copiarlo con el comando p o P
<i>n</i> yy	Marca <i>n</i> líneas para copiarlas con el comando p o P
<i>n</i> Y	Marca <i>n</i> líneas para copiarlas con el comando p o P
-- ml	Marca la posición del cursor con la letra <i>l</i>

### Comandos para deshacer

#### **Comando Descripción**

U	Deshace la última modificación
U	Deshace todos los cambios hechos en la línea actual
P	Coloca el último grupo de líneas guardado o borrado en la línea siguiente
P	Coloca el último grupo de líneas guardado o borrado en la línea anterior
:q!	Abandona vi sin grabar las modificaciones
:e!	Re-edita el archivo (Como salir y editarlo nuevamente)

### Comandos de Grabación y Salida

#### **Comando Descripción**

:q	Sale de vi (Si no se ha modificado desde la última grabación)
:q!	Sale sin grabar
:w	Graba el archivo
:w <i>arch</i>	Graba en el archivo <i>arch</i>
:w >> <i>arch</i>	Agrega el archivo editado al archivo de nombre <i>arch</i>

:w! *Arch* Graba el archivo editado con nombre *arch* sin importar que éste exista  
 :x,y w *arch* Graba de la línea *x* a la *y* en el archivo *arch*  
 :wq Graba y sale  
 :ZZ Graba slo si el archivo ha sido modificado y sale  
 :f *arch* Cambia el nombre del archivo editado a *arch*  
 :r *arch* Agrega el archivo *arch* después de la línea actual

<b>Pasar a modo de entrada</b>		<b>Mover el cursor</b>	
i	Antes del cursor	l o SpaceBar	Un espacio a la derecha
a	Después del cursor	h o BackSpace	Un espacio a la izquierda
I	A principio de línea	j o +	Una línea abajo
A	A final de línea	k o -	Una línea arriba
O	Abre una línea por encima	\$	Al final de línea
O	Abre línea por debajo	^ o 0	A principio de línea
		w	A la siguiente palabra
		e	Al final de la palabra
		b	A la palabra anterior
		nG	A la línea n
<b>Borrar</b>		<b>Cambiar</b>	
dw	Borrar palabra	cw	Cambiar palabra
dd	Borrar línea	cc	Cambiar línea
d	hasta final de línea	C	Hasta final de línea
	El caracter del cursor	r	El caracter del cursor
		R	Hasta ESC
<b>Otras funciones</b>		<b>Control de pantalla</b>	
u	Deshacer	CTRL-D	Deslizar hacia adelante
/	Buscar hacia adelante	CTRL-U	Deslizar hacia atrás
?	Buscar hacia atrás	CTRL-F	Siguiente pantalla
n	Siguiente ocurrencia	CTRL-B	Pantalla anterior
.	Repetir la última acción	CTRL-L	Redibujar pantalla
Y o yy	Copiar línea		
P	Pegar por debajo	<b>Modo de última línea</b>	
P	Pegar por encima	:w	Escribir a archivo
ZZ	Escribir y abandonar	:q	Terminar
ESC	Cancelar una orden	:wq	Guardar y teminar
"	Marcar	:n	Siguiente archivo
		:r <i>archivo</i>	Leer de <i>archivo</i>
<b>En modo de entrada</b>		:e	Editar archivo
BackSpace	Borra caracter	:f o CTRL-G	Nombre de archivo

CTRL-w	Borrar palabra		:set	Modificar opciones
ESC	A modo de orden		:!	Escape al shell
			:n	Línea n

### 3.17. Las herramientas mtools

Las mtools son un conjunto de comandos que permiten manipular ficheros en formato DOS bajo sistemas Linux. Normalmente, los nombres de los comandos provistos por mtools empiezan por m y contienen el nombre del comando original en DOS. Para ver la lista de comandos soportados ejecutar el comando mtools.

Cuando utilizamos estos comandos no es necesario realizar las operaciones de montaje y desmontaje de dispositivos.

<b><u>mtools</u></b>	<b><u>DOS</u></b>
mattrib	attrib
mcd	cd
mcopy	copy
mdel	del
mdeltree	deltree
mdir	dir
mformat	format
mmd	md
mrd	rd
mmove	move
mren	ren
mtype	type

Ejemplo:

```
$ mcopy /home/usuario1/fichero.txt A:
$ mdir A:
```

# 4. ADMINISTRACIÓN BÁSICA

## 4.1. Introducción

Normalmente, toda la administración del sistema se lleva a cabo a través del usuario root o administrador. Existe un máxima en el mundo Unix en general que aconseja utilizar el usuario root el tiempo mínimo imprescindible.

El usuario root puede leer, modificar o borrar cualquier fichero en el sistema, cambiar permisos y ejecutar programas especiales, como pueden ser los que particionan un disco o crean sistemas de ficheros. Con esta libertad es fácil cometer errores que tengan consecuencias importantes, incluso catastróficas cuando hablamos de máquinas utilizadas como servidores.

## 4.2. Arranque del sistema

Lo primero que ocurre en un sistema Linux es la carga del kernel o núcleo del sistema operativo. El kernel toma el mando de todo el hardware del sistema, comprueba y fija la consola e inicializa las interfaces elementales de la placa base. En los siguientes pasos, los distintos drivers que forman parte del núcleo realizan un chequeo del hardware presente para realizar una inicialización en caso necesario. Después del chequeo de la partición y la carga del sistema de ficheros raíz, el kernel ejecuta el programa `/sbin/init`, el cual realiza el auténtico arranque del sistema con sus múltiples programas auxiliares y sus configuraciones. El kernel sigue gestionando el sistema completo y los accesos al hardware.

El proceso `init` es el proceso que se ejecuta para una correcta inicialización del sistema. Se puede decir que es el “padre de todos los procesos” del sistema. Dentro de todos los programas, `init` tiene un status especial: `init` es ejecutado directamente por el kernel y no tiene efecto la señal 9, con la cual todos los procesos pueden ser interrumpidos. Los procesos siguientes son ejecutados directamente por `init` o por uno de sus procesos “hijo”.

`init` se configura de forma centralizada a través del fichero `/etc/inittab`; aquí se definen los llamados niveles de ejecución y se determina qué servicios y demonios deben estar disponibles en los diferentes niveles. Dependiendo de la configuración en `/etc/inittab`, `init` ejecuta diferentes scripts, que por razones de organización se reúnen en el directorio `/etc/init.d`. Por lo tanto, todo el proceso de arranque es controlado por el proceso `init`.

En Linux existen varios niveles de ejecución que definen el estado del sistema. A continuación se muestran los niveles disponibles:

Nivel de ejecución	Significado
0	Parada del sistema
1	Modo monousuario
2	Modo multiusuario local sin red
3	Modo multiusuario completo con red
4	No usado
5	Modo multiusuario completo con red y entorno gráfico
6	Reinicio del sistema

Se puede elegir el nivel de ejecución en el arranque del sistema en el fichero `/etc/inittab`, en la línea:  
`id:3:initdefault:`

Se puede cambiar el nivel de ejecución ejecutando el comando `init` seguido del número de nivel:  
`# init 5`

Por ejemplo, para tareas de administración, se puede pasar al modo monousuario:  
`# init 1`

En este nivel no se puede conectar ningún usuario. Después de acabar los trabajos de administración se puede volver al modo multiusuario:  
`# init 3`

Para arrancar en el nivel 5, tiene que asegurarse de que el sistema X Window ya está correctamente configurado.

### 4.3. Gestión de usuarios

Durante la instalación del sistema, se crea la cuenta del usuario `root`, para el administrador del sistema. Además, se crean una serie de usuarios especiales relacionados con servicios del sistema, tales como `bin`, `ftp`, `wheel`, etc.

El usuario `root` tiene la posibilidad de crear cuentas para el resto de usuarios del sistema. La creación de una cuenta de usuario consiste en añadir una línea al fichero de configuración `/etc/passwd`, con la información sobre el usuario. Además, se creará una línea en el fichero `/etc/shadow` para almacenar la contraseña del usuario.

El formato de una línea del fichero `/etc/passwd` es el siguiente:

```
usuario:x:ID:GID:descripción:directorio_home:shell
```

Por ejemplo:

```
usuariol:x:500:500:usuario de prueba:/home/usuariol:/bin/bash
```

Como se puede observar la información se compone de varios campos separados por el carácter :

1. nombre del usuario, es el nombre que deberá teclear en el login al sistema.
2. campo reservado a la contraseña. En versiones antiguas de UNIX, se almacenaba aquí la contraseña encriptada. Sin embargo en versiones más recientes, en este campo se escribe una `x` indicando que la contraseña se almacena en el fichero `/etc/shadow`. Podemos deshabilitar temporalmente el acceso a un usuario escribiendo un `*` en este campo.
3. ID es el identificador de usuario. Debe ser único en el sistema.
4. GID es el identificador del grupo al que pertenece el usuario por defecto.
5. texto descriptivo del usuario, en algunos sistemas este campo se utiliza para incluir el nombre completo del usuario.
6. directorio home del usuario
7. nombre de la shell por defecto para el usuario cuando acceda al sistema.

El fichero de configuración donde se almacenan los grupos del sistema es `/etc/group`

El formato de este fichero es el siguiente:

```
nombre:x:GID:usuarios
```

1. El campo nombre es el nombre del grupo
2. El campo x, en principio no se utiliza.
3. GID es el identificador del grupo
4. Usuarios. Aquí se especifican los usuarios que pertenecen a un grupo separados por comas(,).

La gestión de usuarios se puede realizar de forma “manual” modificando el fichero `/etc/passwd` o bien utilizar comandos que proporciona el sistema

#### **Creación de usuarios: useradd /adduser**

La ejecución de este comando lleva a cabo todos los cambios necesarios en los ficheros `/etc/passwd` y `/etc/shadow` para dar de alta un usuario. Además crea el directorio home con los permisos correspondientes.

El comando `useradd` admite multitud de parámetros para configurar toda la información del usuario:

```
-c descripción del usuario  
-d directorio home  
-e fecha de caducidad de la cuenta  
-g grupo por defecto  
-G otros grupos  
-p contraseña  
-s shell  
-u identificador del usuario (ID)
```

Se recomienda ejecutar el comando `useradd` para crear el usuario y el comando `passwd` para asignarle una contraseña:

```
# useradd -d /home/usuario2 -g usuarios -s /bin/bash usuario2  
# passwd usuario2
```

#### **Eliminación de usuarios: userdel**

Igual que el comando anterior realiza todos los cambios en los ficheros de configuración para eliminar una cuenta de usuario.

Además, hay que tener en cuenta qué hacer con el directorio home del usuario que queremos dar de baja. Si queremos eliminarlo también podemos utilizar la opción `-r` del comando `userdel`:

```
# userdel -r usuario2
```

#### **Cambio del propietario de un fichero o directorio: chown**

Formato del comando: `chown nuevo_prop fichero...`

```
# chown usuario2 prueba
```

#### **Cambio del grupo de un fichero o directorio: chgrp**

Formato del comando: `chgrp grupo fichero...`

Con la opción `-R`, el comando se ejecuta de forma recursiva, es decir, cambia de grupo a todos los ficheros y directorios del directorio especificado como parámetro.



```
# chgrp -R usuario2 /home/usuario3
```

### Cambio de la máscara de creación de ficheros: umask

Con este comando se puede cambiar la máscara de permisos para la creación de ficheros. Se aplica la regla del formato octal para construir la máscara teniendo en cuenta que los valores que se consideran inhabilitan el permiso correspondiente y el atributo de ejecución no tiene efecto. Sin parámetros se muestra por pantalla el estado actual de la máscara.

```
$ umask 022      ficheros creados con los permisos rw-r--r--
$ umask 002      ficheros creados con los permisos rw-rw-r--
$ umask 222      ficheros creados con los permisos r--r--r--
```

Este comando se ejecuta en uno de los ficheros de configuración de arranque de usuarios: **/etc/bashrc**

## 4.4. Ficheros de configuración de la shell bash de usuario

La shell bash utiliza una serie de ficheros de inicialización, ejecutados en el siguiente orden:

1. **/etc/profile**
2. **/etc/.bashrc**
3. **\$HOME/.bashrc**
4. **\$HOME/.bash\_profile**

Normalmente los ficheros **\$HOME/.bash\_profile** y **\$HOME/.bashrc** no existen. Si un usuario quiere utilizarlos para personalizar el arranque de la shell bash, el procedimiento normal es copiarlos de **/etc/skel/.bash\_profile** y **/etc/skel/.bashrc** a su directorio home y después añadir los cambios que desee para su personalización:

```
$ cp /etc/skel/.bash_profile $HOME
$ cp /etc/skel/.bashrc $HOME
```

Otros ficheros de la shell bash:

**\$HOME/.bash\_logout**

Fichero que se ejecuta al finalizar la sesión de usuario.

**\$HOME/.bash\_history**

Fichero que almacena el histórico de comandos del usuario. Se puede visualizar tecleando directamente el comando **history**.

La configuración sobre los colores que utiliza la shell para los nombres de fichero y directorios se personalizan en el fichero **/etc/DIR\_COLORS**

## 4.5. Ficheros de dispositivos

Una de las características especiales de Linux es su forma de acceder o reconocer los dispositivos físicos para poder hacer uso de ellos. Para Linux, cualquier dispositivo físico depende directamente de los llamados manejadores de dispositivos, que se integran en la estructura de ficheros del sistema dentro del directorio **/dev**. Todos los ficheros que contiene este directorio realmente son

manejadores de dispositivos. Los nombre de estos “ficheros” siguen un estándar para identificarlos con los dispositivos físicos:

**Disqueteras:**

```
/dev/fd0  
/dev/fd1
```

**Unidad CDROM:**

```
/dev/cdrom
```

**Dispositivos IDE:**

```
/dev/hda    Dispositivo 1  
    /dev/hda1  Primera partición primaria  
    /dev/hda2  Segunda partición primaria  
    /dev/hda3  Tercera partición primaria  
    /dev/hda4  Cuarta partición primaria  
    /dev/hda5  Particiones lógicas  
    /dev/hda6  
    ...  
/dev/hdb    Dispositivo 2  
    /dev/hdb1  Primera partición primaria  
    ...  
/dev/hdc    Dispositivo 3  
  
/dev/hdd    Dispositivo 4
```

**Dispositivos SCSI**

```
/dev/sda    Dispositivo 1  
    /dev/sda1  Primera partición primaria  
    ...  
...  
/dev/sdn    Dispositivo n
```

**Puertos serie**

```
/dev/ttyS0  
/dev/ttyS1  
/dev/ttyS2  
/dev/ttyS3
```

**Impresoras por puerto paralelo**

```
/dev/lp0  
/dev/lp1
```

## 4.6. Montar y desmontar dispositivos

Para poder acceder a un sistema de ficheros ubicado en alguno de los dispositivos que formen parte del sistema es necesario realizar una operación llamada montaje. Montar un sistema de ficheros/dispositivo no es más que hacerlo disponible en el árbol de directorios de nuestro sistema. Recordar que el árbol, de directorios de un sistema Linux es único.

En definitiva montar un dispositivo es asociar un directorio del árbol de directorios al sistema de ficheros del dispositivo. Esta operación se realiza con el comando mount:

```
mount -t <sistema_de_ficheros> <dispositivo> <punto_de_montaje>
```

Mediante el parámetro `-t` se especifica el tipo del sistema de ficheros. Los tipos más usuales son:

<code>ext2</code>	nativo de linux (el más común)
<code>ext3</code>	nativo de linux
<code>reiserfx</code>	nativo de linux
<code>msdos</code>	MSDOS
<code>vfat</code>	FAT32
<code>ntfs</code>	NTFS
<code>iso9660</code>	CDROM
<code>auto</code>	Si es posible detecta el tipo de sistema de archivos.

El parámetro `<dispositivo>` es el nombre del dispositivo, utilizando el manejador, por ejemplo `/dev/hda1`, `/dev/fd0`, `/dev/cdrom`

El parámetro `<punto_de_montaje>` es el nombre del directorio donde queremos enlazar la estructura de directorios del dispositivo.

El directorio utilizado como raíz, para montar los diferentes dispositivos removibles del sistema como disquetera y CDROM es `/mnt`

Ejemplos:

```
# mount -t vfat /dev/fd0 /mnt/floppy
```

Monta un disquete en el directorio `/mnt/floppy`. Después de ejecutar el comando se podrá acceder a la información del disquete accediendo a este directorio.

```
# mount -t iso9660 /dev/cdrom /mnt/cdrom
```

Monta un CDROM

```
# mount -t ntfs /dev/hda1 /mnt/windowsxp
```

Monta una partición NTFS de Windows XP ubicada en la primera partición de un disco IDE.

El montaje de las particiones linux que contienen toda la estructura de directorios del sistema se monta en el arranque.

Para desmontar un dispositivo se utiliza el comando `umount`:

```
# umount <directorio | dispositivo>
```

```
# umount /mnt/floppy
```

```
# umount /dev/cdrom
```

```
# umount /dev/hda1
```

Existe un fichero de configuración donde se puede especificar los sistemas de ficheros que existen en nuestro sistema y de que forma se va a realizar el montaje, el fichero `/etc/fstab`

Este fichero está compuesto por líneas con el siguiente formato:

**<dispositivo> <punto\_de\_montaje> <tipo\_sistema\_ficheros> <opciones>**

<code>LABEL=/</code>	<code>/</code>	<code>ext3</code>	<code>defaults</code>	<code>1 1</code>
<code>LABEL=/boot</code>	<code>/boot</code>	<code>ext3</code>	<code>defaults</code>	<code>1 2</code>
<code>none</code>	<code>/dev/pts</code>	<code>devpts</code>	<code>gid=5,mode=620</code>	<code>0 0</code>
<code>none</code>	<code>/proc</code>	<code>proc</code>	<code>defaults</code>	<code>0 0</code>
<code>none</code>	<code>/dev/shm</code>	<code>tmpfs</code>	<code>defaults</code>	<code>0 0</code>
<code>/dev/sda3</code>	<code>swap</code>	<code>swap</code>	<code>defaults</code>	<code>0 0</code>
<code>/dev/cdrom</code>	<code>/mnt/cdrom</code>	<code>udf,iso9660</code>	<code>noauto,owner,kudzu,ro</code>	<code>0 0</code>
<code>/dev/fd0</code>	<code>/mnt/floppy</code>	<code>auto</code>	<code>noauto,owner,kudzu</code>	<code>0 0</code>
<code>/dev/sdb1</code>	<code>/mnt/discousb</code>	<code>auto</code>	<code>noauto,owner,kedzu</code>	<code>0 0</code>

Las opciones más interesantes son:

- **user / nouser**: permite / no permite a un usuario ordinario montar el sistema de ficheros.
- **auto / noauto**: indica si / no se monta el dispositivo al ejecutar el comando `mount -a` (comando ejecutado de forma automática en el arranque del sistema)
- **defaults**: aplica las opciones `rw, suid, dev, exec, auto, nouser, async`
- **ro / rw**: montar sólo lectura o lectura-escritura

Se pueden ver los dispositivos actualmente montados en el fichero `/etc/mtab`

La ventaja de especificar un dispositivo en el fichero `/etc/fstab` es que para montar dicho dispositivo sólo hay que ejecutar el comando `mount`:

```
# mount /mnt/cdrom
# mount /mnt/floppy
```

En el caso de dispositivos extraíbles como CDROM y disquetes, es importante desmontar el dispositivo antes de extraerlo de las unidades lectoras correspondiente.

## 4.7. Copias de seguridad: tar, gzip y gunzip

En los sistemas Linux actuales el método más utilizado para realizar copias de seguridad de los datos del sistema o, en general, de cualquier parte del mismo es utilizando herramientas de empaquetado y compresión.

El comando `tar` es un empaquetador, es decir, agrupa un conjunto de ficheros y directorios en un único fichero, preservando las estructuras de directorios y permisos de los mismos.

Operaciones básicas con `tar`:

Empaquetado: `$ tar -cvf fichero.tar fichero...`

Desempaquetado: `$ tar -xvf fichero.tar`

Ver contenido de fichero `tar`: `$ tar -tvf fichero.tar`

El segundo paso después de empaquetar los ficheros es comprimirlos, para ello se utiliza el comando `gzip`.

```
$ gzip prueba
```

Este comando comprime el fichero `prueba` y crea el fichero `prueba.gz` comprimido.

Si queremos descomprimir, se utiliza el comando `gunzip`:

```
$ gunzip fichero.gz
```

Por tanto, para comprimir ficheros `tar`:

```
$ gzip fichero.tar      Se genera el fichero comprimido fichero.tar.gz
```

Por último, se puede utilizar la opción `-z` del comando `tar` para hacer el empaquetado y la compresión en un solo paso:

Compresión: `$ tar -cvzf fichero.tar.gz ficheros...`

Descompresión: `$ tar -xvzf fichero.tar.gz`

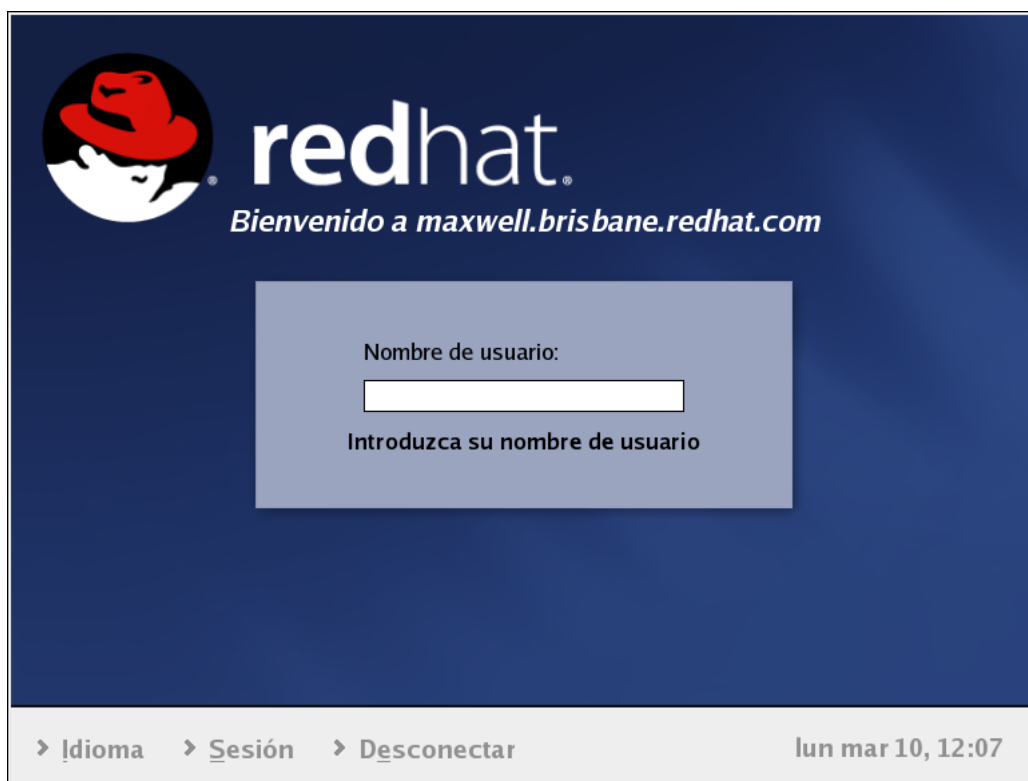
# 5. MODO GRÁFICO

## 5.1. INTRODUCCIÓN

### 5.1.1. Pantalla gráfica de conexión

Al instalar Red Hat Linux se da la oportunidad de instalar un entorno gráfico y de configurar el sistema para que, por defecto, arranque este entorno gráfico. Ya hemos visto que en las distribuciones de Linux se pueden encontrar por lo general varios gestores de ventanas. Los más extendidos y completos son GNOME y KDE. En el caso de la distribución Red Hat, que es la que vamos a utilizar en esta documentación, se utiliza por defecto el gestor GNOME.

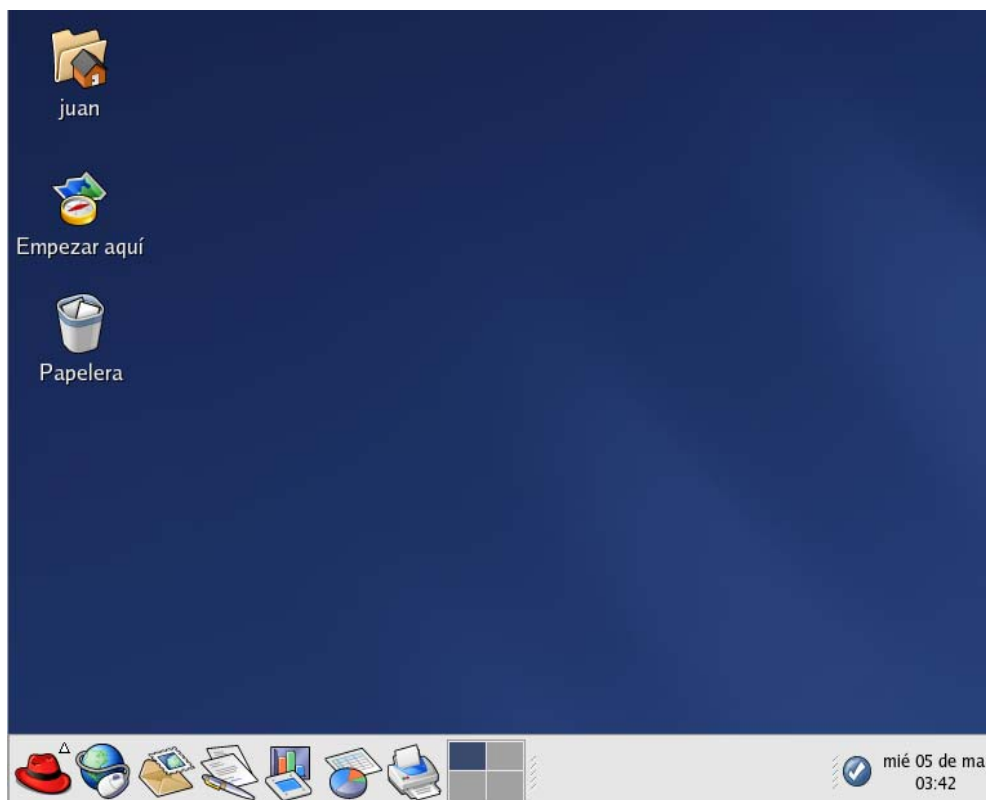
Cuando el sistema haya arrancado, aparecerá una pantalla gráfica de conexión como se muestra en la Figura 5-1. Una vez más, a menos que se haya seleccionado darle un nombre a la máquina, lo cual es usado principalmente en una configuración de red, la máquina tendrá el nombre de *localhost*.



**Figura 5-1. La pantalla gráfica de conexión de GNOME**

Para conectarse como un usuario desde la pantalla gráfica de conexión, hay que teclear el nombre de usuario, pulsar Enter y luego escribir la contraseña del usuario y pulsar de nuevo Enter. La conexión desde una pantalla de conexión gráfica arranca automáticamente el sistema X Window.

Una vez que se inicie el sistema X Window, se encontrará una interfaz gráfica conocida como un escritorio, parecido a la siguiente figura:



**Figura 5-2. El escritorio gráfico**

### 5.1.2. Abrir un intérprete de comandos de la shell

El escritorio ofrece acceso a un intérprete de comandos para poder ejecutar comandos sin salir del interfaz gráfico. Para abrir un intérprete de comandos de la shell hay que seleccionar *Menú principal / Herramientas del sistema / Terminal*.

También se puede arrancar un intérprete de comandos haciendo click con el botón derecho en el escritorio y seleccionando *Nuevo Terminal* desde el menú.

Para salir del intérprete de comandos, se puede hacer click en la X en la esquina superior derecha de la ventana del shell, escribir `exit` en el indicador de comandos, o presionar `[Ctrl]-[D]`.

### 5.1.3. Salir en modo gráfico y apagar el ordenador

Para salir de la sesión de escritorio gráfico, selecciona *Menú principal / Terminar la sesión*. Cuando el diálogo de confirmación aparece como se muestra en la Figura 5-3, selecciona la opción *Terminar la sesión* y haz click en el botón *OK*. Para guardar la configuración del escritorio, así como también cualquier programa que se esté ejecutando, selecciona la opción *Guardar la configuración actual*.

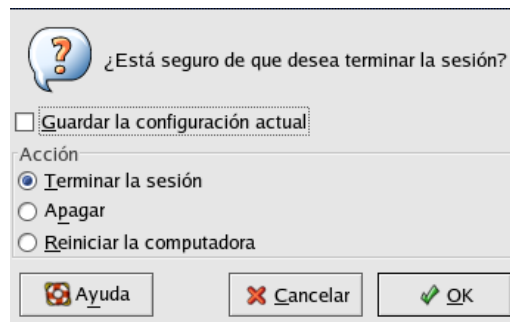


Figura 5-3. Confirmación de la salida

Antes de apagar el ordenador, es importante que cerrar apropiadamente Linux, de lo contrario se pueden perder datos que no han sido guardados o dañar el sistema.

La opción *Apagar* es equivalente al comando *poweroff* y la opción *Reiniciar la computadora* es equivalente al comando *reboot*.

## 5.2. Elementos del escritorio gráfico

En esta sección se ofrecen las nociones básicas del entorno de escritorio GNOME que es el que Red Hat incluye por defecto. El entorno gráfico da acceso a las aplicaciones y configuraciones del sistema. Ofrece tres herramientas principales para hacer uso de las aplicaciones en el sistema: iconos del panel, iconos del escritorio y menús.

La barra larga a lo largo de la parte inferior del escritorio es el *Panel*. El panel contiene iconos lanzadores de aplicación, indicadores de estado y pequeñas aplicaciones llamadas applets.

Los iconos del resto del escritorio pueden ser carpetas de archivos o lanzadores de aplicaciones y dispositivos removibles como CD-ROMs y disquetes cuando han sido montados. Para abrir una carpeta o lanzar una aplicación, hacer doble click en el icono correspondiente.

Para acceder a los sistemas de menú, hacer click en el botón de **Menú principal**:



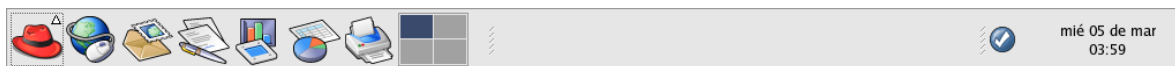
También se pueden encontrar haciendo doble click en el icono **Empezar aquí** en el escritorio y luego haciendo click en el icono de **Aplicaciones**.



El funcionamiento del escritorio gráfico es similar al que se podría esperar cuando trabajas con otros sistemas operativos. Puedes arrastrar y soltar archivos e iconos de aplicaciones en áreas de fácil acceso.

### 5.2.1. El Panel

El Panel del escritorio es la barra que atraviesa toda la parte inferior de la pantalla y que contiene los iconos y las pequeñas aplicaciones que facilitan el uso del sistema. El panel también contiene el **Menú principal**, que incluye los elementos de menú de todas las aplicaciones. Las pequeñas aplicaciones (denominadas applets) permiten ejecutar tareas específicas o controlar el sistema o los servicios.



### 5.2.2. Menú principal

Puedes hacer click en el botón **Menú principal** para expandirlo a un gran número de menús que permiten acceder a las aplicaciones del sistema.



Desde el **Menú principal**, puedes desconectarte, ejecutar aplicaciones desde la línea de comandos, encontrar archivos y bloquear la pantalla (la cual se ejecuta en un protector de pantalla protegido con una contraseña).

### 5.2.3. Applets

Los applets son pequeñas aplicaciones que se ejecutan en el panel. Normalmente permiten controlar varios aspectos del sistema. Algunos applets tienen tareas más específicas y otros están simplemente destinados al entretenimiento. Hay algunos applets que se ejecutan en el panel por defecto. Los más importantes son:

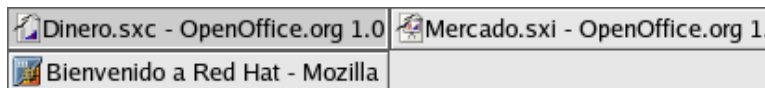
#### Paginador de espacios de trabajo

El escritorio ofrece la facilidad de usar múltiples espacios de trabajo para así no tener que ejecutar todas las aplicaciones en el mismo espacio de trabajo. El **Paginador de espacios de trabajo** representa cada escritorio en cuadros pequeños y muestra las aplicaciones que se están ejecutando en ellos. Hacer click en uno de esos cuadros con el ratón para moverse a ese escritorio. También se pueden usar las combinaciones de teclas [Ctrl]-[Alt]-[flecha arriba], [Ctrl]-[Alt]-[flecha abajo], [Ctrl]-[Alt]-[flecha derecha], o [Ctrl]-[Alt]-[flecha izquierda] para cambiarse entre escritorios.



#### Barra de tareas

Al lado del **Paginador de espacios de trabajo** esta la **barra de tareas**. La **barra de tareas** es un applet que muestra los títulos de las aplicaciones en ejecución en un escritorio virtual. Es muy útil si se decide minimizar una aplicación. Cuando se minimiza, se puede volver a mostrar haciendo click en el título correspondiente en la **Barra de tareas**.



#### Herramienta de notificación de Red Hat Network

El applet muestra diferentes imágenes que indican si el sistema está actualizado o si necesita actualizaciones. Si haces click en el icono, se mostrará una lista de las actualizaciones disponibles.





### El icono de autenticación

El icono de llave que a veces se despliega en el **Área de notificación** es una notificación de seguridad que se despliega cuando se ha obtenido acceso como usuario root para el sistema. Desaparece cuando expira el tiempo autenticación.



### Icono de notificación de impresión

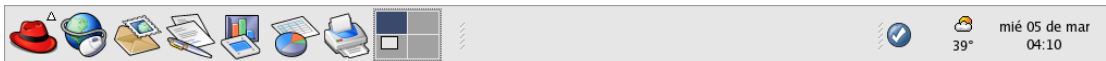
El **icono de notificación de impresión** permite manejar los trabajos de impresión. Haz click en el icono para ver los trabajos en ejecución y cancela trabajos haciendo click con el botón derecho sobre el trabajo y seleccionar **Cancelar**.



## 5.3. Configuración básica del entorno gráfico

### 5.3.1. Añadir iconos y applets al panel

Para añadir un nuevo applet al panel, haz click con el botón derecho en un área que no esté siendo usada en el panel y selecciona *Añadir al Panel* y selecciona alguno de los diferentes tipos de applets. Cuando seleccionas un applet, aparecerá en el panel. En la siguiente figura el applet del *Reporte del clima* ha sido añadido para mostrar la temperatura y el estado del tiempo local.



Para añadir un lanzador (nuevo icono) al panel, haz click en un área no usada del panel y selecciona *Añadir al panel / Lanzador...*. Con esta acción se lanzará un cuadro de diálogo que para introducir el nombre de la aplicación, la ubicación y el nombre del comando con el que el sistema lanzará la aplicación (como por ejemplo `/usr/bin/foo`), y se puede así mismo escoger un icono para la aplicación.

Otro modo sencillo y rápido de añadir un lanzador al panel es hacer click con el botón derecho en un área sin usar del panel y seleccionar *Añadir al Panel / Lanzador desde menú*. Luego selecciona una aplicación que aparecerá en el menú. Esto automáticamente añadirá un icono lanzador basado en las propiedades del ítem en el *Menú principal*.

### 5.3.2. Configuración del panel en el escritorio

Puedes ocultar el panel automáticamente o manualmente, colocarlo en cualquier lado del escritorio, cambiar su tamaño y color o cambiar la forma en que se comporta. Para alterar las configuraciones del panel, haz click con el botón derecho del ratón en un área no usada del panel y selecciona *Propiedades*. Puedes modificar el tamaño del panel, su posición y se desea, ocultar el panel automáticamente (*Autocultar*) cuando no se esté usando. Si seleccionas esta opción, no volverá a aparecer hasta que dirija la flecha del ratón hacia él (esto se llama *hovering*).

### 5.3.3. Uso del icono Empezar aquí

**Empezar aquí** fue diseñado para guardar todas las herramientas y aplicaciones que se pueden necesitar acceder cuando se usa el sistema. Desde aplicaciones favoritas hasta herramientas de configuración, la ventana *Empezar aquí* proporciona una ubicación central para el uso y personalización del sistema.

La pantalla *Empezar aquí* incluye iconos que permiten acceder a las aplicaciones favoritas, preferencias de escritorio, ítems del menú principal, herramientas de configuración del servidor y configuraciones del sistema.

Puedes añadir tus direcciones favoritas en *Marcadores*. Navega hasta la dirección que quieres marcar y luego selecciona *Marcadores / Añadir marcadores*

### 5.3.4. Cambiar el fondo del escritorio

Una forma de cambiar dramáticamente la apariencia del escritorio gráfico es cambiando el fondo utilizando la herramienta **Preferencias del fondo de pantalla**. Se pueden seleccionar imágenes incluidas con Red Hat en el directorio `/usr/share/backgrounds/`, o se pueden usar imágenes propias. Para iniciar la herramienta *Preferencias del fondo de pantalla*, haz click con el botón derecho y selecciona *Cambiar el fondo del escritorio* desde el menú. También puedes hacer doble click en el icono *Empezar aquí*, selecciona *Preferencias* y finalmente selecciona *Fondo*.

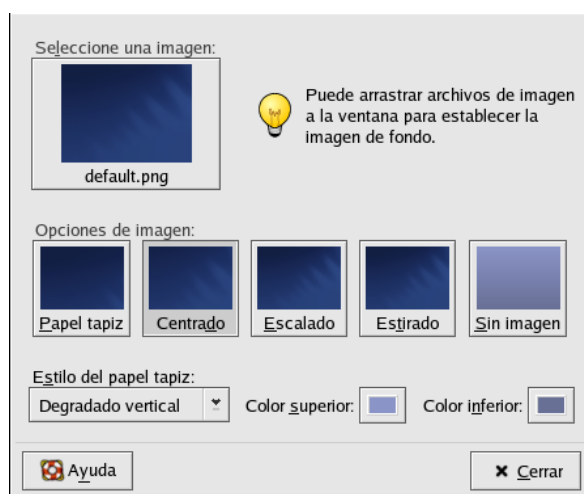


Figura 5-4. La herramienta Preferencias de fondo de pantalla

# 6. APLICACIONES EN ENTORNO GRÁFICO

## 6.1. Administrador de archivos: Nautilus

El escritorio gráfico incluye un administrador de archivos llamado **Nautilus** que da una visualización gráfica del sistema y archivos personales. Sin embargo, *Nautilus* está diseñado para ser mucho más que un listado visual de archivos. Permite configurar el escritorio, configurar el sistema, navegar por una colección de fotos, acceder a los recursos del sistema y mucho más, todo en una interfaz integrada. En esencia, *Nautilus* se convierte en una especie de *shell* de escritorio. Trabajar con *Nautilus* es eficaz y fácil, y proporciona un modo alternativo de buscar entre los distintos submenús conectados al *Menú principal*.

Para iniciar *Nautilus* como un administrador de archivos, haz doble click en el icono del directorio de inicio:



Una vez iniciado *Nautilus*, puedes navegar por el directorio de inicio o por el resto del sistema de archivos. Para volver al directorio de inicio, haz click en el botón *Inicio*.

El marco principal contiene las carpetas y los archivos, que se pueden arrastrar con el ratón para moverlos y copiarlos a nuevas ubicaciones. Se puede abrir otra ventana de *Nautilus* seleccionando *Archivo / Nueva ventana*. Una vez que tenemos otra ventana de *Nautilus*, se pueden arrastrar y soltar archivos a diferentes directorios. Por defecto, al arrastrar un archivo de un directorio a otro mueve el archivo. Para copiar el archivo a otro directorio, presionar la tecla [Ctrl] mientras se arrastra y suelta el archivo.

Por defecto, los archivos de tipo conocido del directorio principal se verán como *miniaturas*. Para los archivos de texto, esto significa que se verá una pequeña porción del texto actual en el icono. Para las imágenes, se verá la versión de la imagen reducida (o *miniatura*). Para desactivar esta característica, selecciona *Editar / Preferencias*. Seleccione la pestaña *Rendimiento*, luego seleccione *Nunca* en la sección *Ver muestras para los archivos de imágenes*. Al desactivar esta característica (y otras) de pre-visualización se incrementa la velocidad de *Nautilus*.

## 6.2. Trabajo con disquetes

### 6.2.1. Montaje y desmontaje de un disquete

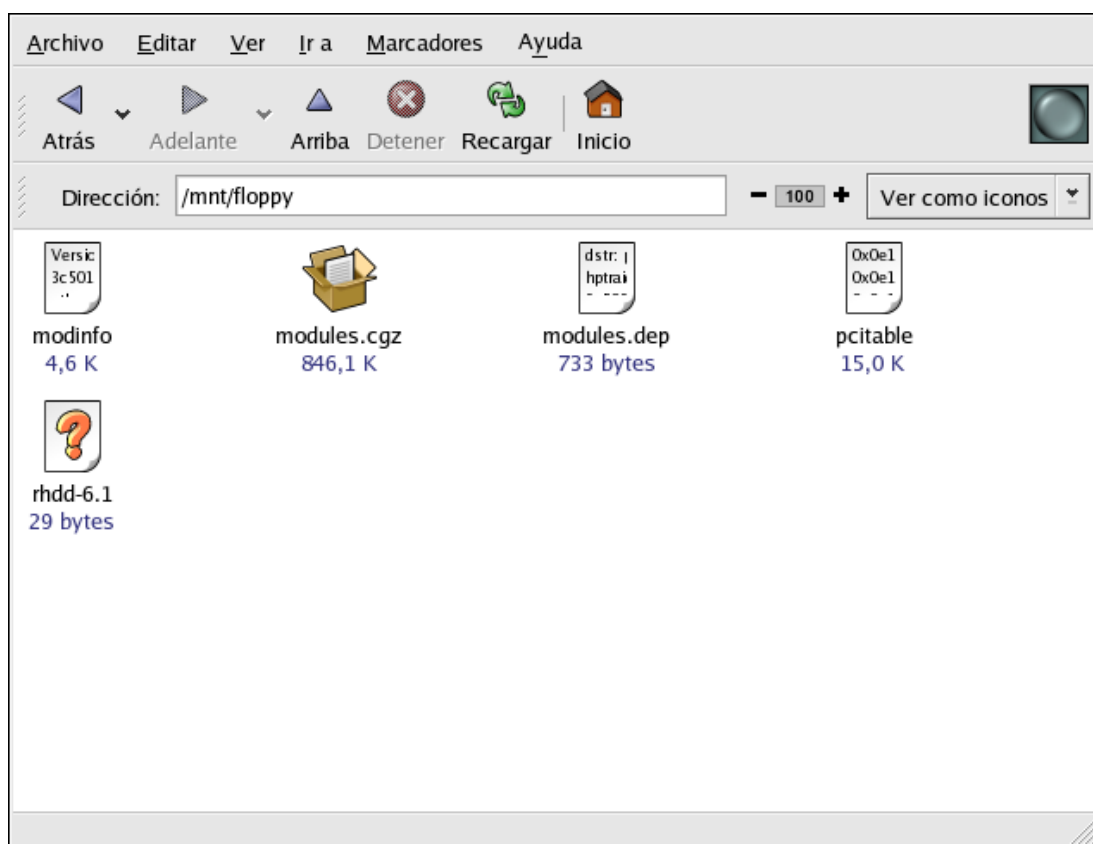
Se puede montar un disquete haciendo click con el botón derecho del ratón en el escritorio y seleccionando *Discos / Disquete*. Con esto se montará el disquete y se agregará el icono al escritorio sobre el que puedes hacer doble click para explorar el contenido del disquete.



El directorio donde se realiza el montaje del disquete es `/mnt/floppy`. Si el archivo `/etc/fstab` está correctamente configurado, el comando equivalente sería:

```
mount /mnt/floppy
```

Ahora que el disquete ya se ha montado, se puede copiar información del mismo y escribirse información en él. Puede abrir, guardar y copiar archivos al / del disquete del mismo modo que se haría con la unidad de disco duro. Incluso se puede explorar el contenido del disquete en la aplicación **Nautilus** (como se muestra en la Figura 6-1) o **Konqueror**.



**Figura 6-1. Ver los archivos de un disquete con la aplicación Nautilus**

Cuando se haya terminado de usar el disquete, se deberá desmontar antes de expulsarlo de la unidad. Para ello, cierra cualquier aplicación que pueda estar utilizando los archivos del disquete o explorando el contenido del disquete (por ejemplo, la aplicación **Nautilus** o **Konqueror**) y haz clic con el botón derecho del ratón en el icono y selecciona *Desmontar el volumen* en el menú. Ahora se puede expulsar de forma segura el disquete de la unidad.

El comando equivalente para desmontar:

```
umount /mnt/floppy
```

### 6.2.2. Formateo de disquetes: **gfloppy**

Para usar un disquete específicamente con el sistema Linux, se deberá dar formato utilizando el sistema de archivos ext2. El sistema ext2 es uno de los sistemas de archivos compatibles con Linux y es el método que usa por defecto para dar formato a los disquetes.

Una vez creado el sistema de archivos ext2 en el disquete, puedes manipular su contenido del mismo modo que los directorios y los archivos de la unidad de disco duro.

Para arrancar **gfloppy**, selecciona *Menú principal / Herramientas del sistema / Formateador de disquetes*. También se puede arrancar desde el indicador de comandos de shell, escribiendo `/usr/bin/gfloppy`. Como se muestra en la Figura 6-2, la interfaz *gfloppy* es pequeña y tiene pocas opciones. La configuración por defecto es suficiente para satisfacer a la mayoría de los usuarios y las necesidades. Sin embargo, se puede dar formato al disquete con el tipo de sistema de archivos de MS-DOS si se desea. También se puede seleccionar la densidad del disquete (si no se utiliza la alta densidad normal de los disquetes de 3,5" de 1.44 MB). También se puede optar por dar *formato rápido* al disquete si anteriormente fue formateado con ext2.

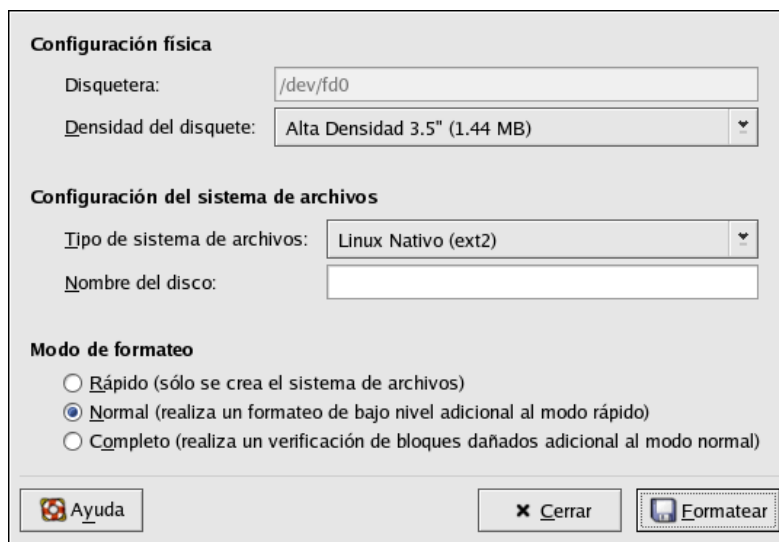


Figura 6-2. **gfloppy**

Inserta un disquete y modifica la configuración en *gfloppy* para adaptarla a las necesidades específicas y, a continuación, haz clic en *Formatear*. El cuadro de estado aparecerá en la parte superior de la ventana principal y mostrará el estado del proceso de formato y verificación. Una vez terminado, se puede expulsar el disquete y cerrar *gfloppy*.

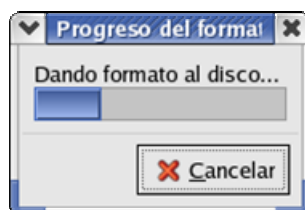
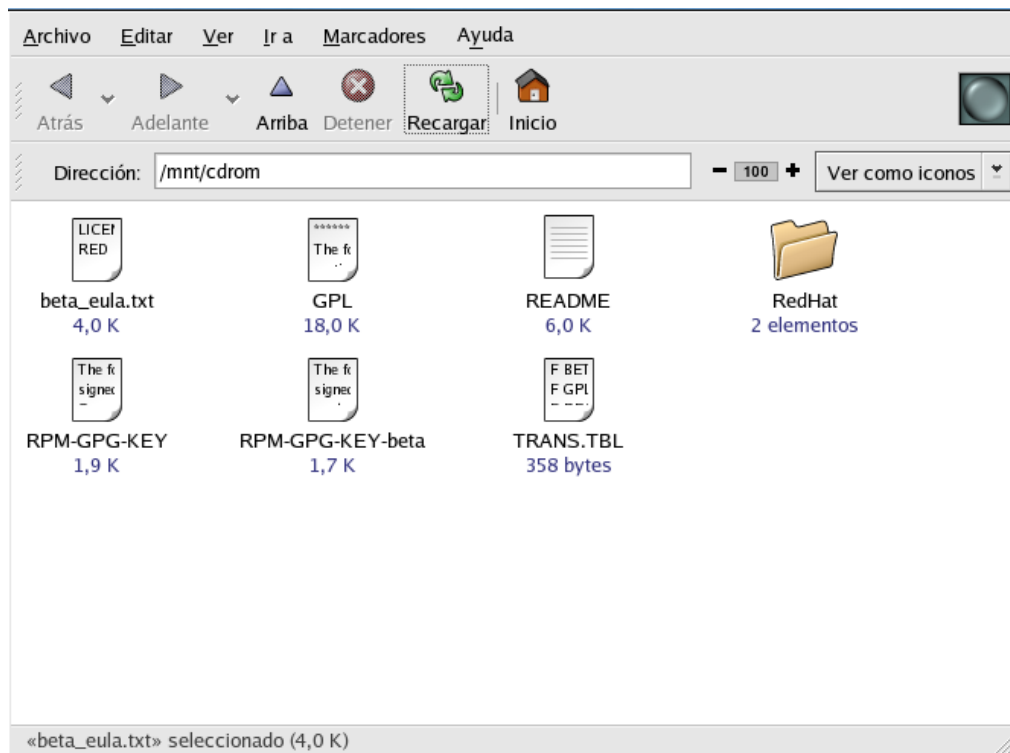


Figura 6-3. Cuadro de estado de **gfloppy**

El comando equivalente para formatear un disquete con formato tipo ext2 sería:  
`/sbin/mke2fs /dev/fd0`

### 6.3. Acceso a un disco CD-ROM

Por defecto, el sistema detecta automáticamente si un disco CD-ROM se inserta en la unidad de CD-ROM. Se montará el disco y el administrador de archivos abrirá una ventana con los contenidos del CD. La Figura 6-4 muestra los contenidos del CD-ROM dentro del administrador de archivos **Nautilus**.



**Figura 6-4. Contenidos de un CD-ROM en Nautilus**

El directorio de montaje utilizado es `/mnt/cdrom/`. Si el archivo `/etc/fstab` está correctamente configurado, el comando equivalente que se ejecuta al insertar un CD-ROM sería:

```
mount /mnt/cdrom
```

Aparecerá un icono del CD en el escritorio, el cual puedes utilizar para desmontar y expulsar el CD-ROM después de usarlo.



Después de trabajar con el CD, debes desmontarlo antes de expulsarlo de la unidad de CD-ROM. Cierra cualquier aplicación o los administradores de archivos que estén utilizando el CD-ROM y con el botón derecho del ratón pulsar el icono para visualizar todas las opciones disponibles. Para desmontar y expulsar el CD-ROM, seleccionar *Expulsar* desde el menú.

El comando en modo texto equivalente para desmontar sería:

```
umount /mnt/cdrom
```

## 6.4. Trabajo con archivos de texto

### 6.4.1. gedit

Los archivos de texto plano son archivos sin formatos de fuente o estilos especiales aplicados a él, tales como archivos de configuración o registros del sistema.

**gedit** es un editor gráfico de texto. Permite abrir, modificar y guardar los archivos de texto. También puede cortar y pegar texto desde y hacia otras aplicaciones de escritorio, crear nuevos archivos de texto e imprimirlos. *gedit* tiene una interfaz clara y fácil de entender que usa pestañas para así poder abrir más de un archivo al mismo tiempo sin abrir más ventanas de *gedit*.

Para arrancar *gedit*, haz click en *Menú principal / Accesorios / Editor de texto*. También puedes arrancar *gedit* escribiendo *gedit* en el intérprete de comandos.

*gedit* sólo se puede usar en un entorno de escritorio gráfico.

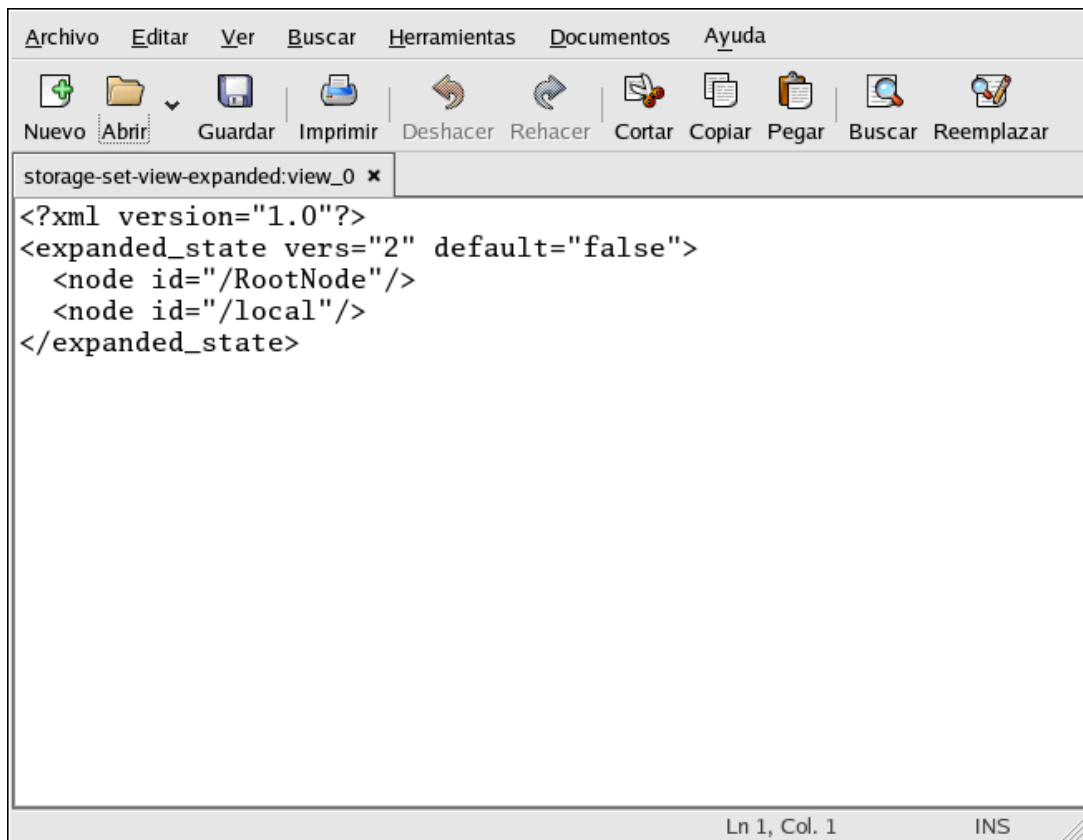


Figura 6-5. gedit

Una vez que *gedit* se esté ejecutando, se presentará con un área de edición en blanco. Se puede comenzar a usar *gedit* haciendo click en el botón *Abrir* para localizar el archivo de texto plano que se desea modificar. El archivo se cargará en el área de edición como se muestra en la Figura 6-5. Se puede navegar en el archivo de texto haciendo click y manteniendo la barra de desplazamiento en el borde derecho de la ventana y moviendo el cursor del ratón hacia arriba y hacia abajo; o usando las teclas de dirección para navegar a través del texto línea por línea. Presiona las teclas [Pag. Arriba] y [Pag. Abajo] para avanzar el documento de página en página.

*gedit* permite abrir varios archivos de texto simultáneamente usando pestañas para separar los archivos. Si el archivo ya está abierto y quieres copiar texto desde otro archivo, haz click en *Abrir*, selecciona el archivo al que quieras acceder, y el archivo se abrirá en una ventana *gedit*. Puedes navegar entre cada archivo haciendo click en la pestaña asociada con un nombre de archivo particular.

### 6.4.2. Visualización de PDFs: *xpdf*

Un archivo PDF (Portable Document Format) es una imagen electrónica de un documento. El PDF captura la información del formato de una gran variedad de aplicaciones de publicaciones con lo que es posible mandar documentos formateados y que aparezcan en el monitor del receptor o imprimirlos. Para visualizar un archivo PDF es necesario que tener un lector PDF.

Red Hat incluye la aplicación open source *xpdf*. La barra de herramientas de *xpdf* en la parte inferior tiene herramientas de navegación que permiten moverse en el documento así como hacer zoom, imprimir y utilizar herramientas para la búsqueda. La página del manual de *xpdf* ofrece información útil sobre las opciones de *xpdf*. Se puede visualizar la página man de *xpdf*, escribiendo en el indicador de comandos de la shell el comando `man xpdf`.

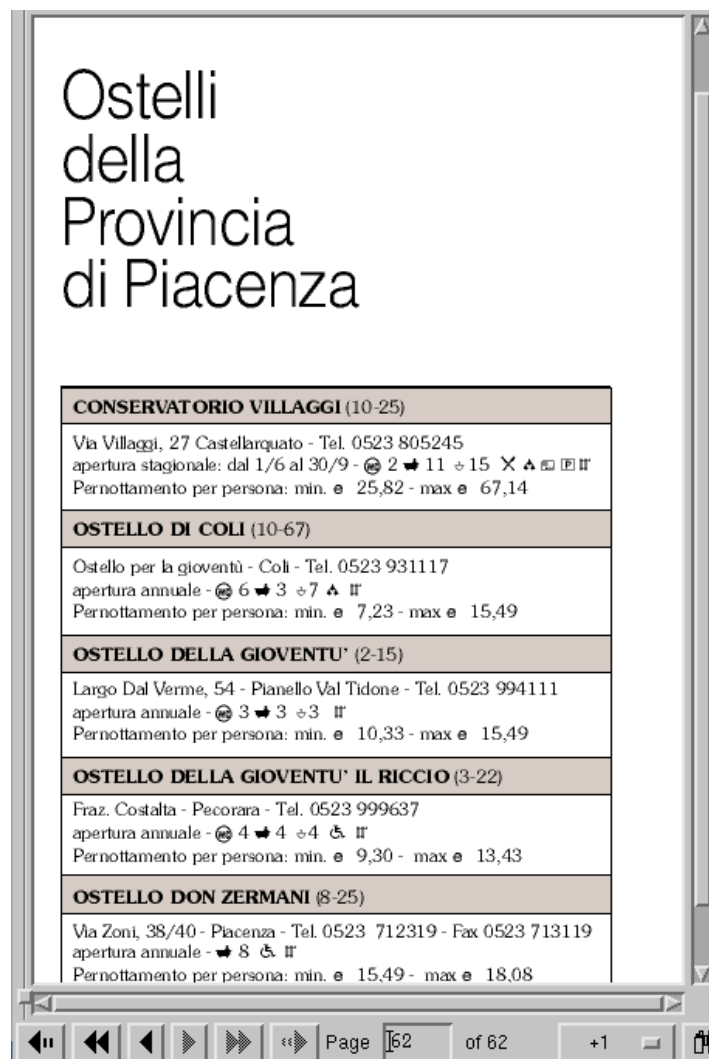


Figura 6-6. *xpdf*



Para visualizar un archivo PDF con *xpdf*:

En el escritorio, ir a *Menú principal / Gráficos / Visor de PDF*. También se puede ejecutar *xpdf* escribiendo *xpdf* desde el indicador de comandos de la shell.

Haz click con el botón derecho del ratón en la pantalla *xpdf* para visualizar una lista con las opciones.

Selecciona *Abrir* para mostrar el navegador de archivos. Selecciona el archivo PDF que desees ver y haz click en **Abrir**.

Otro visualizador de archivos PDF es **Adobe Acrobat Reader**. No está incluido en Red Hat Linux pero lo puedes descargar gratis del sitio <http://www.adobe.com/>.

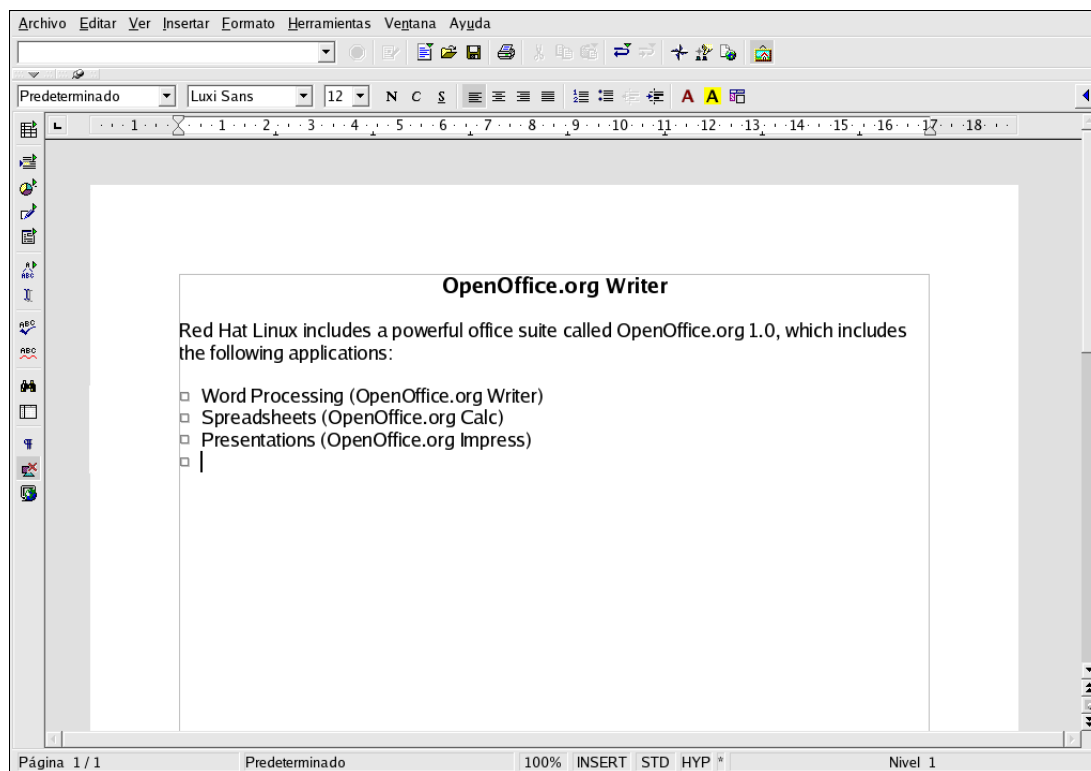
## 6.5. Trabajar con documentos: OpenOffice.org

Red Hat incluye un poderoso paquete ofimático que se llama **OpenOffice.org**, y que incorpora aplicaciones complementarias en un sólo paquete integrado.

El paquete *OpenOffice.org* contiene varias aplicaciones para la creación y la modificación de documentos, hojas de cálculo, presentaciones y diseño. Incluye plantillas y asistentes de configuración para la creación rápida de documentos profesionales básicos y presentaciones. El paquete *OpenOffice.org* puede leer, modificar y crear archivos en varios formatos incluso los que normalmente son del paquete Microsoft Office.

### 6.5.1. OpenOffice.org Writer

Escribir documentos con **OpenOffice.org** es parecido a las aplicaciones de los procesadores de texto que se usan en entorno Windows. Un procesador de texto es como un editor de textos pero tiene características adicionales que te permiten formatear, diseñar e imprimir los documentos sin necesidad de memorizar tipos de formatos complicados o códigos.



**Figura 6-7. OpenOffice.org Writer**

Para arrancar el *OpenOffice.org Writer* desde el panel del escritorio, elige *Menú Principal / Ofimática / OpenOffice.org Writer*; para hacerlo desde el indicador de comandos de la shell escribe el comando `oowriter`.

La interfaz principal es el área para la modificación del documento (el espacio en blanco en el medio de la pantalla en el que escribes el texto). En la parte de arriba de la pantalla se encuentran diversas funciones que aparecen en barras de herramientas que controlan el tamaño de la letra, el tipo de letra, *justificaciones* (para alinear el texto hacia la derecha, al centro o a la derecha) así como otros botones para el formateo del documento. También hay una caja de texto que permite escribir en la posición exacta de un documento en el ordenador y cargar el documento en el área de modificación. Puedes encontrar botones para abrir, salvar e imprimir documentos así como botones para crear nuevos documentos (que abre una hoja en blanco en la que puede escribir un nuevo documento).

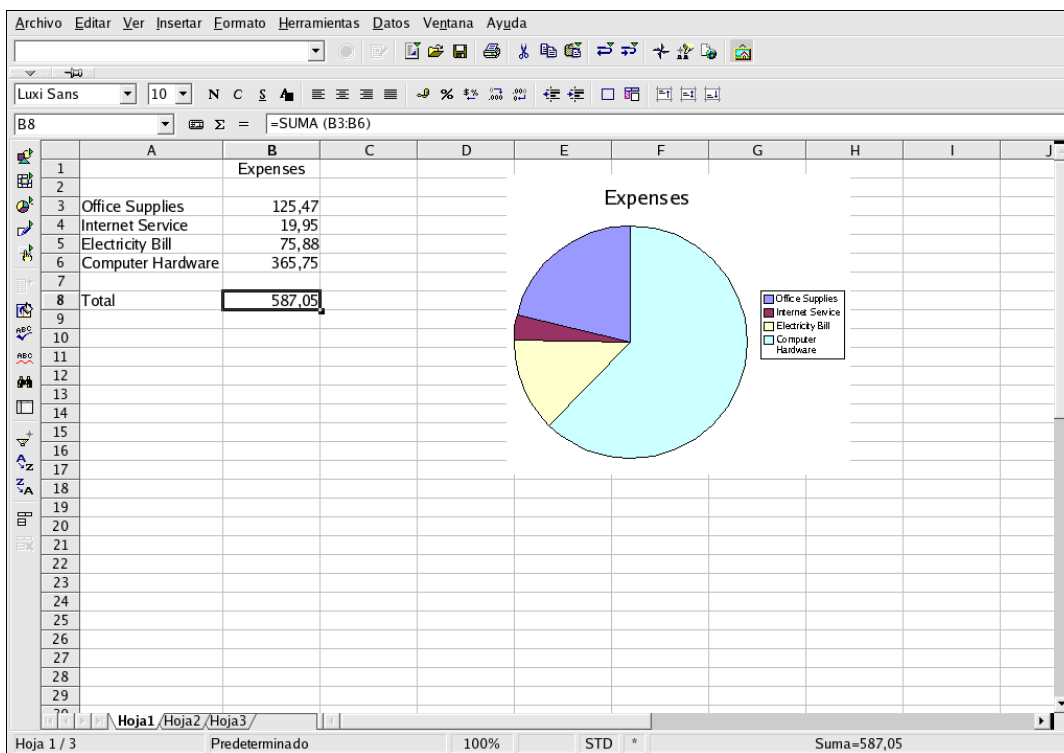
En el margen izquierdo de la pantalla, hay una barra de herramientas con botones para la ortografía, para señalar las palabras mal escritas, para la búsqueda de palabras y frases y otra serie de funciones. Si se pone la flecha del ratón en uno de estos botones, aparece una pequeña explicación sobre para qué sirve el botón. Puedes encontrar más información pulsando en *Ayuda* y escogiendo *Más sugerencias*.

Para salvar el texto, haz click en *Guardar*, el cual abre el navegador de archivos. Puedes escoger el formato de archivos desde el menú desplegable *Tipo de archivo* en la parte inferior de la ventana del navegador. El tipo de archivo predeterminado se adapta solamente a los archivos en los que esté usando las aplicaciones del *OpenOffice.org*. Sin embargo, puedes salvarlo como tipo de archivo *Microsoft Word*. También puede exportar el documento al formato HTTP o PDF, formatos que casi todos los ordenadores con navegador web o aplicación para visualizar PDF pueden leer.

### 6.5.2. OpenOffice.org Calc

**OpenOffice.org Calc** es una aplicación para la creación de hojas de cálculo que permite introducir y manipular datos en *celdas* organizadas en columnas y filas. Una celda es un contenedor para partes individuales de información, tales como cantidades, etiquetas o fórmulas matemáticas. Puedes llevar a cabo cálculos en grupos de celdas (añadiendo o sustrayendo una columna de celdas) o crear gráficos basados en grupos de celdas.

Para arrancar la aplicación *OpenOffice.org Calc* desde el panel del escritorio, selecciona *Menú principal / Ofimática / OpenOffice.org Calc*. Desde el indicador de comandos de la shell, escribe el comando `oocalc`.



**Figura 6-8. OpenOffice.org Calc**

El funcionamiento es similar a otras aplicaciones de hoja de cálculo. Por ejemplo, para crear gráficos o tablas para las presentaciones, *OpenOffice.org* tiene diversas plantillas de gráficos. Selecciona las áreas en las que se desea crear el gráfico o la tabla y luego haz click en *Insertar / Diagrama...* En la ventana de *Tabla*, los rangos de datos que se pueden escoger serán mostrados en la caja de texto para que los personalices aún más si así lo deseas. Haz click en el botón *Siguiente* para ver los diferentes tipos de gráficos y diagramas que puedes crear. Selecciona el estilo que deseas, y haz click en *Crear*. El gráfico o la tabla aparecerá en la hoja de cálculo de la pantalla. Puedes moverlo en cualquier lugar para imprimirlo o salvarlo como objeto que podrás usar más adelante en los documentos *OpenOffice.org Writer* o en las presentaciones *OpenOffice.org Impress*.

Se pueden salvar las hojas de cálculo creadas con *OpenOffice.org Calc* en muchos formatos de archivos, incluyendo el formato nativo `.sxc` así como en formato compatible con Microsoft Office `.xls`.

### 6.5.3. OpenOffice.org Impress

**OpenOffice.org Impress** es una herramienta gráfica para la creación de presentaciones.

Para arrancar *OpenOffice.org Impress* desde el escritorio gráfico, selecciona *Menú principal / Ofimática / OpenOffice.org Impress*. Para ejecutar *OpenOffice.org Impress* desde el intérprete de comandos, escribe `oimpress`.

*OpenOffice.org Impress* contiene un asistente automatizado llamado AutoPilot que permite crear presentaciones con un conjunto de plantillas de estilo. Puedes crear páginas para la presentación con listas de puntos, gráficos o imágenes. Puedes incluso importar gráficos y tablas de la aplicación *OpenOffice.org Calc* a las hojas de la presentación.



**Figura 6-9. OpenOffice.org Impress**

Cuando se arranca por primera vez la aplicación *OpenOffice.org Impress*, verás la pantalla del AutoPilot y te pedirá información básica sobre el tipo de presentación que deseas crear. Puedes elegir el estilo de las páginas, la manera en la que deseas realizar la presentación en papel, en transparencias para los proyectores o en un monitor, así como efectos visuales que deseas aplicar a la presentación desde el ordenador.

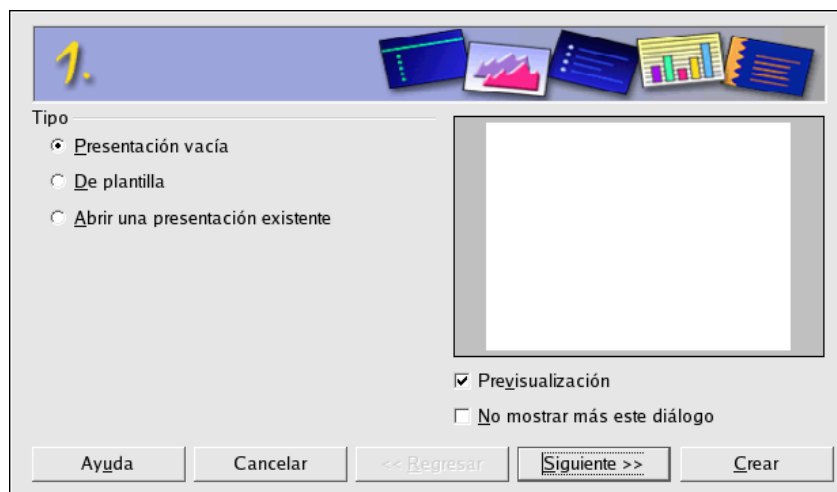


Figura 6-10. OpenOffice.org Impress Asistente de configuración AutoPilot

Una vez que has escogido las preferencias de la herramienta AutoPilot, puedes elegir el tipo de diapositiva que deseas crear. Puedes seleccionar el tipo predeterminado de la lista o empezar una diapositiva en blanco y personalizarla. Para crear diapositivas nuevas, haz click en *Insertar página...* en la barra de herramientas flotante y aparecerá una pantalla desplegable que te permitirá escoger el estilo de la nueva página.

Puedes visualizar la presentación en cualquier momento seleccionando *Presentación* desde los menús. La presentación ocupará toda la pantalla, para salir haz click en cada una de las páginas hasta que llegues a la última o pulsa la tecla [Esc].

Puedes salvar la presentación en diversos formatos como en el formato nativo de *OpenOffice.org Impress* (por ejemplo, *mipresentacion.sxi*), en el formato Microsoft PowerPoint (*mipresentacion.ppt*), o en el formato StarImpress (*mipresentacion.sdd*).

#### 6.5.4. OpenOffice.org Draw

Si quieres crear gráficos para incluirlos en los documentos y presentaciones, puedes usar la herramienta **OpenOffice.org Draw**.

Para arrancar *OpenOffice.org Draw* desde el panel del escritorio, haz click en *Menú principal / Ofimática / OpenOffice.org Draw*. Desde el indicador de comandos de la shell, escribe el comando `oodraw`.

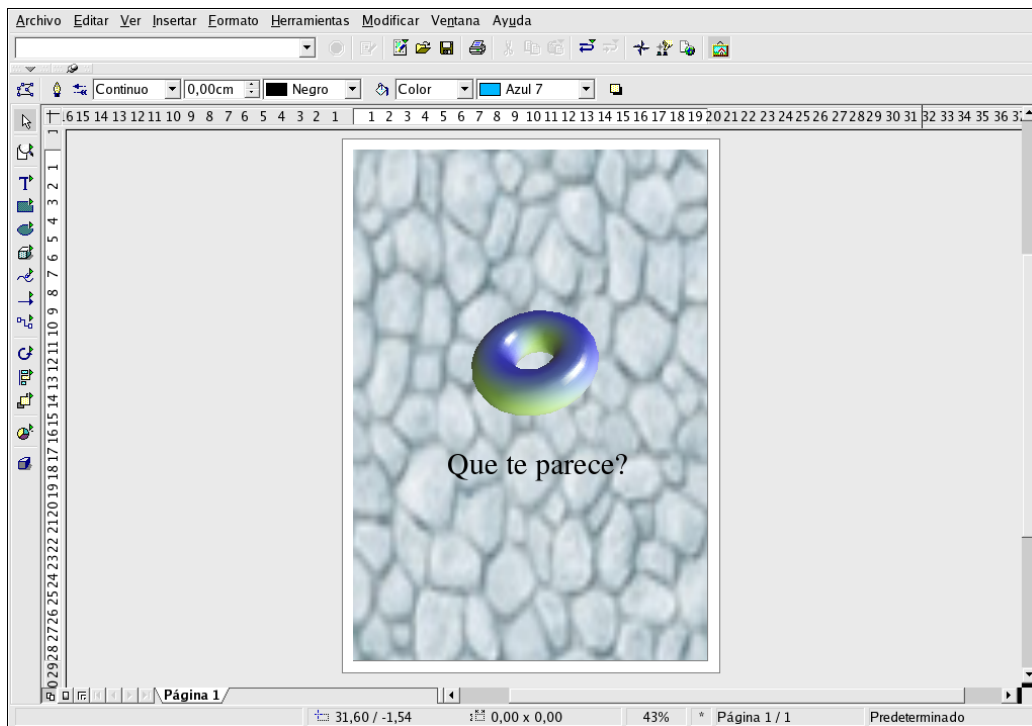


Figura 6-11. OpenOffice.org Draw

*OpenOffice.org Draw* contiene las herramientas básicas para crear ilustraciones o dibujos. Se puede salvar el archivo en uno de los formatos nativos o exportarlo para trabajar con un formato universal tal como .jpg o .png.

## 6.6. Navegación web

### 6.6.1. Mozilla

**Mozilla** es un navegador web potente, integrado, con soporte de estándares, cliente de correo, lector de noticias y más, que forma parte del gran rango de los desarrollos de aplicaciones Open Source de Internet de la organización mozilla.org. El componente del navegador muestra el contenido Web tales como páginas e imágenes. *Mozilla* también usa *plug-ins* para multimedia tal como vídeos o animación Web.

Para iniciar *Mozilla* haz click en el botón del panel *Navegador Web Mozilla* o selecciona *Menú principal / Internet / Navegador Web Mozilla*.



Figura 6-12. Navegador web Mozilla

### 6.6.2. Galeon

**Galeon** es un navegador Web basado en *Mozilla*. La diferencia es que **sólo** es un navegador Web. No contiene funciones para correo electrónico, grupos de noticias ni ninguna otra función que no sea navegar por la Web. Esto hace posible que Galeon sea un navegador rápido y compacto.

Para usar *Galeon*, debe estar instalado y configurado *Mozilla*. *Galeon* usa el motor de la versión de *Mozilla* para mostrar el contenido HTML e imágenes.

Para lanzar *Galeon*, seleccione *Menú principal / Internet / Más Internet / Galeon*.



Figura 6-13. Navegador Galeon

## 6.7. Aplicaciones de correo electrónico

Red Hat incluye varias aplicaciones de correo electrónico, como por ejemplo los clientes de correo electrónico **Evolution** y **Mozilla Mail** (incluye también un cliente en modo texto llamado **mutt**). Otro cliente de correo muy conocido en entorno Linux es **KMail**, que se incluye para el entorno gráfico KDE. Antes de lanzar el cliente de correo electrónico, deberías tener la información necesaria para la configuración de la cuenta de correo:

### La dirección de correo electrónico

La dirección de correo electrónico que utilizarás para enviar y recibir el correo. Normalmente adopta la forma de *tunombre@provedor.es*.

### Tipo de servidor para recibir correo (POP o IMAP)

Para recibir correo, debes saber el tipo de servidor que utiliza el proveedor de servicios de Internet. Esta dirección POP o IMAP normalmente adopta la forma de *mail.proveedor.es*.

POP, acrónimo de *Post Office Protocol*, se usa para enviar correo electrónico desde un servidor de correo al *buzón de correo* del cliente de correo electrónico. La mayoría de los servidores de correo electrónico ISP usan el protocolo POP, aunque puedan utilizar el protocolo más reciente IMAP. IMAP, acrónimo de *Internet Message Access Protocol*, es un protocolo para recuperar los mensajes de correo electrónico desde el servidor de correo electrónico ISP. IMAP se diferencia de POP en que los correos electrónicos de los servidores IMAP están almacenados en el servidor y permanecen allí incluso si descargas y lees el correo electrónico.



### Tipo de servidor para enviar correo electrónico (SMTP)

El *Simple Mail Transfer Protocol* (SMTP) (Protocolo de transferencia de correo simple), es un protocolo para enviar mensajes de correo electrónico entre servidores. La mayoría de los sistemas de correo electrónico que envían correo a través de Internet utilizan el protocolo SMTP para enviar mensajes de un servidor a otro; los mensajes se pueden recuperar posteriormente con un cliente de correo electrónico que utilice el protocolo POP o IMAP. SMTP se utiliza normalmente para enviar mensajes de un cliente de correo a un servidor de correo. Por este motivo, deberás especificar tanto el servidor POP o IMAP como el servidor SMTP cuando configures la aplicación de correo electrónico.

#### 6.7.1. Evolution

La aplicación **Evolution** es más que un simple cliente de correo electrónico. Proporciona todas las funciones de administración de correo electrónico estándar, que incluyen una administración eficaz de buzones, filtros definidos por el usuario y búsquedas rápidas. Admite el uso de un calendario/programa flexible y permite a los usuarios crear y confirmar reuniones de grupo en línea y eventos especiales. *Evolution* es la primera herramienta completa de administración de información personal y de trabajo en grupo para Linux y sistemas basados en UNIX y es el cliente del correo electrónico predeterminado para Red Hat Linux.

Para lanzar *Evolution* en el panel del escritorio, selecciona *Menú principal / Programas / Internet / Correo electrónico*.

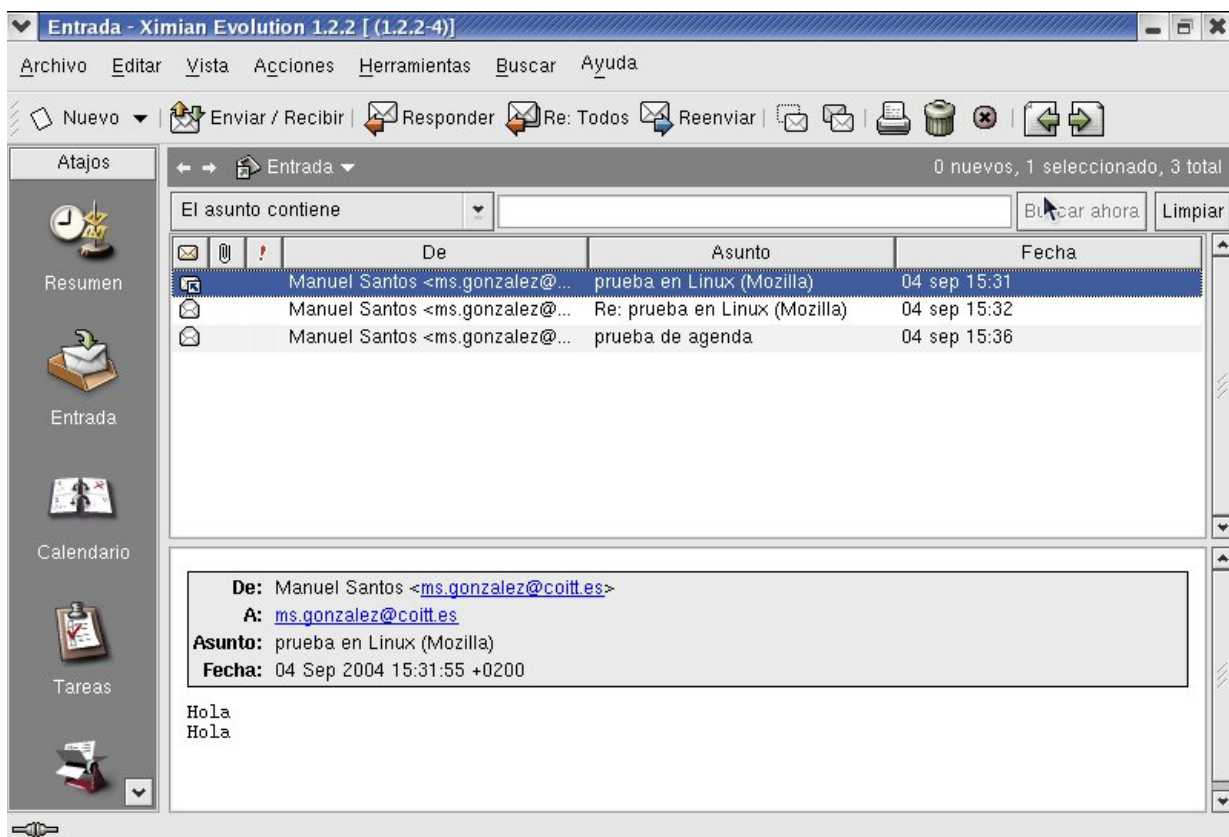



Figura 6-14. Evolution

## 6.7.2. Mozilla Mail

Para iniciar **Mozilla Mail**, selecciona *Menú principal / Todas las aplicaciones Internet / Mozilla Mail*.

Para abrir *Mozilla Mail* mientras está en la aplicación *Mozilla*, haz clic en el icono de correo situado en la esquina inferior izquierda de la pantalla de *Mozilla*. 

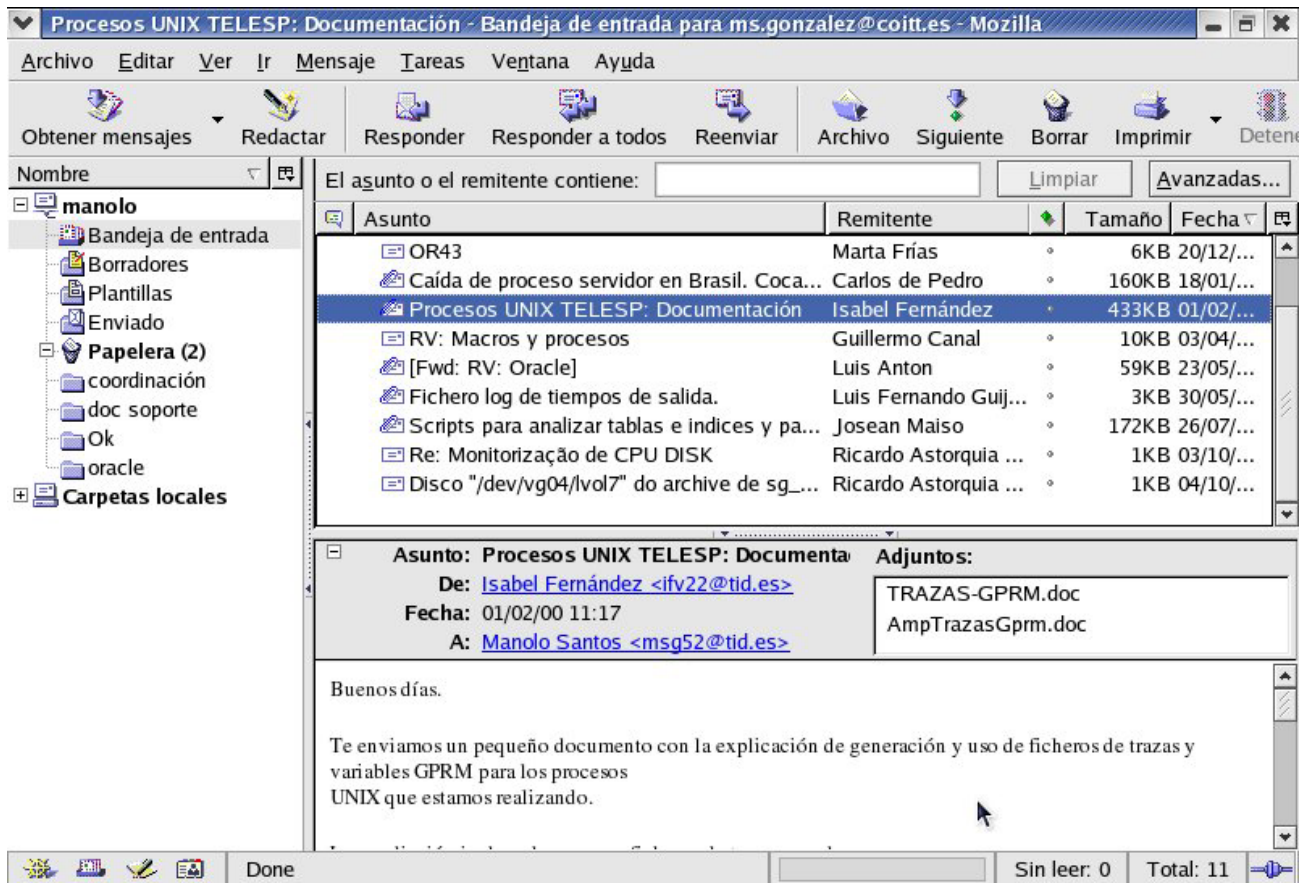


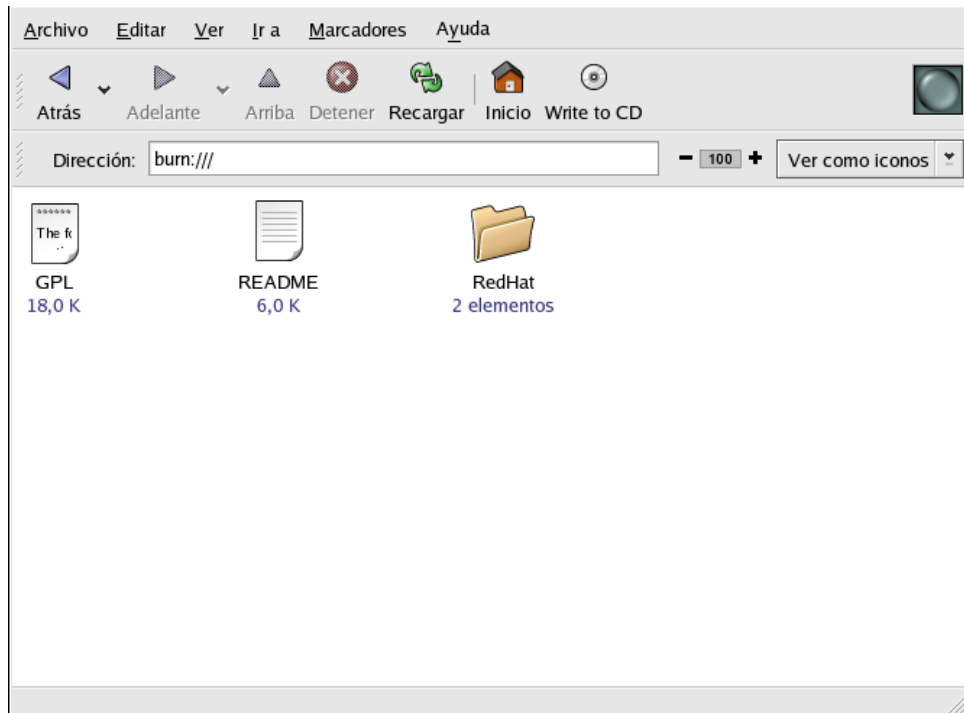
Figura 6-15. Mozilla Mail

## 6.8. Trabajar con CD-Rs y CD-RWs

### 6.8.1. Uso de la aplicación CD Creator

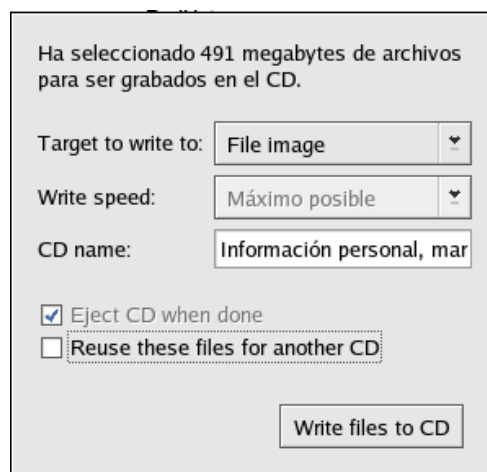
Si deseas realizar un respaldo rápido de un sistema de archivos o directorio a un CD-R o CD-RW, hay una herramienta incluida en el administrador de archivos *Nautilus* llamada **CD Creator**.

*CD Creator* te permite arrastrar y soltar archivos desde una ventana *Nautilus* a la interfaz de *CD Creator*. Para acceder a la característica de *CD Creator* en *Nautilus*, inserta un CD-R(W) en blanco en la unidad y la ventana de *CD Creator* aparecerá automáticamente. También puedes hacer doble clic al icono del directorio principal desde el escritorio y seleccionar *Ir a / CD Creator* desde la ventana de menú. También puedes escribir **burn:** en la barra de *Dirección* para arrancar el *CD Creator*.



**Figura 6-16. La interfaz de CD Creator en Nautilus**

Abre una nueva ventana de *Nautilus* y selecciona los archivos o directorios que deseas escribir al CD-R(W). Cuando esté todo listo para escribir los archivos al CD-R(W), haz click en *Write files to CD* en la ventana del *CD Creator* y se despliega una ventana de diálogo donde puedes seleccionar la velocidad de la escritura, nombre del CD y seleccionar otras opciones.



**Figura 6-17. La ventana de diálogo del CD Creator**

Haga click en el botón *Write files to CD* para comenzar a quemar. Una ventana de estado despliega el progreso de la escritura, como se muestra en la Figura.



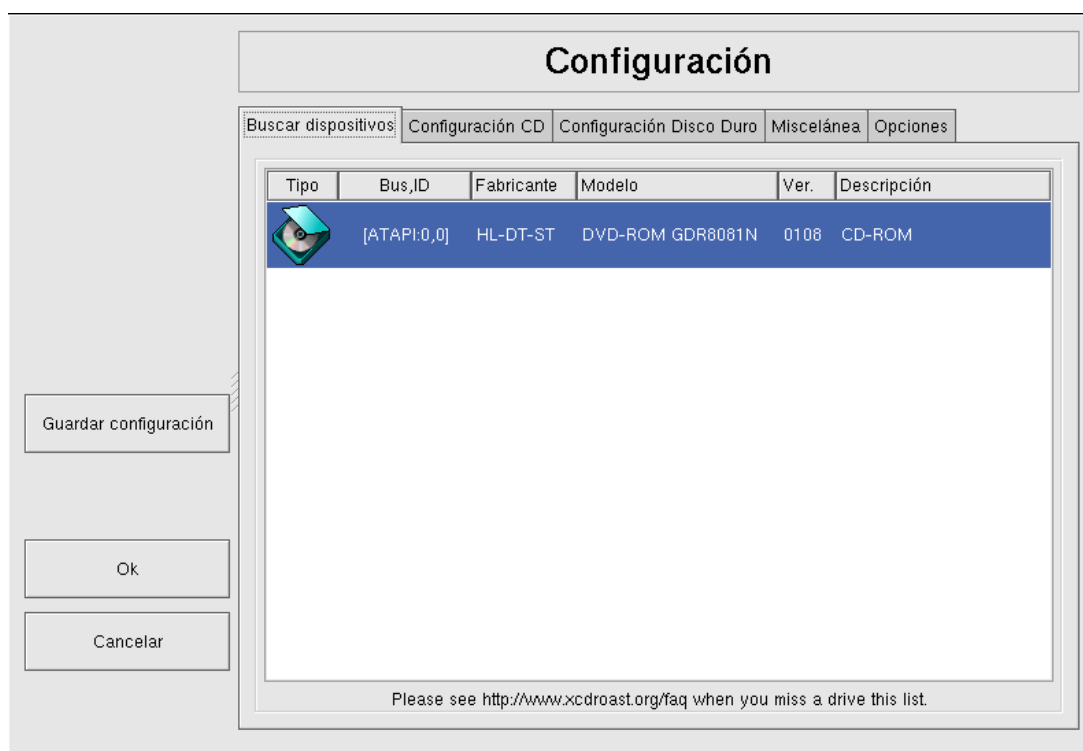
**Figura 6-18.** La ventana de estado de la escritura del CD Creator

Por defecto, el CD-R(W) debería automáticamente expulsar el CD de la unidad cuando termine.

### 6.8.2. Uso de X-CD-Roast

**X-CD-Roast** es una aplicación gráfica para duplicar y crear discos CD-ROM. La aplicación *X-CD-Roast* automatiza el proceso de quemar discos CD-R y CD-RW, y es altamente configurable para las distintas necesidades de creación y duplicación de CD.

Para arrancar *X-CD-Roast* selecciona *Menú principal / Herramientas del sistema / CD Writer*. Para arrancarlo desde el intérprete de comandos, escribe `/usr/bin/xcdroast`. *X-CD-Roast* primero escanea los buses para encontrar la unidad CD-R(W). Luego te permite configurar los parámetros para el CD-writer, unidad de CD-ROM y demás. La Figura ilustra la pantalla de *Configuración* y sus opciones.



**Figura 6-19.** Pantalla de configuración X-CD-Roast

**X-CD-Roast** está bien documentado dentro de la propia interfaz y dispone de varias opciones con sugerencias emergentes largas y descriptivas que te informan de la función asociada con todo

detalle. Puedes tener acceso a estas *sugerencias* colocando el puntero del ratón sobre el botón o menú desplegable durante dos segundos.

### Uso de X-CD-Roast para duplicar CD-ROMs

Para duplicar un CD-ROM existente con el fin de realizar una copia de seguridad, haz clic en el botón “*Duplicar CD*” en el panel principal. Puedes leer todas las *pistas* en un CD — toda la información del CD-ROM, incluidos los datos y los datos de audio se almacena en pistas — si haces clic en “*Leer CD*”. Puedes establecer la velocidad de lectura de un CD-ROM y obtener información sobre la pista de CD-ROM, como, por ejemplo, el tipo y el tamaño. Si copias las pistas de un CD de audio, puedes obtener una vista previa de cada pista haciendo clic en “*Escuchar pistas de audio*”. Puesto que la aplicación *X-CD-Roast* lee por defecto todas las pistas de un CD-ROM, puedes eliminar las pistas que no desees haciendo clic en el botón “*Eliminar pistas*”.

Por último, para quemar las pistas en el medio CD-R(W), selecciona “*Escribir CD*”. La Figura muestra la ventana de diálogo “*Escribir CD*”, donde se puede configurar la velocidad en la que lee y escribe las pistas al CD-R(W), así como también si quieres copiar el CD-ROM *on-the-fly* o crear una imagen del archivo antes de quemarlo. Haz clic en el botón “*Escribir CD*” para comenzar el proceso de quemado.

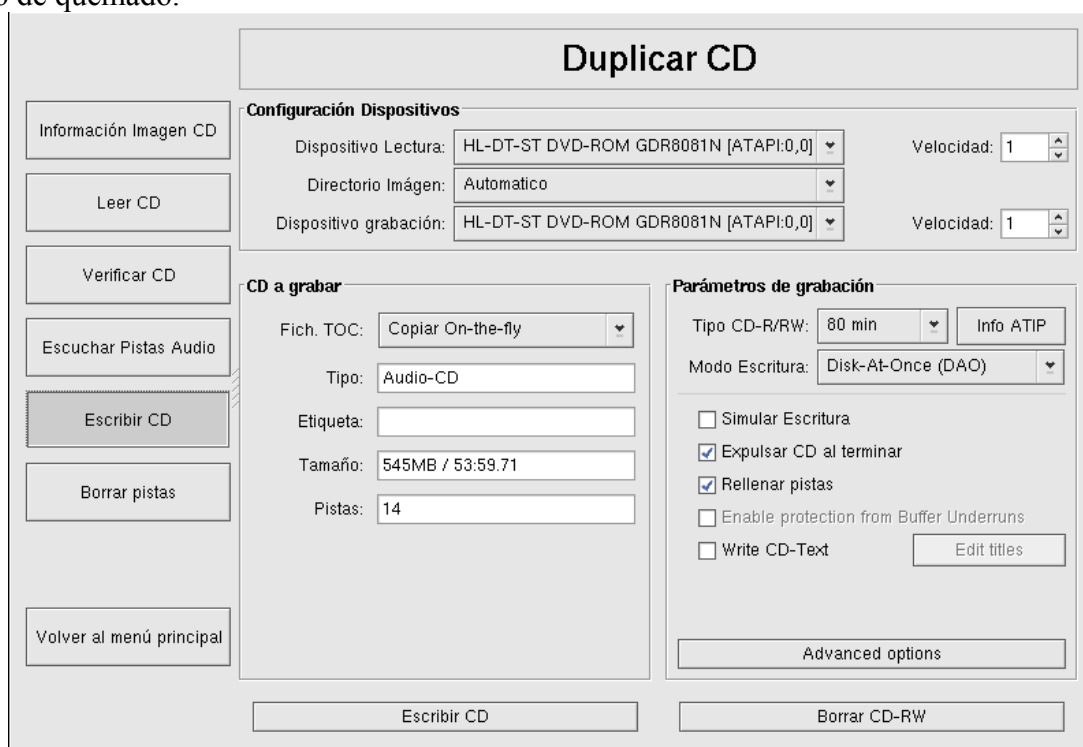


Figura 6-20. Uso de X-CD-Roast para duplicar CDs

### Uso de X-CD-Roast para crear un CD

La aplicación *X-CD-Roast* permite realizar una copia de seguridad de los archivos usando *Crear CD*. Esta función permite agregar archivos y directorios a una *sesión* de CD usando *Pistas maestro*. Hay otras opciones disponibles en la ventana de diálogo *Pistas maestro* que permiten configurar parámetros avanzados; sin embargo, los valores por defecto se establecen correctamente para crear CD-ROMs de datos, de modo que no es necesario realizar una configuración adicional. En la Figura 6-21 se muestra una sesión en la que se prepara todo el directorio `/home` para una copia de seguridad.

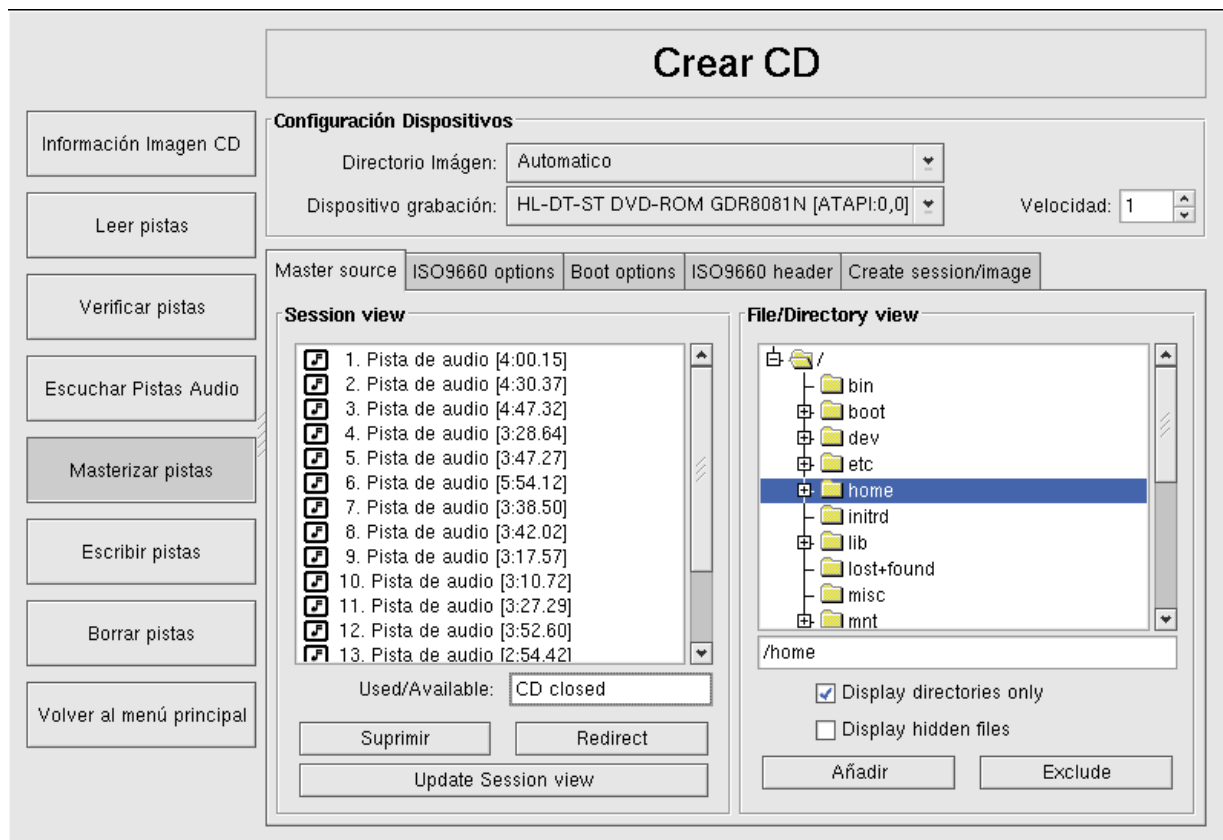


Figura 6-21. Uso de X-CD-Roast para respaldar archivos del disco duro

### Escribir ISOs con X-CD-Roast

Los archivos con la extensión `.iso` se conocen como *ISO9660* (o ISO) archivos de imagen. Por ejemplo, Red Hat Linux está disponible de forma gratuita mediante archivos ISO que puedes descargar y grabar en la unidad CD-R(W). En los sitios FTP y Web, hay otros archivos de imagen ISO disponibles. Existen otros tipos de archivos que se pueden quemar como imágenes, como por ejemplo `.img` y `.raw`, pero las imágenes ISO constituyen el formato de imagen del CD más común.

Para escribir un archivo de imagen ISO a una unidad CD-R(W) con la aplicación *X-CD-Roast* mueve el archivo ISO a la ruta especificada durante la instalación y, a continuación, haz click en “*Crear CD*”. En la pestaña *Formato de las pistas*, selecciona el archivo de imagen ISO que deseas quemar y haz click en *Añadir*, luego en *Aceptar formato de las pistas*. Esto carga automáticamente la pestaña *Escribir pistas*, donde puedes hacer click en *Escribir pistas* para quemar la imagen al CD-R(W).

Si se desea utilizar el indicador de comandos de la shell para escribir imágenes en las unidades CD-R y CD-R(W), hay dos utilidades disponibles: `mkisofs` y `cdrecord`. Estas dos utilidades disponen de una serie de opciones avanzadas cuya explicación no se trata en este manual. Sin embargo, para la creación y escritura de imágenes básicas, estas herramientas contribuyen a ahorrar tiempo respecto a las alternativas gráficas, como la aplicación *X-CD-Roast*.

## 6.9. Aplicaciones de audio: CD Player y XMMS

### 6.9.1. CD Player

Para escuchar un CD, introduce el CD en el CD-ROM. Aparecerá la aplicación **CD Player** automáticamente y comenzará a tocar la primera pista de audio. Si la interfaz no aparece, haz click en *Menú principal / Sonido & Video / Lector de CD* para lanzar la aplicación.

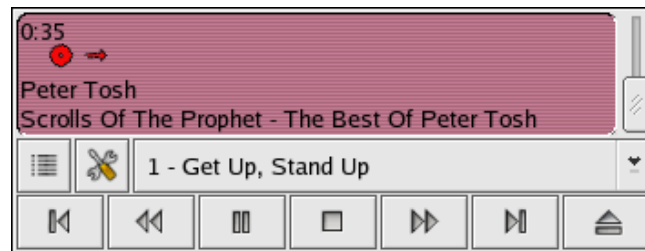


Figura 6-22. Interfaz del Lector de CD

La interfaz del *Lector de CDs* se comporta como un lector de CD estándar, con funciones de lectura, pausa y parada. Existe una barra que le permite ajustar el volumen. Pulsa los botones *Siguiente pista* y *Pista anterior* para saltar hacia adelante o atrás; también puedes ver la *Lista de pistas* para seleccionar una pista de la lista disponible.

También puedes modificar la lista de las pistas de tus CDs y cambiar el modo en que funciona la utilidad pulsando sobre el botón *Abrir editor de pistas*. También puedes modificar la forma en que la aplicación funciona haciendo click en el botón *Abrir Preferencias*. Aquí puedes configurar preferencias de temas para el reproductor así como también del comportamiento de la unidad de CD-ROM cuando abras o cierres la aplicación *Lector de CDs*.

### 6.9.2. XMMS

**XMMS** es un lector de multimedios que permite escuchar música en archivos de diferentes formatos.

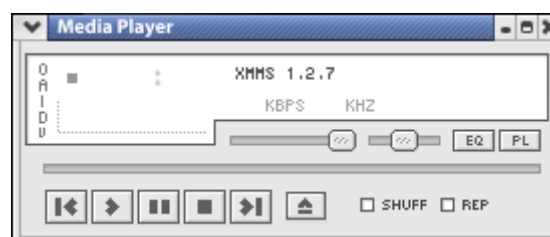


Figura 6-23. Interfaz de XMMS

*XMMS* puede usarse para más que simplemente escuchar los archivos audio. Por defecto, *XMMS* pueden leer audio Ogg Vorbis, RIFF wave y la mayoría de los formatos de módulos. Adicionalmente, *XMMS* puede extenderse con plugins para leer un número de otros formatos de multimedia.

Para lanzar *XMMS* vaya a *Menú principal / Sonido & Video / XMMS*.

Para lanzar *XMMS* desde el intérprete de comandos, escriba el comando `xmms`.

## 6.10. Trabajar con imágenes

### 6.10.1. Nautilus para ver imágenes

El programa Nautilus se puede utilizar como visor de imágenes. Cuando se selecciona un directorio que contiene imágenes, Nautilus crea de forma automática miniaturas de las imágenes si se hace doble click en una de ellas se visualiza la imagen en su tamaño original.

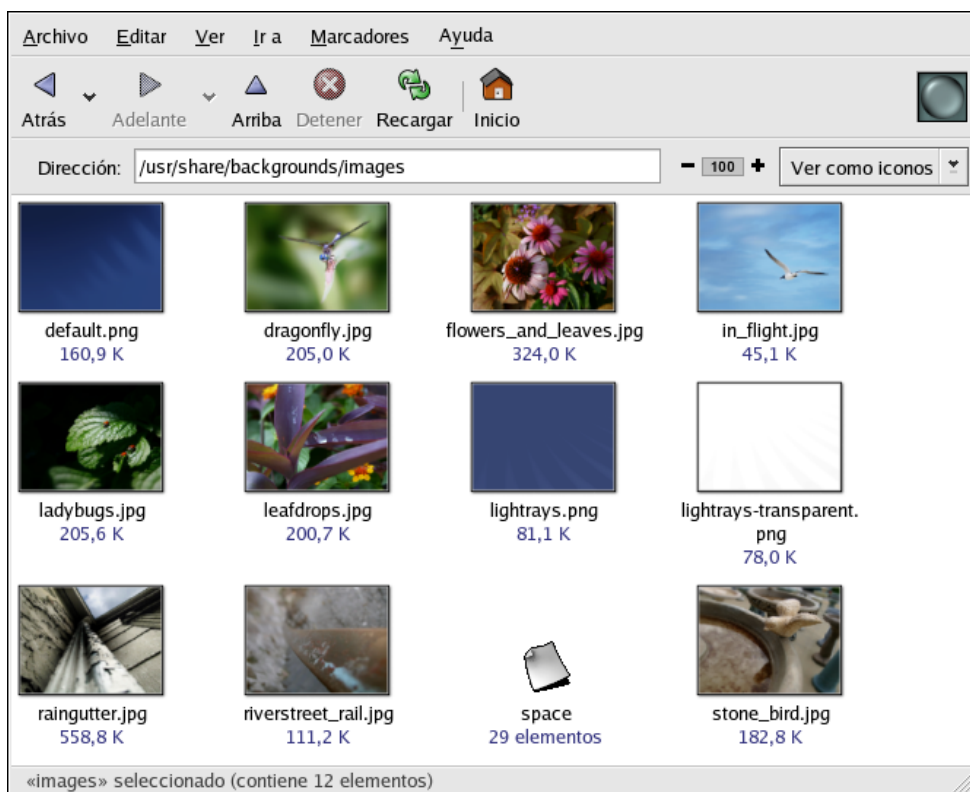


Figura 6-24. Contenido de una carpeta de Nautilus

### 6.10.2. Visor de imágenes gthumb

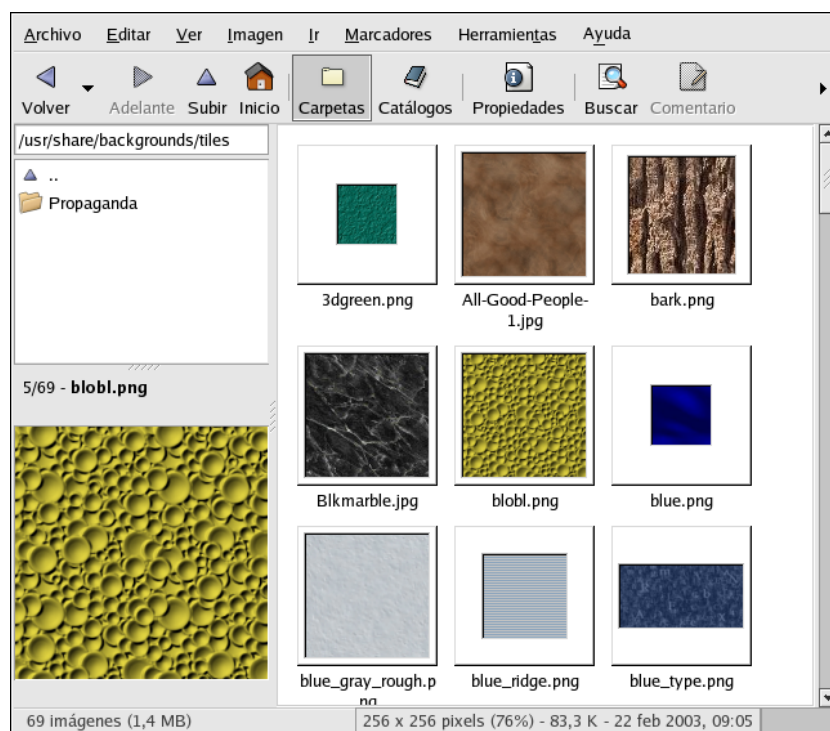
**gThumb** es un visor muy poderoso de imágenes para usuarios de escritorios gráficos que soporta muchos formatos de archivos de imagen, incluyendo:

- JPG/JPEG
- GIF
- PGM
- XPM
- PNG
- PCX
- TIF/TIFF
- PPM
- BMP



*gThumb* es útil para visualizar archivos de imagen individuales así como también para hojear colecciones de archivos en carpetas. Soporta las funciones de zoom, así como también los iconos vista previa en miniatura de todos los archivos de imágenes dentro de un directorio. También soporta muchas opciones avanzadas no disponibles en *Nautilus*.

**gThumb** puede ser ejecutado desde el panel del escritorio. Selecciona *Menú principal / Gráficos / Visor de imágenes gThumb* o escribe `gthumb` en el intérprete de comandos para lanzar la aplicación. *gThumb* por defecto hojeará el directorio principal.



**Figura 6-25. gThumb mostrando una carpeta de imágenes**

La interfaz de *gThumb* es muy sencilla. Haz doble-click en una miniatura de una imagen para verla dentro del área principal de la galería. La imagen puede ser vista con el zoom, colocarse en pantalla completa (lo que cubre completamente la pantalla con la imagen), o ser impresa en la impresora configurada. La barra de herramientas permite ajustar la imagen a la ventana de visualización, reunir varios archivos en un catálogo para un acceso más fácil si estos están ubicados en directorios diferentes y escribir descripciones sobre las imágenes.

La interfaz de *gThumb* también tiene un campo de texto para escribir la ruta particular a los directorios de imágenes. Presionando el botón derecho sobre una imagen en el área de visualización abre un menú con las opciones de administración de archivos tales como renombrar, mover, copiar y convertir una imagen de un formato a otro. Desde este menú también se puede colocar una imagen como papel tapiz del escritorio.

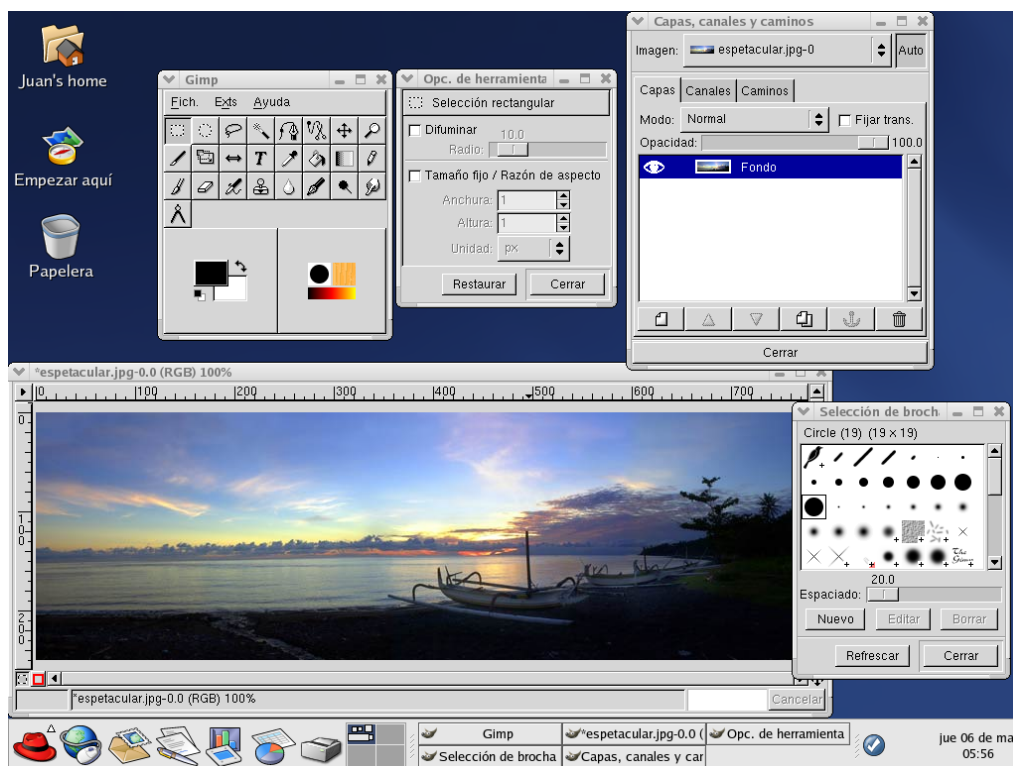
Puedes combinar funciones dentro de *gThumb* y crear un efecto de presentación dinámico para grupos de imágenes dentro de un directorio. En el campo de texto bajo la barra de herramientas, escribe la ruta al directorio donde están ubicadas las imágenes y resalta la primera imagen en el panel principal de la galería. Haz click en el botón *Presentación* en la barra de herramientas y comenzará el despliegue de imágenes de pantalla completa en el que *gThumb* muestra las imágenes. Por defecto, cada imagen en la presentación es presentada por 4 segundos. Puedes detener la

presentación en cualquier momento presionando [Esc] o moviendo el cursor del ratón y haciendo click en el botón *Restaurar vista normal* que aparece en la esquina superior izquierda de la pantalla.

### 6.10.3. GIMP

El programa **GIMP** (GNU Image Manipulation Program, Programa de manipulación de imágenes de GNU) es una eficaz herramienta para crear, alterar, manipular y mejorar archivos de imágenes digitales (fotografías, imágenes escaneadas e imágenes generadas por ordenador, entre otras). En esta sección se ofrece una rápida descripción del programa *GIMP*.

Desde el indicador de comandos de la shell, inicia la aplicación *GIMP* utilizando el comando `gimp`, o puedes arrancar la aplicación *GIMP* desde el escritorio, seleccionando *Menú principal / Gráficos / GIMP*.



**Figura 6-26. Sesión GIMP en ejecución**

Cuando guardes una imagen con *GIMP*, tendrás que seleccionar un formato de imagen. La aplicación *GIMP* soporta una amplia variedad de formatos de imágenes, incluyendo .gif, .png, .jpg, y .bmp.

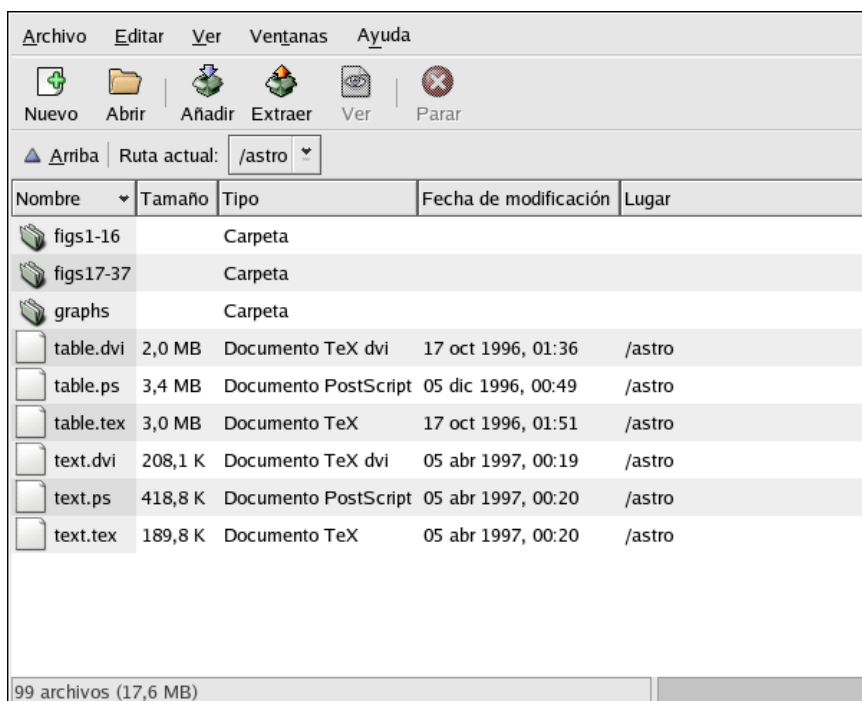
Al igual que muchas otras aplicaciones, *GIMP* proporciona más de un método para realizar las tareas. El modo más sencillo de trabajar con imágenes es hacer click con el botón derecho del ratón en la imagen, acción con la que se mostrará un conjunto de menús que contienen la mayoría de las distintas funciones de *GIMP*, incluyendo funciones para cambiar el tamaño de imágenes, girar imágenes y aplicar filtros.

Desde *Herramientas* también se puede tener acceso fácilmente a una serie de funciones. La opción *Herramientas* permite agregar texto a las imágenes, borrar regiones de una imagen o incluso rellenar regiones específicas con el color que seleccione.

## 6.11. Compresión de archivos: File Roller

Red Hat Linux incluye una utilidad gráfica llamada **File Roller** que puede comprimir, extraer y archivar archivos y directorios. *File Roller* soporta los formatos de archivado y compresión de archivos UNIX y Linux y posee una interfaz sencilla y documentación de ayuda si la necesitas. Está integrado en un entorno de escritorio para hacer la manipulación de archivos más fácil.

Para arrancar *File Roller* haz click en *Menú Principal / Accesorios / File Roller*. También puedes ejecutar *File Roller* desde el intérprete de comandos del shell escribiendo `file-roller`. La Figura 6-27 muestra *File Roller* en acción.



**Figura 6-27. File Roller en acción**

Si estás usando un administrador de archivos (tal como *Nautilus*), haz doble click en el archivo que deseas extraer y descomprimir para arrancar *File Roller*. La ventana del navegador de *File Roller* aparecerá con el archivo descomprimido/desempaquetado en una carpeta para que la extraigas o navegues por ella.

Para desempaquetar y/o descomprimir un archivo, haz click en el botón *Abrir* de la barra de herramientas. Aparecerá un menú de archivos que te permitirá escoger el archivo con el que deseas trabajar. El archivo aparecerá en la ventana del navegador principal *File Roller* como una carpeta, con la que puedes navegar haciendo doble click en el icono de la carpeta. *File Roller* preserva todas las jerarquías del directorio y subdirectorio, lo que es muy conveniente si estás buscando un archivo en particular en el archivo comprimido. Puedes extraer archivos individuales o el archivo completo haciendo click en el botón *Extraer*, escogiendo el directorio en el que quieres salvar los archivos extraídos y haciendo click en *OK*.

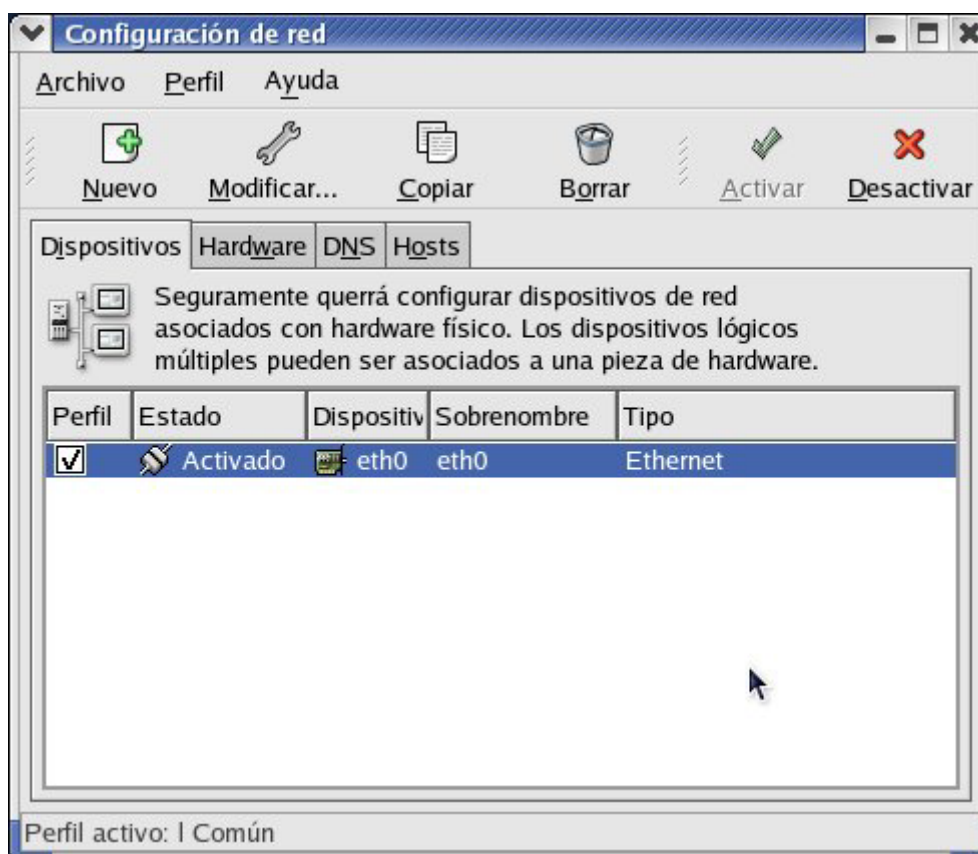
Para crear un nuevo archivo, haz click en *Nuevo* en la barra de herramientas. Aparecerá un navegador de archivos, permitiéndote especificar un nombre de archivo y la técnica de compresión. Por ejemplo, puedes escoger un formato *Tar Compressed with gzip (tar.gz)* desde el menú desplegable y escribir el nombre del archivo que deseas crear.

# 7. CONFIGURACIÓN Y TRABAJO EN RED

## 7.1. Configuración en entorno gráfico

Desde el entorno gráfico, la configuración de la red se realiza con la herramienta gráfica de configuración de la red: Network Administration Tool (/usr/bin/redhat-config-network)

El acceso a esta herramienta gráfica se puede realizar por el menú: *Menú principal / Configuración del sistema / Red.*



**Figura 7-1. Configuración de red**

En la pestaña *Dispositivos* aparecen todos los dispositivos físicos de red. Para una tarjeta de red Ethernet, el nombre suele ser eth0. Si hubiera una segunda tarjeta de red, se denominaría eth1. En esta pantalla podemos habilitar o deshabilitar el dispositivo de red.

Pulsando el botón “*Modificar...*” podemos configurar los parámetros importantes de la tarjeta de red como la dirección IP, máscara y la dirección de la puerta de enlace. También se puede seleccionar la obtención de direcciones IP de forma automática por DHCP.

En la pestaña DNS se pueden configurar las direcciones IP de los servidores DNS así como el nombre de la máquina.

En la pestaña *Hosts* se pueden añadir direcciones IP estáticas de máquinas a las cuales se quiere tener acceso a través de un nombre. Estas direcciones tienen preferencia sobre los DNS, es decir, si se especifica un nombre de máquina para cualquier operación, primero se busca ese nombre en la información que aparece en esta pestaña y si no aparece se realiza la petición al DNS.

## 7.2. Configuración en modo comando

En Linux, todas las comunicaciones de red se producen entre las interfaces software configuradas y los dispositivos físicos de red conectados en el sistema. Hay varias categorías de ficheros para la configuración y el uso de las interfaces de red:

### Ficheros de configuración de red

#### **/etc/hosts**

Utilizado para resolver nombres de máquinas que no pueden resolverse de otra forma, es decir, a través de DNS. También debería contener el nombre y dominio de la máquina local con la dirección de loopback (127.0.0.1).

#### **/etc/resolv.conf**

Este fichero especifica las direcciones IP de los servidores DNS (especificado mediante la palabra clave *nameserver*) y el nombre de dominio de búsqueda (especificado con la palabra clave *search*).

#### **/etc/sysconfig/network**

Contiene información del nombre de la máquina y alguna otra información común a todos los interfaces de red

### Ficheros de configuración de la interfaz

Controlan el interface software para cada dispositivo individual de red. Normalmente se nombran como `ifcfg-<nombre_interface>`, donde `<nombre_interface>` se refiere al nombre del dispositivo que el fichero de configuración controla. El nombre completo de estos ficheros sería:

```
/etc/sysconfig/network-scripts/ifcfg-<nombre_interface>
```

- Interfaces Ethernet

- ifcfg-eth0**, controla la primera tarjeta de red Ethernet del sistema.

- ifcfg-eth1**, controla la segunda tarjeta de red Ethernet del sistema.

- ...

- Interfaces de marcado

- ifcfg-ppp0**, controla la conexión a Internet a través de una conexión de marcado con protocolo PPP.

- ...

- Interface de bucle local (loopback)

- ifcfg-lo**, controla el interface de un bucle local utilizado sobre todo para pruebas.

- Aunque hay aplicaciones que lo utilizan de forma interna.

En caso de tener una tarjeta de red, esta se denominará `eth0`, y su fichero de configuración será `/etc/sysconfig/network-scripts/ifcfg-eth0`, en este fichero se almacenan los parámetros más importantes de la interfaz como la dirección IP, máscara de red, puerta de enlace, etc.

Ejemplo de fichero ifcfg-eth0:

```
# Please read /usr/share/doc/initscripts-*/sysconfig.txt
# for the documentation of these parameters.
USERCTL=no
PEERDNS=yes
TYPE=Ethernet
DEVICE=eth0
HWADDR=00:0c:29:65:3c:4b
BOOTPROTO=none
NETMASK=255.255.255.0
ONBOOT=yes
IPADDR=192.168.0.12
NETWORK=192.168.0.0
BROADCAST=192.168.0.255
GATEWAY=192.168.0.1
```

### Script de control de la interfaz

Son ficheros ejecutables que se utilizan para activar y desactivar las interfaces de red. Son los siguientes:

#### **/sbin/ifdown nombre\_interfaz**

Desactiva la interfaz de red para el dispositivo especificado. Por ejemplo:

```
# ifdown eth0
```

Desactiva la interfaz para la primera tarjeta de red del sistema.

#### **/sbin/ifup nombre\_interfaz**

Activa la interfaz de red para el dispositivo especificado. Por ejemplo:

```
#ifup eth0
```

Activa la interfaz de red para la primera tarjeta de red del sistema.

#### **/sbin/ifconfig**

Muestra información del estado de las interfaces de red que están activadas.

Ejemplo de salida de ifconfig:

```
eth0      Link encap:Ethernet  HWaddr 00:0C:29:65:3C:4B
          inet addr:192.168.0.12  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2345 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2294 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:1531608 (1.4 Mb)  TX bytes:291512 (284.6 Kb)
          Interrupt:10 Base address:0x10e0

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
```

```
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:102992 errors:0 dropped:0 overruns:0 frame:0
TX packets:102992 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:6642842 (6.3 Mb) TX bytes:6642842 (6.3 Mb)
```

Existe un comando para poder realizar acciones de control sobre todas las interfaces de red de forma simultánea:

**/sbin/service network <action>**

Donde <action> puede ser : start, restart, reload, stop o status

Por ejemplo, para reiniciar todas las interfaces de red:

```
# service network restart
```

Este comando es útil cuando hacemos algún cambio en algún parámetro de las interfaces. No es necesario reiniciar el sistema completo, sólo habría que reiniciar las interfaces de red con el comando anterior.

El comando service se puede utilizar de igual forma en cualquiera de los servicios que se encuentran definidos en el directorio /etc/init.d

### 7.3. Otros comandos

**ping <máquina>**

Utilizado para enviar paquetes de prueba a <máquina>. Sirve para comprobar si existe o no conectividad con otra máquina de la red.

```
# ping 192.168.0.20
```

**telnet <máquina>**

Comando para realizar una conexión a otra máquina remota. Para poder realizarse la conexión, la máquina remota debe tener activo un servidor telnet que atienda la petición de conexión.

**ssh <máquina>**

Comando que tiene el mismo propósito que telnet, sin embargo utiliza un protocolo de comunicación encriptado, más seguro que el utilizado por telnet. Por defecto toma el nombre de conexión del usuario local, para utilizar otro, emplear la opción -l:

```
$ ssh -l usuario2 192.168.0.20
```

**ftp <máquina>**

Comando para realizar transferencias de ficheros con otra máquina utilizando el protocolo ftp (File Transfer Protocol).



## 7.4. Uso del servicio NFS

Una de las herramientas de red más interesantes y fáciles de usar para compartir información es la utilización del protocolo NFS (Network File System). A través de este protocolo podremos montar en nuestro sistema de ficheros, un sistema de ficheros de otra máquina. Por tanto, una vez realizada dicha operación, el acceso a ficheros “remotos” se hará de forma transparente, como si los ficheros pertenecieran a nuestro sistema.

### Configuración de un cliente

Primero, vamos a ver cómo podemos añadir a nuestro árbol de directorios, una estructura de ficheros de otra máquina a través del protocolo NFS. Para ello, la máquina remota debe estar configurada de forma correcta. Supongamos que la máquina remota se llama *serv\_fich* y tiene la dirección IP 192.168.0.50. El directorio remoto al cual se va a poder acceder a través de NFS, por ejemplo, es */datos/pub*. La forma de incluir ese directorio de la máquina remota en nuestro árbol de directorios es:

```
# mount -t nfs serv_fich:/datos/pub /mnt/compartidos
```

o también:

```
# mount -t nfs 192.168.0.50:/datos/pub /mnt/compartidos
```

El directorio */mnt/compartidos* se ha elegido de forma arbitraria. Se puede utilizar cualquier punto de montaje. Después de ejecutar este comando, tendremos acceso a los ficheros del directorio */datos/pub* de la máquina *serv\_fich* ejecutando el comando `cd /mnt/compartidos`

Se puede modificar el fichero */etc/fstab* para incluir el montaje NFS anterior, añadiendo la siguiente línea:

```
serv_fich:/datos/pub    /mnt/compartidos  nfs    rw,hard    0 0
```

De esta forma, en el arranque de nuestro sistema se montará de forma automática el sistema de ficheros remoto.

### Configuración de un servidor

A continuación, se explica también la forma de configurar nuestra máquina para compartir algún directorio a través de NFS. Primero hay comprobar que el paquete que contiene el servidor *nfs* está instalado en el sistema. Su nombre es *nfs-utils*:

```
# rpm -qa | grep nfs-utils
```

Si no aparece nada, habría que instalarlo (ver capítulo de instalación de aplicaciones ). Después habría que arrancar el servidor *nfs*. Para ello ejecutar el comando:

```
# /etc/init.d/nfs start
```

o bien:

```
# service nfs start
```

Puede que queramos que el servidor *nfs* se inicie de forma automática al arrancar la máquina. Para ello, se utiliza el comando *chkconfig*. Este comando es genérico, es decir, se puede utilizar para añadir o quitar servicios del arranque de nuestra máquina.

Primero podemos comprobar en qué niveles de ejecución está incluido este servicio:

```
# chkconfig --list nfs
```

El comando `chkconfig` con la opción `--list` muestra los niveles de ejecución y si el servicio está activado o desactivado en cada nivel.

Por ejemplo, para añadir el servicio `nfs` al nivel 3, es decir, cuando el sistema arranque o pase a nivel de ejecución 3, se inicia el servidor `nfs`, ejecutar el siguiente comando:

```
# chkconfig --level 3 nfs on
```

Para comprobar que se ha incluido correctamente, se debe haber creado un fichero en el directorio `/etc/rc.d/rc3.d` llamado `SXXnfs`, `XX` es un número que indica el orden en el arranque del servicio:

K05saslauthd	K50snmptrapd	K95firstboot	S20random	S80sendmail
K15httpd	K50tux	S05kudzu	S24pcmcia	S85gpm
K24irda	K50vsftpd	S08iptables	S25netfs	S90crond
K25squid	K54pxe	S09isdn	S26apmd	S90cups
K34yppasswdd	K70aep1000	S10network	S28autofs	S90xfs
K35smb	K70bcm5820	S12syslog	S55sshd	S95anacron
K35winbind	K74ntpd	S13portmap	S56rawdevices	S95atd
K45named	K74ypserv	S14nfslock	S56xinetd	S97rhnsd
K50snmpd	K74ypxfrd	S17keytable	<b>S60nfs</b>	S99local

Hay que comprobar que el servicio está activo en memoria. Además, hay que comprobar que estén activos dos servicios más, que `nfs` utiliza: `portmap` y `rpc.mountd`

```
# service nfs status
# service portmap status
```

En este punto, ya estaría el servidor arrancado. Sólo falta configurar adecuadamente un fichero para realizar el control de acceso de `nfs`: `/etc/exports`

En este fichero debe haber una línea para cada sistema de ficheros exportado, indicando las máquinas que tienen acceso a ese sistema de ficheros y las opciones que indiquen la forma de acceso. Por ejemplo, para el configurar el servidor del ejemplo anterior. Se quiere dar acceso al directorio `/datos/pub` a tres máquinas llamadas `grupo1`, `grupo2` y `grupo3`:

```
# fichero /etc/export
/datos/pub          grupo1 grupo2 grupo3(ro)
/datos/admin        192.168.0.50(rw)
```

La opción `ro` da permisos de solo lectura. Para dar permisos de lectura/escritura utilizar la opción `rw`. También se puede dar acceso a máquinas especificando su dirección IP.

Por último, para que el servidor `nfs` actualice los cambios habría que reiniciar el servicio:

```
# service nfs reload
```

o también:

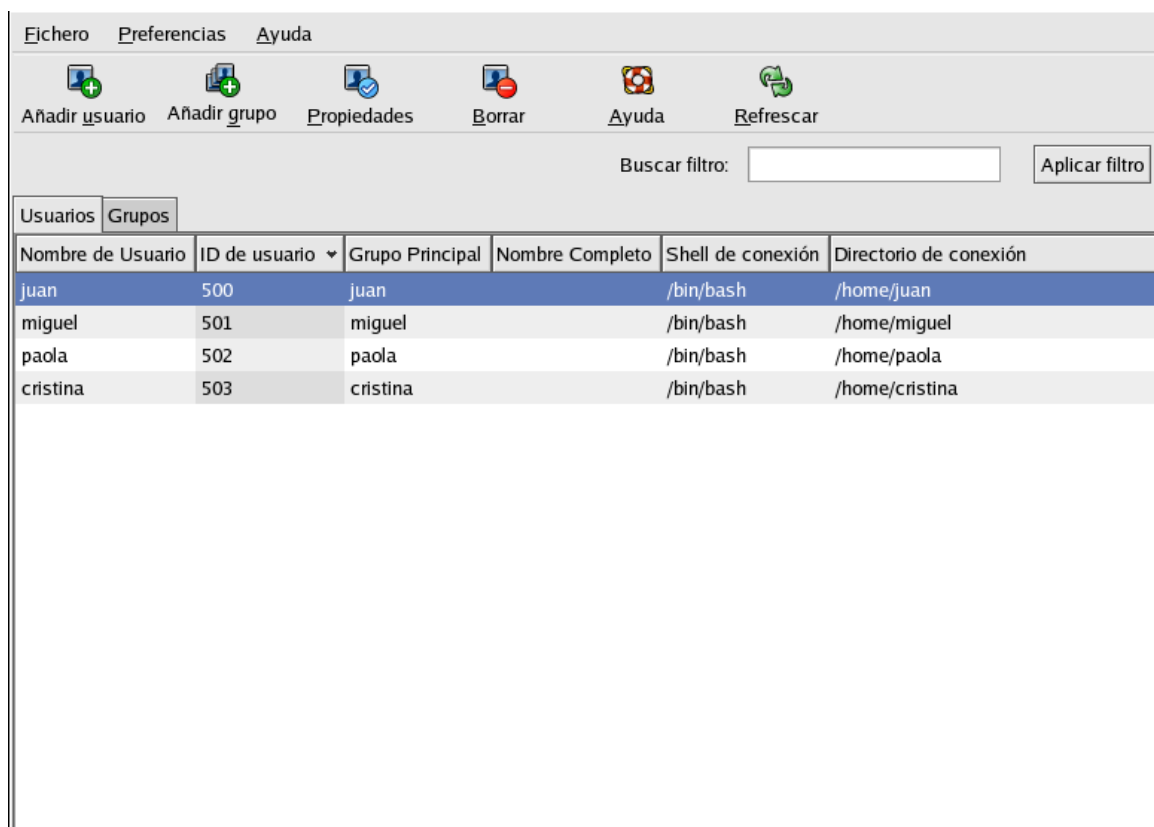
```
# /usr/sbin/exportfs -a
```

# 8. ADMINISTRACIÓN AVANZADA

## 8.1. Gestión de usuarios en entorno gráfico

Se puede crear una cuenta de usuario en modo gráfico usando el **Administrador de usuarios**. Para ello haz click en el icono de *Empezar aquí* en el escritorio. En la nueva ventana que se abrirá, haz click en *Configuración del sistema* y luego haz click en el icono *Usuarios & Grupos*. También puedes seleccionar *Menú principal / Configuración del sistema / Usuarios & Grupos* desde el panel.

Se puede iniciar el *Administrador de usuarios* al escribir `redhat-config-users` en un intérprete de comandos de la shell. Si no estás conectado como root, pedirá que introduzcas la contraseña de root. Cuando se abra la herramienta de configuración, haz click en *Añadir usuario*.



The screenshot shows the Red Hat User Manager GUI. At the top, there is a menu bar with 'Fichero', 'Preferencias', and 'Ayuda'. Below the menu bar is a toolbar with icons for 'Añadir usuario', 'Añadir grupo', 'Propiedades', 'Borrar', 'Ayuda', and 'Refrescar'. A search bar labeled 'Buscar filtro:' is located below the toolbar, with an 'Aplicar filtro' button to its right. Below the search bar are two tabs: 'Usuarios' (selected) and 'Grupos'. The main area contains a table with the following data:

Nombre de Usuario	ID de usuario	Grupo Principal	Nombre Completo	Shell de conexión	Directorio de conexión
juan	500	juan		/bin/bash	/home/juan
miguel	501	miguel		/bin/bash	/home/miguel
paola	502	paola		/bin/bash	/home/paola
cristina	503	cristina		/bin/bash	/home/cristina

**Figura 8-1. El Gestor de usuarios de Red Hat**

Introducir en la ventana de diálogo *Crear nuevo usuario* un nombre de usuario (puede ser una abreviatura o un apodo corto), el nombre completo del usuario para el que está creando la cuenta y la contraseña (que hay que volver a introducir para la verificación). El nombre del directorio home del usuario y el nombre de la shell por defecto. Después de hacer click en *OK*, aparecerá el nuevo usuario en la lista de usuarios, indicando que la creación de la cuenta del usuario se ha completado. Las acciones que se realizan al crear el usuario en modo gráfico son las mismas que en modo comando, es decir, se modifican los ficheros `/etc/passwd` y `/etc/shadow`. Desde la misma herramienta también se puede realizar las gestión de grupos.

## 8.2. Comandos de gestión de discos

### /sbin/fdisk

Comando para visualizar la tabla de particiones, crear y eliminar particiones. Para visualizar la tabla de particiones de todos los discos del sistema ejecutar:

```
# fdisk -l

Disco /dev/sda: 4294 MB, 4294967296 bytes
255 cabezas, 63 sectores/pista, 522 cilindros
Unidades = cilindros de 16065 * 512 = 8225280 bytes

Disposit. Inicio Principio Fin Bloques Id Sistema
/dev/sda1 * 1 13 104391 83 Linux
/dev/sda2 14 498 3895762+ 83 Linux
/dev/sda3 499 522 192780 82 Linux swap

Disco /dev/hda: 6284 MB, 6284113920 bytes
255 cabezas, 63 sectores/pista, 764 cilindros
Unidades = cilindros de 16065 * 512 = 8225280 bytes

Disposit. Inicio Principio Fin Bloques Id Sistema
/dev/hda1 1 637 5116671 2d Desconocido
/dev/hda2 638 764 1020127+ 6 FAT16

Disco /dev/hdb: 9426 MB, 9426170880 bytes
255 cabezas, 63 sectores/pista, 1146 cilindros
Unidades = cilindros de 16065 * 512 = 8225280 bytes

Disposit. Inicio Principio Fin Bloques Id Sistema
/dev/hdb1 1 1146 9213277+ 83 Linux
```

Para ver la tabla de particiones de un disco concreto:

```
# fdisk -l /dev/sdb
```

Si se ejecuta sin opciones, el comando es interactivo, es decir, se deben teclear letras para indicar la acción a realizar:

```
# fdisk /dev/sdb
```

Orden (m para obtener ayuda):

```
Orden Acción
a Conmuta el indicador de iniciable
b Modifica la etiqueta de disco BSD
c Conmuta el indicador de compatibilidad con DOS
d Suprime una partición
l Lista los tipos de particiones conocidos
m Imprime este menú
n Añade una nueva partición
o Crea una nueva tabla de particiones DOS vacía
p Imprime la tabla de particiones
q Sale sin guardar los cambios
s Crea una nueva etiqueta de disco Sun
t Cambia el identificador de sistema de una partición
u Cambia las unidades de visualización/entrada
v Verifica la tabla de particiones
w Escribe la tabla en el disco y sale
x Funciones adicionales (sólo para usuarios avanzados)
```

Orden (m para obtener ayuda) :

### **/sbin/mkfs**

Comando para formatear discos con el formato nativo de Linux. Por ejemplo, para dar formato a la tercera partición del segundo disco duro IDE del sistema, con el tipo de sistema de ficheros nativo de Linux ext3:

```
# mkfs -t ext3 /dev/hdb3
```

Si no se especifica la opción `-t` se formatea con el tipo ext2:

```
# mkfs /dev/hdb3
```

Comando equivalente que realiza el formateo sólo para sistemas de ficheros Linux ext2/ext3:  
**mke2fs**

### **/sbin/fsck**

Chequea y repara sistema de ficheros Linux. Con la opción `-A` se chequean todos los sistemas de ficheros montados en el sistema, si no se especifica la opción `-A` se debe indicar el sistema de ficheros a chequear

```
# fsck /  
# fsck /dev/sda
```

Otras opciones: `-a` repara automáticamente los errores que encuentra sin pedir confirmación. `-R` no chequea el directorio `/`.

## **8.3. Cuotas**

Las cuotas son una herramienta útil para supervisar el espacio en disco que utilizan los usuarios. Las cuotas pueden ser configuradas para usuarios individuales o para grupos. Para configurar las cuotas de disco seguir los siguientes pasos:

1. Habilitar las cuotas al sistema de ficheros, modificando el fichero `/etc/fstab`  
Para ello se utilizan las opciones **usrquota** y **grpquota**, como opciones en la línea correspondiente en el fichero `/etc/fstab`. Por ejemplo:

```
/dev/hdb1          /home             ext2 defaults,usrquota,grpquota 1 2
```

Aquí habilitamos las cuotas para el sistema de ficheros `/home`

2. Después del cambio en `/etc/fstab` hay que desmontar y volver a montar el sistema de ficheros implicado. Por ejemplo:  

```
# umount /home  
# mount /home
```
3. Crear los ficheros de cuota y generar la tabla de uso del disco.  
Para ello, ejecutar el comando `quotacheck`, el cual examina el sistema de ficheros habilitado para el uso de cuotas y construye una tabla con el uso de disco. Para crear los ficheros de



un usuario puede utilizar. Para cada tipo de cuota tenemos dos valores: soft y hard. El valor soft se utiliza como aviso. Cuando un usuario llega al valor soft se le permite mantener esa cantidad de bloques un periodo de tiempo llamado periodo de gracia (por defecto 7 días). Cuando llega al valor hard, no se le permite utilizar más bloques de disco.

En el ejemplo se observa que se ha asignado un valor soft de 200 MB y una hard de 250 MB para la cuota de bloques.

Una alternativa para asignar cotas es utilizar el comando `setquota`, que tiene esta forma de uso:

```
setquota nombre_usu block-soft block-hard inode-soft inode-hard filesystem
```

Otro comando importante es `quota`, que se utiliza para comprobar la cuota asignada a un usuario. Este comando lo puede ejecutar cada usuario para comprobar la cuota de disco que tiene disponible: **quota nombre\_usuario**

```
# quota usuariol
```

```
Disk quotas for user usuariol (uid 500):
```

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/hdb1	34778	200000	250000		422	0	0	

5. Informe de cuotas de disco. Se puede obtener un informe de las cuotas existentes con el comando `repquota`:

```
# repquota /home          Muestra coutas de /home
# repquota -a            Muestra coutas de todos los sistema de ficheros
```

6. Comandos para habilitar y deshabilitar cuotas:

```
# quotaoff -vaug         Deshabilita todas las cuotas de todos los sistemas de ficheros
# quotaon -vaug          Habilita todas las cuotas de todos los sistemas de ficheros
# quotaon -vug /home     Habilita las cuotas sólo de /home
```

7. Mantenimiento: se recomienda ejecutar periódicamente el comando `quotacheck` para actualizar los ficheros de cuota. Para ello, antes deshabilitar las cuotas. Un buen lugar para realizar esta tarea es en el cron, programando una hora de bajo uso del sistema.

```
# Actualización periódica sólo para el sistema de ficheros de /home
quotaoff /home
quotacheck -vug /home
quotaon /home
```

## 8.4. Ejemplo de tarea de administración

Suponemos que administramos un sistema Linux en el que inicialmente existe un sólo disco duro (/dev/hda) con una partición para swap y el resto del espacio para otra partición donde está instalado Linux. Hemos adquirido un nuevo disco duro y queremos utilizarlo para albergar en él, el directorio /home del sistema.

Suponemos que el disco se conecta como secundario del primer canal IDE, por lo tanto, su nombre en el sistema será `/dev/hdb`. Después de instalar el nuevo disco duro, hay que crear una partición primaria que ocupe todo el disco.

```
# fdisk /dev/hdb
```

Dentro de `fdisk` elegir la opción `n` para añadir una partición.

Y a continuación formatear la nueva partición con el comando `mkfs`. Para un sistema de ficheros `ext2` (por defecto):

```
# mkfs /dev/hdb1
```

Para un sistema de ficheros `ext3`:

```
# mkfs -t ext3 /dev/hdb1
```

El nuevo disco ya está formateado. A continuación hay que montarlo en el sistema de ficheros del sistema. De forma temporal, lo vamos a montar en el directorio `/mnt/HOME` para copiar todos los datos del directorio `/home` original:

```
# mkdir /mnt/HOME
# mount -t auto /dev/hdb1 /mnt/HOME
```

Con esto, ya podemos acceder al disco. Ahora vamos a copiar todo el contenido del directorio `/home` al nuevo disco, es decir, a `/mnt/HOME`, para ello utilizamos el comando `tar`. No se puede utilizar el comando de copia `cp` porque este comando cambia el propietario de todos los ficheros que se copian al usuario que realiza la copia, en este caso `root`. Sin embargo, nosotros queremos hacer la copia sin alterar el propietario original de cada fichero.

```
# cd /home
# tar cvf USERS.tar *
```

Este comando crea un fichero `tar` con el contenido de todo el directorio `/home`. Ahora sólo tenemos que moverlo al nuevo disco y desempaquetarlo allí.

```
# mv USERS.tar /mnt/HOME
# cd /mnt/HOME
# tar xvf USERS.tar
```

Comprobamos que toda la estructura de directorios de `/home` la tenemos en `/mnt/HOME`. Borramos el contenido de `/home`.

```
# rm -rf /home/*
```

Lo siguiente es desmontar el nuevo disco de su ubicación provisional y montarlo en la definitiva:

```
# umount /mnt/HOME
# mount -t auto /dev/hdb1 /home
```

Después de esto, comprobamos que todo está correcto y si es así, sólo queda modificar el fichero `/etc/fstab` para que el montaje del disco se realice siempre en el arranque del sistema. Añadimos la siguiente línea:



```
/dev/hdb1          /home          ext2  defaults          1 2
```

Con esto quedaría completado el procedimiento de añadir un nuevo disco y asignarle el directorio /home de nuestro sistema Linux.

## 8.5. Archivos de registro del sistema

Los archivos de registro o archivos log son ficheros que contienen mensajes del sistema, incluyendo mensajes del kernel, de los servicios y las aplicaciones que se ejecutan. Hay diferentes archivos log para diferentes tipos de información.

Los ficheros log pueden ser muy útiles cuando intentamos resolver un problema en el sistema como la carga de un driver del kernel o cuando queremos ver los intentos de acceso fallidos al sistema.

Algunos de los ficheros log se controlan por el servicio **syslogd**. La lista de ficheros log controlados por syslogd se puede ver en su fichero de configuración */etc/syslog.conf*

Los ficheros log se encuentran en el **directorio /var/log**. La mayoría de ellos son ficheros cíclicos, es decir, cuando se alcanza un determinado tamaño se vacían y se comienza a escribir de nuevo. Esto es así ya que estos ficheros almacenan mucha información y en poco tiempo podrían tener un tamaño demasiado grande. Además, se puede configurar para que se almacenen varias copias sucesivas del mismo fichero antes de sobrescribirse. Los ficheros de configuración para esta característica cíclica son */etc/logrotate.conf* y */etc/logrotate.d*

Los ficheros log más comunes son:

<i>/var/log/boot.log</i>	Fichero cíclico de mensajes de arranque del sistema, con fecha y hora
<i>/var/log/cron</i>	Fichero cíclico de mensajes del proceso crond, con fecha y hora
<i>/var/log/dmesg</i>	Fichero de mensajes del kernel en el último arranque
<i>/var/log/maillog</i>	Fichero cíclico de mensajes del sistema de correo (sendmail)
<i>/var/log/secure</i>	Fichero cíclico de mensajes de gestión de usuarios, con fecha y hora
<i>/var/log/messages</i>	Fichero cíclico de mensajes generales del sistema, con fecha y hora

Estos ficheros son de texto, por lo tanto, se pueden visualizar por cualquier editor. Además, en el entorno gráfico de Red Hat existe una aplicación gráfica: */usr/bin/redhat-logviewer*

Para acceder a ella, desde el menú: *Menú Principal / Herramientas del sistema / Registro del sistema*

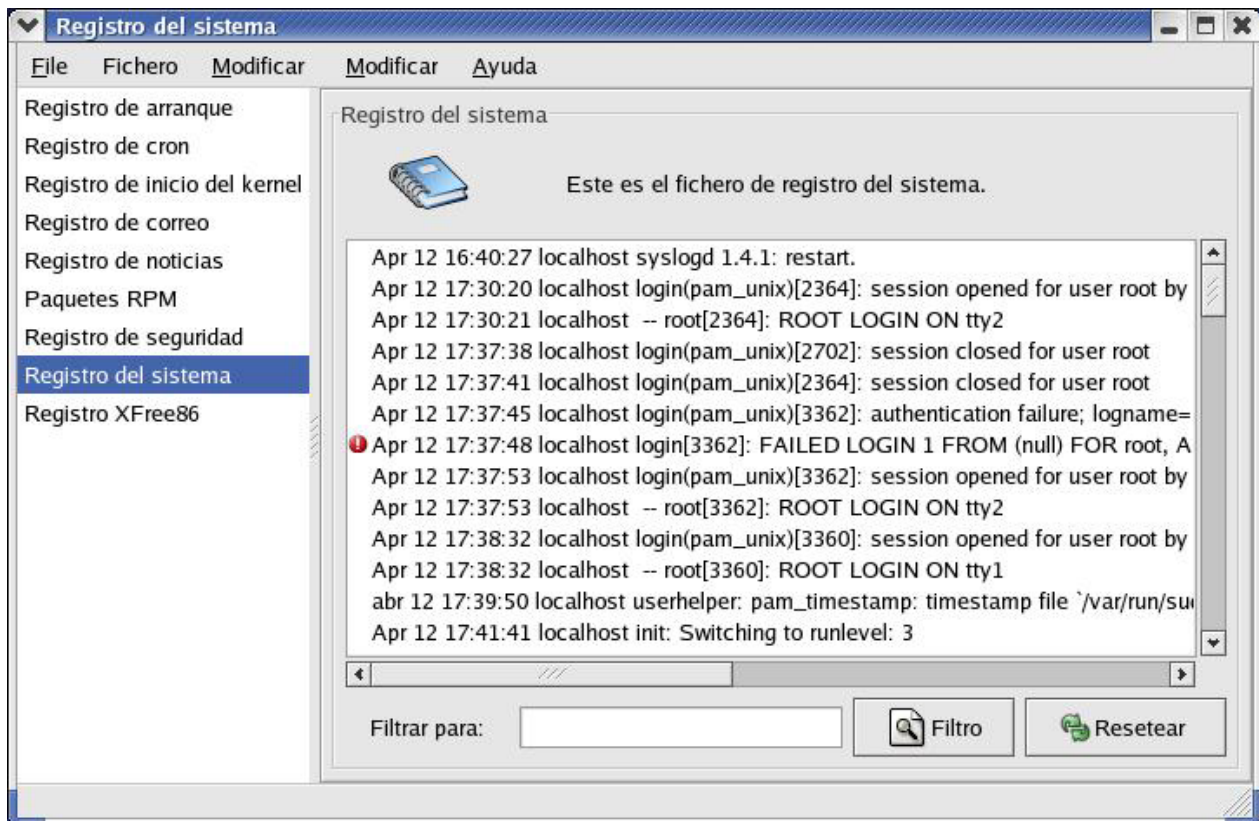


Figura 8-3. Aplicación para visualizar los archivos de registro

## 8.6. Planificación de trabajos: cron

cron es un demonio (servicio) del sistema Linux que se usa para poder planificar la ejecución de tareas de acuerdo con una combinación de hora, día del mes, mes, día de la semana y semana.

cron asume que el sistema está permanentemente activo. Es decir, si el sistema no está arrancado cuando una tarea está planificada, ésta no se ejecutará. Para que funcione correctamente, el servicio **crond** debe estar arrancado. Para comprobarlo, ejecuta el comando:

```
# /sbin/service crond status
```

O también:

```
# ps -ef | grep crond
```

El fichero de configuración para cron es **/etc/crontab**

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

Las primeras líneas son variables usadas para configurar el entorno en el cual las tareas cron se ejecutan. A continuación aparecen las líneas de planificación. Cada línea representa una tarea y tiene el siguiente formato:

```
minuto hora día mes día_semana comando
```

El comando que se lanza en cada línea produce la ejecución de los scripts que se almacenen en los directorios `/etc/cron.hourly`, `/etc/cron.daily`, ...

Cada usuario del sistema pueden utilizar cron para planificar sus propias tareas. La información sobre las tareas planificadas por cada usuario se almacena en el directorio `/var/spool/cron`, en un fichero de nombre igual al del usuario. Las tareas planificadas por root se almacenan en `/var/spool/cron/root`. Suelen ser tareas de administración y mantenimiento del sistema.

Este fichero no se debe editar directamente. Para ello, ejecutar el comando `crontab -e`. El formato es el mismo que el fichero `/etc/crontab`. Para visualizar el contenido del fichero: `crontab -l`

Ejemplos:

```
# almacena el uso de memoria del sistema cada lunes
# a las 3:30 AM en el fichero /tmp/meminfo
30 3 * * mon cat /proc/meminfo >> /tmp/meminfo
# ejecuta un script para realizar un backup el primer
# día de cada mes a las 4:10 AM
10 4 1 * * /root/scripts/backup.sh
```

# 9. INSTALACIÓN DE APLICACIONES

## 9.1. RPM: Red Hat Packet Manager

RPM es un sistema desarrollado por Red Hat (aunque actualmente se usa también en otras distribuciones) para facilitar la instalación de componentes de Linux. Los paquetes rpm son ficheros que contienen todos los archivos que componen un determinado programa. Además contienen la documentación, configuración, información sobre su instalación y desinstalación, y en general todo lo necesario para que el mantenimiento de las aplicaciones sea lo más robusto posible.

Para ejecutar el comando rpm es necesario entrar como usuario root.

Formato del nombre de los archivos rpm:  
nombre-versión-arquitectura.rpm

Por ejemplo:

```
xcircuit-3.1.18-1.i386.rpm
```

Nombre del paquete: xcircuit-3.1.18-1

Aplicación: xcircuit

Versión: 3.1.18-1

Arquit.: i386 (Intel)

```
aumix-2.7-16.i386.rpm
```

Opciones de instalación /desinstalación:

-i	Instalar
-v	Verbose, muestra información
--test	Realiza un test sin hacer la instalación
--force	Fuerza la instalación
--nodeps	No verifica dependencias
--replacpkgs	Si el paquete ya está instalado, lo reemplaza
-U	Actualiza un paquete guardando toda su configuración
-e	Desinstalar

Para instalar un programa correctamente hay que instalar el paquete correspondiente y todas las dependencias. Para ver si un paquete tiene dependencias y cuales son, ejecutar el comando:

```
# rpm -iv --test nombre_archivo.rpm
```

Si el comando anterior muestra que existen dependencias, éstas tienen que instalarse primero.

```
# rpm -iv --test gimp-1.2.3-16.i386.rpm
warning: gimp-1.2.3-16.i386.rpm: V3 DSA signature: NOKEY, key ID db42a60e
error: Failed dependencies:
        gimp-print-plugin is needed by gimp-1.2.3-16
```

En el ejemplo anterior vemos que al hacer el test de la instalación de un paquete nos da un error de dependencias, es decir, el paquete que queremos instalar depende de otro llamado gimp-print-plugin así que habrá que buscarlo e instalarlo primero. Después de la instalación de las dependencias, se puede ejecutar el comando:

```
# rpm -iv nombre_archivo.rpm
```

Por ejemplo:

```
# rpm -i gimp-1.2.3-16.i386.rpm
warning: gimp-1.2.3-16.i386.rpm: V3 DSA signature: NOKEY, key ID db42a60e
Preparing packages for installation...
gimp-1.2.3-16
```

Para desinstalar el paquete:

```
# rpm -e nombre_paquete
```

Opciones de información

```
-qi   Muestra información sobre un paquete
-ql   Muestra los archivos pertenecientes al paquete
-qs   Visualiza el estado de los archivos del paquete
-qd   Visualizar los archivos de documentación del paquete
-qc   Visualizar los archivos de configuración del paquete
-qa   Visualizar todos los paquetes instalados
-qf fichero  Muestra el paquete al cual pertenece el fichero especificado
-V    Verificar la correcta instalación de un paquete
```

Las opciones de información se utilizan pasando como segundo argumento el nombre del paquete. Por ejemplo:

```
# rpm -qi xcircuit-3.1.18-1
```

Si queremos obtener información sobre el contenido de un archivo rpm sin instalarlo, utilizar las opciones:

```
rpm -qpi nombre_archivo.rpm   Información del paquete
rpm -qpl nombre_archivo.rpm   Lista de ficheros del paquete
```

Podemos verificar todos los paquetes del sistema con el comando:

```
# rpm -Va
```

Para buscar un paquete que necesitamos instalar existen dos opciones:

1. Utilizar los discos de la distribución. En todos ellos, el directorio donde se almacenan los ficheros rpm es /RedHat/RPMS
2. Utilizar un buscador Web. Sitios Web de búsqueda de paquetes RPM:  
<http://rpmfind.net>  
<http://freshmeat.net>  
[www.sourceforge.com](http://www.sourceforge.com)

## 9.2. Otros métodos de instalación

### Aplicación aptget

Aplicación de instalación avanzada, busca las dependencias en la red. Esta aplicación proviene de la distribución de Debian.

### Programas de instalación ejecutables mediante ficheros .bin o .sh

Las aplicaciones que utilizan este método suelen utilizar ficheros comprimidos de extensión .tar.gz, conteniendo los ficheros de instalación. Ejemplos: Acrobat Reader o MatLab.

## 9.3. Herramienta de instalación gráfica

El entorno gráfico de Red Hat incluye una herramienta gráfica de instalación de aplicaciones que utiliza también el formato de ficheros RPM, llamada **Herramienta de administración de paquetes**.

Puedes colocar el primer CD-ROM de Red Hat Linux en la unidad de CD-ROM. Selecciona el botón *Si* cuando pregunta si se desea ejecutar el programa autorun desde el CD. Luego pedirá la contraseña de root para poder instalar los paquetes.

Si escribes la contraseña correcta, aparecerá la interfaz de *Herramienta de administración de paquetes* que le permite escoger los grupos de paquetes a instalar así como también paquetes individuales dentro de grupos.

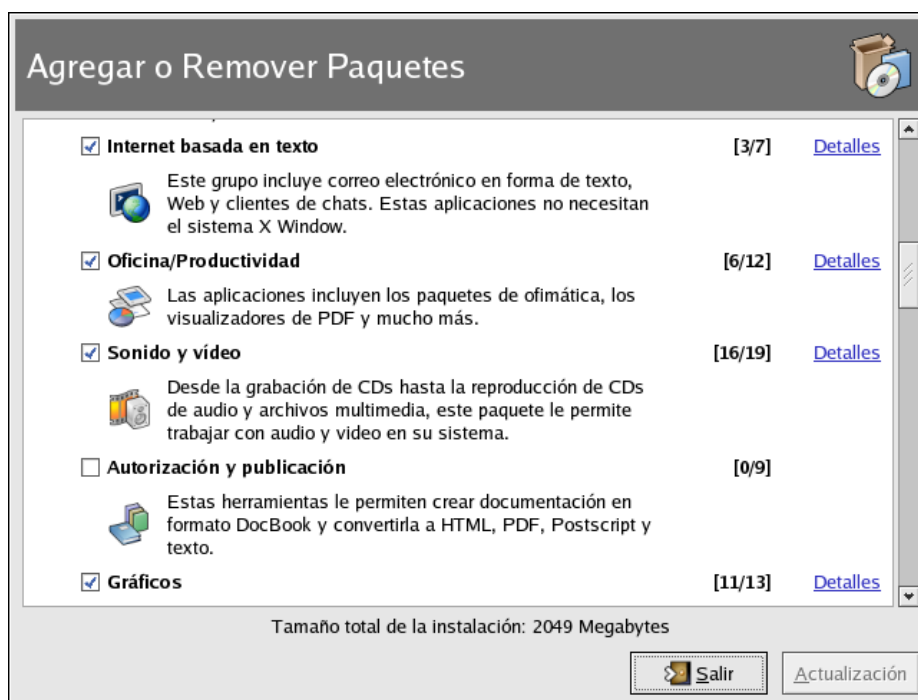
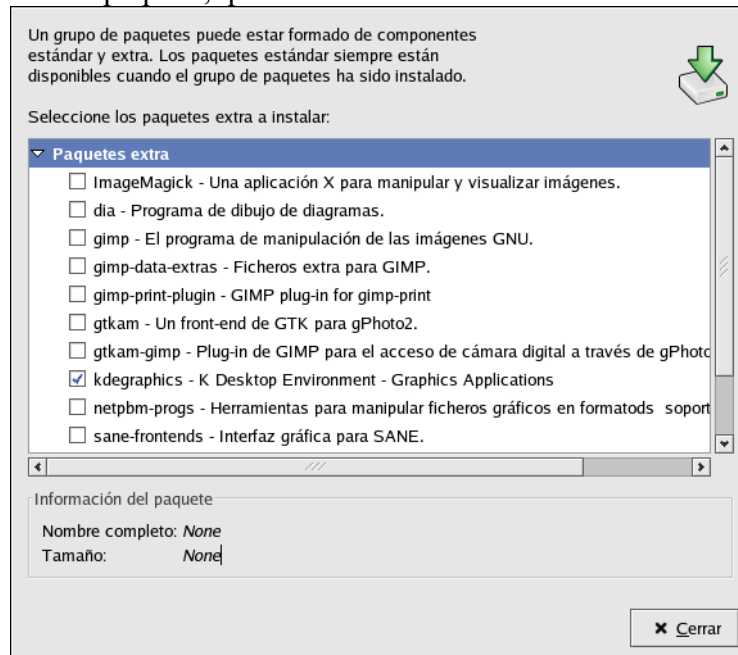


Figura 9-1. Instalación de software con la Herramienta de administración de paquetes

La *Herramienta de administración de paquetes* marca qué paquetes están ya instalados en el sistema. Puedes agregar paquetes haciendo click en la casilla de verificación al lado de cada paquete. Para desinstalar un paquete, quita la marca de verificación.



**Figura 9-2. Selección individual de paquetes**

Después de seleccionar los paquetes, presione el botón **Update** para instalar o desinstalar los paquetes seleccionados.

# 10. BIBLIOGRAFÍA

Red Hat Linux Customization Guide  
Red Hat Linux Reference Guide  
Red Hat Linux 9: Manual del principiante de Red Hat Linux  
FAQ Linux v2.0.2  
SuSe Linux 8.1 Manual de Administración  
La biblia de UNIX. Steve Moritsugu. Ed. Anaya  
[www.controltotal.org](http://www.controltotal.org)  
[www.linuxfocus.org](http://www.linuxfocus.org)

Cualquier comentario, sugerencia u opinión se puede dirigir a:  
Manuel Santos González    [ms.gonzalez@coitt.es](mailto:ms.gonzalez@coitt.es)